REDEFINING "COMPUTATION" ITSELF

THE HYPERDIMENSIONAL ALGORITHM

A NEW DEFINITION OF "COMPUTATION"

超维算法

核心要点：

1 多维结构
数据不是单一输入输出，
而是多起点、多结果的结构网络

2 非线性因果
结果可以先于原因，
多路径并行存在

3 随机有序
随机性与结构性同时存在于系统内部

4 一数多果
同一数据可对应多个结果，
不依赖单一计算路径

5 入口触发
通过改变入口而非修改数据，
实现不同结果输出

MULTI-DIMENSIONAL STRUCTURE
Data as nodes. Not just inputs or outputs, but multiple origins and multiple results.

NON-LINEAR CAUSALITY
Results can precede causes. Multiple paths. Recursive and interconnected.

RANDOM YET ORDERED
Randomness and order coxrist as an intrinsic part of the algorithmic structure.

ONE DATA, MANY RESULTS
The same metadata can correspond to different results through different structural entrances.

ENTRANCE-BASED ACTIVATION
Change the entrance, not the data. Activate different states and outcomes.

A FOUNDATIONAL FRAMEWORK
FOR HYPERDIMENSIONAL COMPUTATION
AND A NEW SCIENTIFIC PARADIGM

Contents

[极限算法]超维算法 1
引言：为什么要重新讨论“算法” 1
第一部分：当下对“算法”的标准定义 2
第二部分：“超维算法”的定义 3
第三部分：超维算法与传统算法的根本差异 5
第四部分：果因论哲学 6
第五部分：一个日常例子——裤子与腰部折叠 7
第六部分：不修改元数据的数据观 8
第七部分：重新定义“超算” 9
第八部分：实证基础——多领域系统验证与新范式判定 10
第九部分：常见误解澄清 11
第十部分：超维算法与人类计算史的关系 12
第十一部分：总结——这不是优化，而是重定义 12
附录：本文的边界与开放问题 14
参考文献性说明 15
[Extreme Algorithm] Hyperdimensional Algorithm 16
Introduction: Why We Need to Rediscuss "Algorithm" 17
Part One: The Current Standard Definition of "Algorithm" 18
Part Two: Definition of the "Hyperdimensional Algorithm" 19
Part Three: Fundamental Differences Between the Hyperdimensional Algorithm and Traditional Algorithms 22
Part Four: The Effect-Cause Theory Philosophy 24
Part Five: An Everyday Example—Pants and Folding the Waist 25
Part Six: A Data View of Not Modifying Metadata 28
Part Seven: Redefining "Supercomputing" 29

Part Eight: Empirical Foundations—Multi-Domain System Validation and New Paradigm Criteria 30
Part Nine: Clarifying Common Misunderstandings 31
Part Ten: The Hyperdimensional Algorithm and the History of Human Computation 33
Part Eleven: Conclusion—This Is Not Optimization, but Redefinition 34
Appendix: Boundaries of This Paper and Open Questions 35
Bibliographic Note 37
[Algorithme Extreme] Algorithme Hyperdimensionnel 38
Introduction : Pourquoi nous devons rediscuter de l'« algorithme » 39
Première partie : La définition standard actuelle de l'« algorithme » 40
Deuxième partie : Définition de l'« algorithme hyperdimensionnel » 42
Troisième partie : Différences fondamentales entre l'algorithme hyperdimensionnel et les algorithmes traditionnels 45
Quatrième partie : La philosophie de la théorie effet-cause 47
Cinquième partie : Un exemple quotidien — le pantalon et le pliage de la taille 48
Sixième partie : Une vision des données consistant à ne pas modifier les métadonnées 51
Septième partie : Redéfinir le « supercalcul » 52
Huitième partie : Fondements empiriques — Validation par des systèmes multidomaines et critères d'un nouveau paradigme 53
Neuvième partie : Clarification des malentendus courants 55
Dixième partie : L'algorithme hyperdimensionnel et l'histoire du calcul humain 56
Onzième partie : Conclusion — Ce n'est pas une optimisation, mais une redéfinition 57
Annexe : Frontières de cet article et questions ouvertes 59
Note bibliographique 60
[Algoritmo Extremo] Algoritmo Hiperdimensional 62
Introducción: Por qué necesitamos rediscutir el "algoritmo" 63
Primera parte: La definición estándar actual de "algoritmo" 64

Segunda parte: Definición del "algoritmo hiperdimensional" 66
Tercera parte: Diferencias fundamentales entre el algoritmo hiperdimensional y los algoritmos tradicionales 69
Cuarta parte: La filosofía de la teoría efecto-causa 71
Quinta parte: Un ejemplo cotidiano — pantalón y doblar la cintura 72
Sexta parte: Una visión de datos de no modificar metadatos 75
Séptima parte: Redefiniendo el "supercálculo" 76
Octava parte: Fundamentos empíricos — Validación por sistemas multidominio y criterios de nuevo paradigma 77
Novena parte: Aclaración de malentendidos comunes 79
Décima parte: El algoritmo hiperdimensional y la historia del cálculo humano 80
Undécima parte: Conclusión — Esto no es optimización, sino redefinición 81
Apéndice: Límites de este artículo y preguntas abiertas 83
Nota bibliográfica 84
[極限アルゴリズム] 超次元アルゴリズム 86
序論：なぜ「アルゴリズム」を再議論する必要があるのか 87
第一部：現在の「アルゴリズム」の標準的な定義 88
第二部：「超次元アルゴリズム」の定義 89
第三部：超次元アルゴリズムと伝統的アルゴリズムの根本的差異 92
第四部：果因論の哲学 93
第五部：日常的な例——ズボンとウエスト折り 94
第六部：メタデータを変更しないデータ観 97
第七部：「超計算」の再定義 98
第八部：実証的基盤——複数領域システム検証と新パラダイム判定 99
第九部：よくある誤解の明確化 100
第十部：超次元アルゴリズムと人類の計算史 101
第十一部：結論——これは最適化ではなく、再定義である 102
付録：本稿の境界と未解決問題 104
参考文献的注記 106

[المتطرفة الخوارزمية] الأبعاد فائقة الخوارزمية 107
مقدمة: لماذا نحتاج إلى إعادة مناقشة "الخوارزمية" 107
الجزء الأول: التعريف المعياري الحالي للخوارزمية 108
الجزء الثاني: تعريف الخوارزمية فائقة الأبعاد 109
الجزء الثالث: الاختلافات الجوهرية بين الخوارزميات التقليدية والخوارزمية فائقة الأبعاد 111
الجزء الرابع: فلسفة نظرية التأثير والسبب 112
الجزء الخامس: مثال يومي — الخصر وطي البنطال 113
الجزء السادس: نظرة بيانات تعديل عدم بيانات الوصفية 115
الجزء السابع: إعادة تعريف "الفائقة الحوسبة" 116
الجزء الثامن: الأسس التجريبية — التحقق عبر أنظمة متعددة المجالات ومعايير النموذج الجديد 116
الجزء التاسع: توضيح المفاهيم الخاطئة الشائعة 117
الجزء العاشر: تاريخ الأبعاد فائقة الخوارزمية والبشرية الحوسبة 118
الجزء الحادي عشر: الخاتمة — ليس هذا تحسينًا، بل إعادة تعريف 119
الملحق: حدود هذه الورقة والأسئلة المفتوحة 121
ملاحظة ببليوغرافية 122
[Extremalgorithmus] Hyperdimensionaler Algorithmus 123
Einleitung: Warum wir "Algorithmus" neu diskutieren müssen 124
Teil Eins: Die aktuelle Standarddefinition von "Algorithmus" 125
Teil Zwei: Definition des "hyperdimensionalen Algorithmus" 127
Teil Drei: Grundlegende Unterschiede zwischen dem hyperdimensionalen Algorithmus und traditionellen Algorithmen 130
Teil Vier: Die Philosophie der Wirkung-Ursache-Theorie 132
Teil Fünf: Ein alltägliches Beispiel – Hose und Falten der Taille 133
Teil Sechs: Eine Datenperspektive des Nicht-Modifizierens von Metadaten 136
Teil Sieben: Neudefinition von "Supercomputing" 137
Teil Acht: Empirische Grundlagen – Mehrbereichs-Systemvalidierung und neue Paradigmenkriterien 138
Teil Neun: Klärung häufiger Missverständnisse 140

Teil Zehn: Der hyperdimensionale Algorithmus und die Geschichte des menschlichen Rechnens 141
Teil Elf: Fazit – Dies ist keine Optimierung, sondern Neudefinition 143
Anhang: Grenzen dieses Papers und offene Fragen 145
Bibliografische Anmerkung 146
[Algoritmo Extremo] Algoritmo Hiperdimensional 148
Introdução: Por que precisamos rediscutir "algoritmo" 149
Parte Um: A definição padrão atual de "algoritmo" 150
Parte Dois: Definição do "algoritmo hiperdimensional" 151
Parte Três: Diferenças fundamentais entre o algoritmo hiperdimensional e os algoritmos tradicionais 154
Parte Quatro: A filosofia da teoria efeito-causa 156
Parte Cinco: Um exemplo cotidiano — calça e dobrar a cintura 157
Parte Seis: Uma visão de dados de não modificar metadados 160
Parte Sete: Redefinindo "supercomputação" 162
Parte Oito: Fundamentos empíricos — Validação por sistemas multi-domínio e critérios de novo paradigma 163
Parte Nove: Esclarecimento de mal-entendidos comuns 164
Parte Dez: O algoritmo hiperdimensional e a história da computação humana 166
Parte Onze: Conclusão — Isto não é otimização, mas redefinição 167
Apêndice: Limites deste artigo e questões em aberto 169
Nota bibliográfica 170
[Экстремальный алгоритм] Гипермерный алгоритм 172
Введение: Почему нам нужно переобсудить «алгоритм» 173
Часть первая: Современное стандартное определение «алгоритма» 174
Часть вторая: Определение «гипермерного алгоритма» 176
Часть третья: Фундаментальные различия между гипермерным алгоритмом и традиционными алгоритмами 179
Часть четвертая: Философия теории следствия-причины 181

Часть пятая: Повседневный пример — брюки и складывание пояса 182
Часть шестая: Взгляд на данные как на неизменность метаданных 185
Часть седьмая: Переопределение «супервычислений» 187
Часть восьмая: Эмпирические основы — многодоменная системная валидация и критерии новой парадигмы 188
Часть девятая: Прояснение распространенных недоразумения 189
Часть десятая: Гипермерный алгоритм и история человеческих вычислений... 191
Часть одиннадцатая: Заключение — Это не оптимизация, а переопределение192
Приложение: Границы данной статьи и открытые вопросы 194
Библиографическое примечание 195
[익스트림 알고리즘] 초차원 알고리즘 197
서론: 왜 우리는 '알고리즘'을 재논의해야 하는가 197
제 1 부: 현재의 '알고리즘'에 대한 표준적 정의 198
제 2 부: '초차원 알고리즘'의 정의 200
제 3 부: 초차원 알고리즘과 전통적 알고리즘의 근본적 차이 202
제 4 부: 결과-원인 이론 철학 204
제 5 부: 일상적 예시 — 바지와 허리 접기 205
제 6 부: 메타데이터를 수정하지 않는 데이터관 207
제 7 부: '초월 계산'의 재정의 208
제 8 부: 실증적 기반 — 다영역 시스템 검증과 새 패러다임 판정 209
제 9 부: 흔한 오해의 해소 210
제 10 부: 초차원 알고리즘과 인류 계산사 212
제 11 부: 결론 — 이것은 최적화가 아니라 재정의이다 212
부록: 본 논문의 경계와 열린 질문들 214
참고문헌적 설명 216

[极限算法]超维算法

一种对“计算”本身的重新定义

作者：巫朝晖 JEFFI CHAO HUI WU

摘要

本文提出“超维算法”这一新概念，并将其作为对传统算法定义边界的一次结构性重审。传统算法通常建立在输入、步骤、规则、输出的线性框架之上，强调有穷性、确定性、能行性、可行性以及清晰的输入输出边界。现代超级计算机虽然拥有极高算力，但其底层逻辑仍然主要是在更大规模上执行既定算法范式。本文认为，真正值得讨论的“超算”并不只是算力规模的增长，而是计算范式本身的改变。超维算法不是传统算法的加速版，也不是数学算法的扩展，而是一种多维、叠加、随机而有序的结构性计算观。在这一结构中，数据不再只是被动输入或终点输出，而是同时具备多个起点与多个结果的节点属性；结果不再只是终点，也可以成为新的入口；元数据不必被反复修改，而可以通过结构入口、关系变化与条件激活，对应不同状态与结果。本文以“果因论”哲学和“裤子腰部折叠”这一日常例子，说明超维算法如何通过改变结构入口，使目标结果直接成立，从而减少重复计算，甚至绕开原有计算路径。本文旨在为“超维算法”建立初步定义、结构边界与思想基础，为未来进一步展开超维计算、极限算法与新科学体系提供原始理论节点。按“存在且成立即为新范式”的标准，超维算法作为一种已被多领域系统验证的新计算结构范式，在当下已经成立。

关键词： 超维算法；极限算法；超维计算；果因论；元数据；计算范式；结构入口；非线性因果；新科学

引言：为什么要重新讨论“算法”

我提出“超维算法”，并不是为了给现有算法换一个更大的名称，也不是为了把传统算法包装成某种新奇概念。恰恰相反，我要讨论的是一个更根本的问题：人类当下对“算法”的理解，是否已经被限制在一个过于狭窄的框架之中？

今天人们一说算法，通常想到的是数学公式、程序代码、逻辑步骤、输入输出、模型训练、数据处理、路径优化、搜索排序、自动推荐和人工智能推理。这些当然都是算法的重要形式，也支撑了现代计算机、互联网、数据库、人工智能、超级计算、工业系统和信息社会的运行。但是，如果把算法只理解为“一套有序步骤”，只理解为“从输

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

入开始，经过规则处理，最后得到输出”的过程，那么这种理解本身就已经把计算限制在一个低维框架之内。

我们生活在一个被算法支配的时代。从搜索引擎到推荐系统，从天气预报到人工智能，算法的边界就是人类智能的边界。但有一个问题很少被追问：算法只能是这样吗？为什么计算必须遵循“输入→步骤→输出”的线性模式？为什么相同的问题每次都要从头计算？本文试图挑战这些默认假设，提出一种完全不同的计算范式——超维算法。按“存在且成立即为新范式”的标准，超维算法作为一种已被多领域系统验证的新计算结构范式，在当下已经成立。

需要特别说明的是，本文提出的“超维算法”与学术界已有近三十年发展历史的“超维计算”（Hyperdimensional Computing, HDC）是完全不同的概念。HDC 使用极高维度的随机二进制向量进行编码和运算，追求更高效的表示和计算，但仍属于“输入→处理→输出”的线性范式之内。超维算法则完全不同：它追问的是结果是否可以通过结构直接成立，是否可以绕过计算路径本身，是否可以不修改元数据而适配多个结果。两者名称相近，但内涵相反。超维算法不是对算法的升级，而是对“算法是否必须存在”这一前提的重新提问。

第一部分：当下对“算法”的标准定义

在主流计算机科学、数学和工程体系中，算法有一个被长期接受的基本定义：算法是一个定义良好的、由有限步骤构成的计算过程，它接受一个或多个输入，经过确定性的规则转换，在有限时间内产生一个或多个输出。这一理解不是某一个人的随意观点，而是现代计算文明长期形成的基础共识。它支撑了软件、硬件、数据库、网络系统、人工智能模型和超级计算中心的底层逻辑。无论算法表现得多么复杂，其核心仍然围绕输入、规则、步骤和输出展开。

这一标准算法观念可以拆解为几个核心特征。

第一，有穷性。算法必须在有限步骤内终止，不能无限循环，也不能永远运行下去。一个永远不会停下来的过程，即使形式上可以被描述，也很难被视为有效算法。所有科学计算程序、数据处理流程，都有明确的结束条件。

第二，确定性。相同的输入，在相同条件下，应当产生相同的输出。即使某些算法引入随机数，也通常是受控的随机、可复现的随机，或者在统计意义上可解释的概率机制，而不是完全不可把握的内在随机。数值模拟的结果必须可复现，这是科学计算的基本要求。

第三，明确的输入输出边界。输入在前，处理在中，输出在后，起点和终点具有清楚的结构边界。你给算法什么数据，它处理完之后给你什么结果，这个边界是分明的。输入在前，处理在中，输出在后，顺序不可颠倒。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第四，能行性。算法中的每一步都必须是实际可执行的操作，不能包含无法执行、无法描述、无法验证的跳跃。每一步都必须是人类能用纸笔或者计算机能实际执行的基本操作——加减乘除、逻辑判断、数据读写等。

第五，可行性。一个算法即使在理论上成立，如果消耗的时间、算力、内存或存储资源完全不可接受，在工程意义上也难以成为有效算法。一个算法如果理论上正确但需要几亿年才能算完，在工程上不被视为可行的算法。

这一整套算法观念，最终可以追溯到图灵机模型。图灵机作为现代计算理论的核心抽象，给出了“可计算”的基本边界。无论今天的计算机多么强大，无论芯片多么先进，无论超级计算中心规模多么庞大，其底层逻辑仍然没有真正脱离这一条路径：输入、规则、步骤、输出。现代超级计算机只是把这一过程用更大的硬件规模、更高的并行度、更复杂的系统调度加速执行。也就是说，传统语境中的“超算”，主要仍然是算力规模的扩张，而不一定是计算范式的根本改变。算力可以成千上万倍增长，但如果底层逻辑仍然是“输入、步骤、输出”，那么它仍然处在传统算法范式之内。

第二部分：“超维算法”的定义

我所提出的“超维算法”，正是在这一背景下出现的。它不是传统算法的改良版，不是更快的算法，不是更复杂的数学公式，也不是某种更高级的程序技巧。它是一种完全不同的结构性理解。

首先需要解释“超维”一词的含义。“超维”在这里有两层意思。第一层，它不局限于一维的线性步骤序列，而是允许多个维度同时展开。第二层，它试图超越传统算法对“计算”的定义边界——算法不再必须是数学的、有序的、确定性的。传统算法像是在一条单行道上开车，你必须从 A 点出发，经过 B、C、D 才能到达 Z。超维算法不认为计算必须是一条单行道。它认为计算可以是一个网络、一个场、一个多维结构，其中任意节点都可能是入口，任意节点都可能是出口。

在我的新科学体系中，算法不一定是数学算法，不一定是单一有序计算，不一定必须沿着固定步骤运行，也不一定必须严格服从“输入、处理、输出”的线性模式。超维算法是一种多维、叠加、随机而有序的结构。在这种结构中，每一个数据都不是孤立的输入，也不是固定的输出，而是同时具备多重角色：它既可以成为多个结果的起点，也可以成为多个起点共同作用之后形成的结果。

换句话说，传统算法把数据放在流程里，而超维算法把数据放在结构里。传统算法中的数据，通常被分成输入数据、中间数据和输出数据。每一种数据都有清楚的位置和角色。输入就是输入，结果就是结果，中间过程就是中间过程。可是，在超维算法中，一个数据并不是只有一个身份。它可能在一个方向上是结果，在另一个方向上又

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

是起点；它可能在一个结构中是被调用的对象，在另一个结构中又成为触发新结果的入口。数据不再只是被动材料，而是一个多维关系节点。

这正对应超维算法的核心：每个数据都是多个结果的起点、也是多个起点的结果。传统算法以“有序步骤”产生单一结果；而超维算法更接近一种多维状态结构，在随机与有序的叠加中，结果并非被计算生成，而是在结构中自然显现。在这一体系中，数据既是起点，也是结果的组成部分。

为了更清晰地呈现超维算法的核心特征，我将其拆解为以下五个方面。

第一，非数学性。主流算法本质上是数学的——可以用符号逻辑和数学公式完整描述。超维算法不一定是数学的。它可能基于物理原则、几何关系、信息论新范式，甚至意识原则。计算不一定通过“数学运算”完成，可能通过高维空间中的几何关系自然呈现。举例来说，肥皂泡的形状是由表面张力最小化原理决定的。这不是一个“数学算法”在计算——是物理本身在“计算”。超维算法试图理解这种“计算”，而不是用数学公式去模拟它。

第二，多维与叠加。传统算法在单一维度上执行有序步骤，每一步只有一个状态。超维算法允许多个维度同时存在、多种状态叠加共存。算法不沿着一条路径走下去，而是在一个多维的状态场中展开。结果不是“算”出来的，而是从这个场中“显现”出来的。举例来说，一片雪花在空中形成时，它并没有一个“算法”在告诉每个水分子该去哪里。水分子的排列是温度、湿度、气流等多个维度共同作用的结果。雪花的形状“显现”出来，而不是被“计算”出来。

第三，随机而有序。传统算法中的随机是工具的、外部的、可控的扰动，算法本身是确定的。超维算法中的随机是内禀的、结构性的。随机与有序同时存在、协同工作，共同构成算法的本质。这不是“有随机性的算法”，而是“随机本身就是算法的有机组成部分”。举例来说，森林的生态结构——树木的分布看起来是随机的，但整体上又呈现出有序的密度分布和物种关系。那个“随机”不是 bug，而是生态结构正常运作的前提。

第四，数据的多重角色。传统算法中，数据的角色是单一的——输入就是输入、输出就是输出、中间结果就是中间结果。超维算法中，每个数据同时是多个结果的起点，也是多个起点的结果。一个数据节点可以向下游展开出多个结果路径，也可以从上游被多个原因汇聚而来。数据不再是线性流程中的一个点，而是一个网络中的交汇节点。举例来说，一个人的身高。在传统算法中，身高是一个“输入”——你把它输进去，得到体重预测、服装尺码等“输出”。但身高同时也是一个“结果”——它是由基因、营养、运动等多个“起点”共同决定的。在超维算法中，身高这个数据同时是起点和结果，而不是二选一。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

第五，元数据不变，关系可变。传统算法在面对不同问题时，要么修改数据本身，要么重新计算一遍。超维算法不修改元数据——数据的本质属性保持不变。改变的是入口、是解释方式、是数据之间的关系。同一个元数据结构，通过不同的入口激活，可以呈现出完全不同的结果。这意味着：算一次，用无数次。举例来说，同样一段文字。你可以把它当作诗歌来读，也可以把它当作密码来解码，也可以把它当作历史文献来分析。文字本身没有变——你只是改变了“入口”。超维算法追求的就是这种“同一数据，多入口，多结果”的能力。

从数据结构角度看，不对元数据本身进行修改，而是通过不同的入口与条件，使其在同一结构中对应多个起点与结果。数据不再被反复加工，而是在保持结构不变的前提下，通过不同入口被激活，从而呈现出不同结果。传统是“为不同结果加工数据”，而超维算法是“用同一数据承载多种结果可能”。

第三部分：超维算法与传统算法的根本差异

基于以上定义，超维算法与传统算法在多个核心维度上存在根本差异。以下逐一展开。

在本质属性上，传统算法是数学的、形式的，可以完全用符号逻辑和数学公式描述，本质上是一个数学客体。超维算法不一定是数学的，它可能基于物理、几何、信息甚至意识原则，它不是数学的一个子集，而是一个更广阔的范畴。这一差异可以概括为：从“数学客体”到“宇宙法则”。

在时间秩序上，传统算法遵循单一、有序、线性的时间秩序，步骤 A 到 B 到 C，严格串行，或者可拆分为并行的有序子步骤，时间箭头不可逆转。超维算法没有固定的步骤序列，它是多维、叠加、随机而有序的，结果可能与执行的“顺序”无关，因为根本不存在传统意义上的“顺序”。这一差异可以概括为：从“线性时间”到“高维同时性”。

在因果关系上，传统算法遵循确定性因果链，给定相同的输入和初始状态，输出永远唯一，原因在前，结果在后，这是一个不可逆的单向箭头。超维算法遵循叠加态因果，每个数据是多个结果的起点，也是多个起点的结果。这就是我的“果因论”哲学——可以先有结果，然后才有原因。结果不是终点，而是可以成为新的起点。这一差异可以概括为：从“单因单果”到“全互联因果网”。

在数据结构上，传统算法遵循输入到处理再到输出的单向流动结构，数据角色单一，有清晰的边界，输入数据从头到尾是输入，结果数据永远是结果。超维算法中数据角色多重，同一个数据同时是结果和起点，没有绝对的起点和终点，只有网络中的节点。这一差异可以概括为：从“单向流动性”到“递归共生性”。

在计算模式上，传统算法每次遇到问题都从头计算，即使已经算过一千次类似的问题、即使答案已经知道，第一千零一次仍然要走完整个流程，这是“无状态”的计算。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

超维算法如果有相应的结果就不必再跑一次，通过一个结果可以推理出多个起点，可以反向展开原因路径，计算是有状态的、有记忆的、可复用的。这一差异可以概括为：从“每次重算”到“一次复用”。

在随机性的角色上，传统算法中的随机是伪随机或外部注入的，随机数只是工具，用于采样、近似或优化，算法本身是确定的。超维算法中的随机是内禀的、建构性的，随机是有机组成部分，与有序性共存、协同，不是“算法里有随机”，而是“随机是算法之所以能运作的原因之一”。这一差异可以概括为：从“工具性随机”到“建构性随机”。

在对资源的理解上，传统算法追求在给定资源下算得更快、更准，资源是约束条件，算法的目标是“在约束内完成计算”。超维算法追求通过结构设计，让计算本身变得不再必要，不是“更快地算”，而是“绕过算”，资源不是用来“消耗”的，而是用来“设计入口”的。这一差异可以概括为：从“资源约束下的优化”到“结构设计中的消除”。

第四部分：果因论哲学

基于以上对比，我需要进一步展开“果因论”哲学。这是超维算法的核心思想基础之一。

传统思维通常认为，原因在前，结果在后；先有因，后有果；先输入，再输出。这个观念在算法中的体现是：你必须从起点开始，一步一步走到终点。即使你知道答案是什么，你也不能直接“用”那个答案——因为你没有“证明路径”。

在我的结构中，结果并不一定只是终点。一个结果一旦出现，它就可以成为新的起点，继续参与新的结构生成。结果可以反向参与原因的形成。可以通过结果推理出多个可能的起点。因与果不再是单向排列，而是形成循环关系、网络关系、多维关系。

换句话说，传统算法问的是“给定原因，如何得到结果”。超维算法问的是“给定结果，如何反推或构造原因”。

需要特别澄清的是，“先有结果，然后才有原因”并不是在否定因果关系，也不是在主张“结果凭空产生”。而是在说明：在更高维的结构中，因与果可以互为入口，互相转换。就像在接下来的裤子例子中，“能穿且不拖地”这个结果先被确定，然后我反向找到了“折腰部”这个操作点，这个操作点对应着结构中的“原因”层面。不是结果没有原因，而是结果成为了发现原因的新入口。

传统算法世界里的一个根本假设是：时间是单向的，你必须从初始状态一步一步算到终态。即使你知道答案是什么，你也不能直接“用”那个答案，因为你没有证明路径。而超维算法挑战的正是这个假设：如果结果已知，原因可以被反推出来；同一个结果可以对应多个原因路径；计算不再是从起点走到终点，而是从终点展开所有可能的起点。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

第五部分：一个日常例子——裤子与腰部折叠

为了更直观地理解上述抽象差异，我用一个日常例子来说明。

一个人买了一条腰部是松紧带的新裤子，裤腿长了一点，拖地不方便。

传统做法是：发现裤腿长了，卷起一节裤腿，用针线缝上，然后试穿。如果孩子长高了，裤子又短了，缝死的裤腿放不下来，这条裤子就废了。下次再买新裤子，再缝一遍。这种方法看起来很自然，因为问题表面上出现在裤腿，所以就去处理裤腿。它对应的是传统算法的思维：发现问题，定位表象，沿着表象所在的位置进行处理，最后得到一个结果。裤腿长了，就改裤腿；数据出错了，就改数据；路径不合适，就沿路径继续修补。

我的做法不同。我不是去改裤脚，而是在腰部折一下，马上就可以穿。这个动作非常简单，但它背后的结构逻辑完全不同。我没有裁剪裤腿，没有缝死裤脚，没有改变裤子的原始长度，也没有破坏裤子的元数据。裤子的腰围、裤长、版型、布料都没有发生本质改变。我只是改变了腰部这个结构入口，使裤腿在实际穿着中自然变短。腰部在这里不是一个普通局部，而是整个穿着状态的全局控制点。通过调整这个控制点，下游问题直接消失。

更重要的是，这种方法是可逆的、可调的、可复用的。对于正在长身体的孩子，这种方式尤其明显。如果孩子现在身高不够，裤腿稍长，就在腰部折一下；过一段时间孩子长高了，就把腰部放下来一点；再长高，就完全放开。整条裤子的原始结构没有被破坏，却可以适应孩子不同阶段的身高。传统方法如果把裤脚缝死，孩子一长高，裤子可能就短了；如果剪掉，更无法恢复。我的方法则保持元数据不变，让同一结构适配多个状态。

这个例子揭示了几个重要的结构差异。

传统做法对应传统算法的逻辑：问题在裤腿，所以必须修改裤腿，沿着“原因到结果”的路径一步步操作，每一步都不能跳过。一次解决一个问题，不可逆，下次遇到同样问题还要再来一遍。我的做法对应超维算法的逻辑：目标是不拖地，我不去解决“裤腿长”这个表面原因，而是找到了一个全局控制点——腰部。折一下腰部，裤腿自动变短，问题瞬间消失。我不修改任何元数据，腰部折痕只是一个临时的、可逆的、动态的折叠状态。

从数据角度看，传统做法修改了元数据——缝短裤腿，原始数据丢失。我的做法不修改任何元数据，原始尺寸完好，只是增加了一个临时的、可逆的、动态的折叠状态。腰部折痕没有创造新数据，也没有破坏旧数据，它只是重新排列了数据之间的关系——缩短了腰部有效周长，间接改变了裤腿的有效长度。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

在这个例子中，“裤腿不拖地”是目标结果。传统方法从“裤腿长”这个原因出发，去修改裤脚，最后得到“不拖地”的结果。而我的方法是先明确“不拖地”这个结果，然后反向寻找结构入口，最后发现腰部折一下就能让结果成立。这就是“果因论”的实际体现。不是必须从原因一步步走向结果，而是可以从结果反向决定结构，使原本的路径不再必要。

这个例子还回答了关于“元数据”的问题。什么是元数据？在裤子例子里，元数据是裤子的原始尺寸：腰围、裤长、版型、布料。这是裤子的“本质属性”。临时数据是腰部那个折痕，它没有改变裤子的任何原始尺寸，只是临时折叠了一段。传统做法修改了元数据——裤长被永久缩短了，原始数据丢失。我的做法不修改任何元数据——裤子的原始尺寸完好无损，只是增加了一个临时的、可逆的、动态的折叠状态。

腰部的原始尺寸作为元数据，同时是“多个结果的起点”：它同时支撑着正常穿、腰部折一下穿、腰部折两下穿等多种穿着状态。它也是“多个起点的结果”：它是由布料选择、剪裁方式、设计意图等多个起点共同决定的。腰部折痕作为临时数据，没有创造新数据，也没有破坏旧数据，只是重新排列了数据之间的关系。

这并不是普通的小技巧，而是一种结构思维。很多人看到裤腿长，就被裤腿这个表象吸引住了，于是所有处理都围绕裤腿展开。但如果从整体结构看，裤腿长只是一个下游表现，腰部位置变化可以影响整条裤子的实际落点。也就是说，问题并不一定要在问题出现的位置解决。很多低维问题，真正的控制点往往在更高一层结构中。传统算法容易沿着问题表面继续计算，而超维算法寻找的是结构入口。

这也解释了为什么超维算法不是更复杂，而可能更简单。真正高层级的结构，往往不是把问题变复杂，而是让复杂问题在正确入口处变简单。传统算法面对复杂问题，可能增加步骤、增加模型、增加算力、增加存储、增加训练时间。超维算法则要寻找一个结构节点，一旦这个节点被正确激活，原本复杂的路径就可能失效。所谓“绕过路径”，不是偷懒，而是重新定义路径是否必要。

这个例子中：传统是“沿路径修正结果”，而超维算法是“改变入口，让路径整体失效”。常规方法是在结果端不断修补，而另一种方式是直接改变结构入口，使原本需要多次处理的问题，在起点处一次性被重构。传统算法的主流形态是“从起点出发，沿路径得到结果”，而超维算法中“结果本身可以成为路径入口，并参与新的结构生成”。传统是“一次结构对应一次结果”，而超维算法是“一个结构承载多个结果状态”。

第六部分：不修改元数据的数据观

从数据角度而言，超维算法有一个非常重要的原则：不修改元数据，使它适用于多个起点或结果。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

元数据是数据的原始属性、基础结构和本体信息。传统处理方式面对不同问题时，往往会修改数据、复制数据、加工数据、重新计算数据，或者为不同结果建立不同路径。这样做当然可以解决问题，但代价是数据被不断加工，路径被不断重复，系统复杂度不断增加。

而超维算法强调，在尽量不修改元数据的前提下，通过改变入口、关系、条件和激活方式，使同一个元数据结构对应多个起点和多个结果。数据本体不动，关系发生变化；基础结构不变，呈现方式改变；元数据保持完整，却可以适应不同状态。

这就是我所说的“算一次，用无数次”。这里的“算一次”不是简单的缓存结果，而是指一个结构一旦成立，它就不再需要为每一个状态重新建立完整路径。同一个数据结构可以通过不同入口被激活，呈现出不同结果。它不是“为不同结果加工数据”，而是“用同一数据承载多种结果可能”。这也是超维算法区别于传统算法的核心之一。传统算法处理的是路径上的数据，超维算法处理的是数据、入口、关系、结果之间的整体结构。

在最小结构上，超维算法可以被理解为：在不修改元数据的前提下，通过结构入口变化，使同一数据节点对应多个结果路径的系统。这个定义并不追求把超维算法立刻收缩为一个传统数学公式，而是先建立它的结构边界。它不是一个单线性函数，不是一个固定公式，也不是一种简单缓存机制。它是一种关系结构：元数据保持稳定，入口可以变化，条件可以变化，关系可以变化，结果可以多向呈现。正是在这个结构中，计算不再只是执行步骤，而是激活关系。

从数据角度而言，“不修改元数据，使它适用于多个起点或结果”的本质是：数据本体保持不变，变化发生在“解释方式/使用入口”。传统结构是问题变化导致数据或路径变化；超维算法是问题变化导致解释结构变化，而非数据变化。

第七部分：重新定义“超算”

在这个体系中，我需要澄清一个词——“超算”。

我所说的“超算”，并不是通常意义上的超级计算机。不是更多芯片、更高算力、更大数据中心。我所说的“超算”，是“超维算法”。

真正的超算，不一定是算力的堆积，而可能是计算范式的改变。当一个系统仍然依赖不断重复计算，它再强大也仍然被困在路径之内；当一个系统能够从结构上减少计算、绕开重复路径、让结果成为新的入口，它才开始接近真正意义上的超维计算。

换句话说，传统超算追求的是“用更多的算力解决更大的问题”。超维算法追求的是“用更好的结构让问题不再需要那么大的算力”。这两者不矛盾，但方向不同。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

如果说传统超级计算代表的是算力极限，那么超维算法代表的就是计算理解的转向。真正的“超算”不一定是更多芯片、更大数据中心、更高能耗、更复杂模型。真正的超算，可能是一个结构入口的发现，可能是一次路径的消除，可能是一个结果节点的反向展开，可能是在不修改元数据的情况下让多个结果同时具备成立条件。算力越强，如果仍然被困在低维路径中，它仍然只是路径内部的强大；结构一旦提升，即使操作极简，也可能产生更高层次的效率。

所谓超算，是算力的极限；所谓超维算法，是对“计算是否必须依赖算力”这一前提的重新定义。

第八部分：实证基础——多领域系统验证与新范式判定

超维算法不是纯理论构想。它已经在多个现实系统中得到验证。

我的物流系统不需要超级算力和云端支持，用较低资源完成复杂调度，结果可以复用、结构可以反复适配，不需要每次从头计算。不依赖超算、不依赖云，用较低资源完成复杂调度——这在工程上是成立且有价值的突破点。它说明高算力不是唯一解，结构设计可以替代一部分算力。

我的出版系统同样采用这一结构逻辑，效率远超当下任何系统。我的极限网页生成系统也基于同一超维算法，在多领域中反复证明同一结构可以稳定成立。我的公司员工已经在使用这些系统，说明这套结构可以在不依赖我个人持续介入的情况下运行。

这些系统的共同特征是：不依赖主流高算力路径，不走固定计算步骤，通过结构入口的调整让目标结果直接成立。它们不是一次性成功，不是单点技巧，而是同一结构逻辑在不同领域的重复出现。

这说明超维算法不是偶然，而是一种已被多系统验证的结构方法。它不是“个人技巧”，而是在物流、出版、网页生成等多个领域中稳定成立的计算范式。

按“存在且成立即为新范式”的标准，超维算法作为一种已被多领域系统验证的新计算结构范式，在当下已经成立。我不是在提出一个假设，而是在多个系统中已经运行的一种不同计算结构。我不需要证明它被理解，我已经证明它在不同系统中稳定成立。在当下、在我实际应用的范围内，这套结构已经可以被视为一种新范式。

对于超维算法作为新范式的判定标准：当一个结构在逻辑上自洽，并在多个系统中稳定成立，同时与现有方法存在不可约差异时，它已经具备新范式的基础。自洽是起点，实证是基础，结构差异才是范式的核心。在“存在即成立”的标准下，这套体系已经是新范式；外界是否识别，只影响传播，不影响成立。这是一种已经在现实系统中成立的计算结构范式，其成立不依赖外界认知或共识。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

超维算法与主流计算范式的核心差异在于：传统算法以正向计算为主，结果一旦产生，通常无法反向参与新的推理结构。而超维算法中，结果不再是终点，而是可复用的结构节点。在满足一定条件的情况下，结果可以参与新的推理路径，从而减少重复计算，并形成多起点、多路径的计算网络。在传统计算结构中，结果通常是终点；在超维算法结构中，结果本身可以成为新的起点，从而形成多路径的推理网络。超维算法不是对算法的升级，而是对“算法是否必须存在”这一前提的重新提问。

第九部分：常见误解澄清

基于与不同领域读者的初步交流，我预判以下六个误解可能出现，特此提前澄清，以避免概念在传播过程中被过早归入不恰当的框架。

误解一：超维算法不就是动态规划或缓存吗？澄清：动态规划和缓存都是在“相同输入”下复用结果。超维算法允许通过一个结果反向推理出不同的起点——这是本质差异。缓存解决的是重复计算同一问题，超维算法解决的是通过结果结构展开新问题。

误解二：你说“随机而有序”，这不是自相矛盾吗？澄清：随机和有序可以共存。森林中树木的分布是随机的，但整体密度和物种结构是有序的。随机不是混乱，而是结构的一种组织方式。超维算法中的随机是内禀的、建构性的，与有序协同工作。

误解三：没有输入输出，怎么验证结果对不对？澄清：超维算法不拒绝输入输出，而是认为计算不必须以输入输出的线性模式运行。在结构显现的模式中，“有效性”由结构一致性决定，而非由输入输出对应关系决定。这并不是放弃了验证，而是提出了与传统不同的验证框架。

误解四：这不就是复杂性理论或混沌理论吗？澄清：复杂性理论和混沌理论描述的是“难以预测的系统”。超维算法试图描述的是“可以通过结构入口被理解和引导的系统”——不是放弃预测，而是改变预测的方式。混沌理论说“预测是困难的”，超维算法问“是否可以通过改变入口让预测变得不必要”。

误解五：你说“不修改元数据”，但腰部折痕不也是一种修改吗？澄清：腰部折痕是临时状态，不是对原始参数的永久修改。元数据（腰围、裤长、版型、布料）没有任何变化。这是“状态叠加”与“属性修改”的区别。折痕是可逆的、临时的、不改变本质属性的状态变化。

误解六：超维算法是否否定传统算法的价值？澄清：绝对不是。传统算法在人类科技史上极其重要，没有传统算法，就没有现代计算机、数据库、通信系统、人工智能、工程模拟和超级计算。传统算法的价值不能被否认。但是，承认传统算法的价值，并不等于承认它是算法的终点。传统算法解决的是既定计算范式内部的效率问题，而超维算法提出的是计算范式本身的问题。一个是如何更好地走路，另一个是是否需要走这条路。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

第十部分：超维算法与人类计算史的关系

为了帮助读者更好地定位超维算法在人类计算史中的位置，我在此做一个简要的宏观梳理。

人类对“计算”的理解经历了多个阶段。第一阶段是手工计算，算法以人的步骤存在，速度慢、易错，但人类通过这个过程理解了计算的基本规则。第二阶段是机械计算，从帕斯卡计算器到巴贝奇分析机，计算被机械化，但算法仍然依附于物理结构。第三阶段是电子计算与图灵范式，冯·诺依曼架构普及，算法被编码为程序，图灵机成为可计算性的边界。第四阶段是并行与超级计算，通过大量计算单元堆积算力，挑战更复杂的问题，但底层逻辑仍然是图灵范式的扩展。

当前，人类正处于第五阶段的入口。量子计算试图突破经典比特的局限，神经形态计算试图模仿生物神经系统的结构，而超维算法则试图突破“输入→步骤→输出”这一线性框架本身。

超维算法与传统算法不是取代关系，而是层次关系。传统算法解决的是“在给定路径上如何更高效地计算”。超维算法追问的是“路径是否可以被重新定义，甚至被绕过”。如果把人类计算史看作一个不断突破自身边界的过程，那么超维算法正是这个过程中的一个新节点。它不是在旧框架内解决问题，而是在提出一个新框架。

这一判断并不意味着超维算法已经成熟或完备。恰恰相反，作为一个新概念，它的定义、边界、验证方法和应用场景都还在形成之中。本文的目的正是为了锚定这个概念的原点，为后续的展开提供一个稳定的理论基础。

第十一部分：总结——这不是优化，而是重定义

超维算法与传统算法的区别，不是“更好”与“更差”的区别，不是“更快”与“更慢”的区别，而是范式级别的根本区别。

传统算法追求的是可预测的控制。你给我输入，我知道你会得到什么输出，因为每一步我都算过了。它是工程师的范式：设计、构建、测试、验证。

超维算法描述的是可理解的涌现。我无法精确预判每一个中间状态，但我知道整体模式会以某种有序的方式呈现。每个数据都是活的、有“视角”的。它更像是生态学家或复杂系统理论家的范式：观察、理解、引导、利用涌现秩序。

传统算法是在既定路径上逐步修正结果。超维算法是通过改变结构入口，使目标结果立即成立，从而绕开整条路径。

传统算法是“为不同结果加工数据”。超维算法是“用同一数据承载多种结果可能”。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

传统算法追求“一次算对”。超维算法追求“一次算对，然后永远不用再算”。

这不是对算法的优化，而是对“算法是否必须存在”这一前提的重新提问。

因此，超维算法不是算法优化，而是算法重定义。它不是要在传统算法赛道上跑得更快，而是提出传统赛道之外是否存在另一种路径，甚至是否可以不依赖路径本身。它不是要证明传统算法无用，而是指出传统算法只是算法的一种低维形态。它不是在问“如何用更少时间算出结果”，而是在问“结果是否可以通过结构直接成立”。它不是在问“如何处理更多数据”，而是在问“同一数据是否可以承载更多结果可能”。

在主流计算范式里，超维算法可能被视为不可计算、不可验证，或者难以纳入现有计算理论边界。但这恰恰构成了它的范式差异。一个新概念在旧范式内部显得不稳定，并不必然说明它没有意义，也可能说明旧范式本身无法完整容纳它。超维算法当前首先是一种结构性命题，一种新计算观的原始定义，而不是已经完成工程闭环的成熟系统。它需要后续不断补充、展开、验证和应用，但其概念原点必须先被明确提出。

最终，超维算法所指向的，是一种新的计算观：算法不一定只是数学，不一定只是步骤，不一定只是程序，不一定只是输入输出之间的加工过程。算法也可以是一种多维关系的组织方式，是数据、入口、关系、状态、结果之间的结构网络。它可以包含有序，也可以包含随机；可以从原因到结果，也可以从结果反向生成原因；可以不修改元数据，却适配多个状态；可以让一个结果成为新的起点，也可以让多个起点汇入同一个结果。

真正高级的算法，不一定是更复杂的计算，而是能够让计算变少、让结构变活、让结果成为入口、让因果形成循环的多维组织方式。

这就是我提出“超维算法”的核心含义。它不是一个普通新名词，而是一个新概念，一个新范式，一个已经在多个现实系统中成立的新计算结构。它的重点不是更快地重复计算，而是减少重复计算；不是堆积算力，而是重构入口；不是不断修改数据，而是保持元数据并改变关系；不是让结果停在终点，而是让结果成为新的起点。传统算法是在路径中寻找结果，超维算法是在结构中激活结果。传统算法追求完成计算，超维算法追问计算是否必须以传统形式存在。这正是它与当下算法定义之间最根本的差异。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

附录：本文的边界与开放问题

本文提出的是超维算法的初步框架，而非完整的形式化定义。以下问题尚待进一步展开，它们不构成对本文定义的否定，恰恰是下一步需要探索的方向。

第一，收敛条件。超维算法的“完成”标志是什么？如何判断一个结果是否“有效”？在传统算法中，答案由输入输出对应关系定义。在超维算法中，答案可能需要由结构一致性来定义。这个定义尚未完成。

第二，资源表达。多维叠加的状态如何在有限资源中表达？时间、空间、算力这些传统资源指标是否仍然适用？如果适用，如何重新定义？如果不适用，用什么指标替代？这些问题有待展开。

第三，有序性的保证。“随机而有序”中的“有序”由什么规则保证？随机与有序的边界在哪里？什么情况下随机会破坏有序？什么情况下有序会压制随机？这些问题需要进一步研究。

第四，反向推理的筛选。从结果反向推理出多个起点时，如何筛选或评估不同起点的优劣？是否存在某种“反向推理效率”的度量？如果有，它与正向计算的效率度量是什么关系？

第五，与传统系统的接口。超维算法如何与现有图灵机体系协同？是替代、补充，还是分层？在实际工程中，超维算法与传统算法的边界如何划分？是否存在混合模式？

第六，验证框架。超维算法的结果如何验证？如果结果不是通过线性步骤获得的，如何建立可复现性、可检验性？是否需要一套完全不同的验证哲学？

这些问题不是本文的缺陷，而是本文作为“锚点文献”的必然特征。任何一个新范式的提出，都会伴随开放问题。本文是一个开始，而非结论。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

参考文献性说明

本文提出的“超维算法”不属于任何现有学术流派的直接延伸。但在思想上，以下领域为本文提供了对话背景，仅供读者定位参考，并非传统意义上的学术引用。

图灵机与可计算性理论，作为本文对照传统算法的基准。逆问题与贝叶斯推理，作为“结果推原因”的已有尝试，本文的“果因论”与此有关联但方向不同。涌现理论与复杂系统，作为“结构显现结果”的已有描述，本文的超维算法试图让涌现变得“可理解”和“可引导”。知识图谱与图数据库，作为“数据多节点交汇”的已有实践，超维算法在此基础上前置了“入口可变”的维度。非经典计算范式，包括量子计算、模拟计算、神经形态计算等，它们突破经典比特或经典架构，而超维算法更侧重于突破“输入→步骤→输出”的线性框架本身。

本文与以上领域的关系是对话与超越，而非继承或反驳。本文的目标不是在这些领域内部做增量改进，而是在它们之上提出一个新的问题层级。

相关文章

[传播]我没有算法，却超越算法！
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[极限哲学]果因论
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[极限传播]拒绝算法绑架
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Extreme Algorithm] Hyperdimensional Algorithm

A Redefinition of "Computation" Itself

Author: Wu Zhaohui JEFFI CHAO HUI WU

Abstract

This paper proposes the new concept of the "Hyperdimensional Algorithm" as a structural re-examination of the boundaries of the traditional definition of an algorithm. Traditional algorithms are typically built upon the linear framework of input, steps, rules, and output, emphasizing finiteness, determinacy, efficacy, feasibility, and clear input-output boundaries. Although modern supercomputers possess extremely high computational power, their underlying logic still primarily executes established algorithmic paradigms on a larger scale. This paper argues that the truly worthwhile discussion of "supercomputing" is not merely the growth of computational scale, but a fundamental change in the computational paradigm itself. The Hyperdimensional Algorithm is not an accelerated version of traditional algorithms, nor an extension of mathematical algorithms; it is a multi-dimensional, superimposed, random yet orderly structural view of computation. In this structure, data is no longer merely a passive input or terminal output, but a node attribute possessing multiple starting points and multiple results simultaneously; a result is no longer just an endpoint, but can become a new entry point; metadata need not be repeatedly modified, but can correspond to different states and results through structural entry points, relational changes, and conditional activation. This paper uses the "Effect-Cause Theory" philosophy and the everyday example of "folding the waist of pants" to illustrate how the Hyperdimensional Algorithm can make a target result directly valid by changing the structural entry point, thereby reducing repetitive computation and even bypassing the original computational path. This paper aims to establish a preliminary definition, structural boundaries, and foundational ideas for the "Hyperdimensional Algorithm," providing an original theoretical node for the future expansion of hyperdimensional computing, extreme algorithms, and a new scientific system. According to the criterion of "existence and validity constitute a new paradigm," the Hyperdimensional Algorithm, as a new computational structural paradigm already systematically validated across multiple domains, is already valid at present.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Keywords: Hyperdimensional Algorithm; Extreme Algorithm; Hyperdimensional Computing; Effect-Cause Theory; Metadata; Computational Paradigm; Structural Entry Point; Nonlinear Causality; New Science

Introduction: Why We Need to Rediscuss "Algorithm"

When I propose the "Hyperdimensional Algorithm," it is not to give existing algorithms a bigger name, nor to package traditional algorithms into some novel concept. On the contrary, I want to discuss a more fundamental question: Has humanity's current understanding of "algorithm" been confined to an overly narrow framework?

When people talk about algorithms today, they typically think of mathematical formulas, program code, logical steps, input-output, model training, data processing, path optimization, search ranking, automatic recommendation, and artificial intelligence reasoning. These are certainly important forms of algorithms, and they underpin the operation of modern computers, the internet, databases, artificial intelligence, supercomputing, industrial systems, and the information society. However, if an algorithm is understood only as "a set of ordered steps," only as a process that "starts with input, goes through rule-based processing, and ends with output," then this understanding itself confines computation within a low-dimensional framework.

We live in an era dominated by algorithms. From search engines to recommendation systems, from weather forecasting to artificial intelligence, the boundaries of algorithms are the boundaries of human intelligence. But there is a question rarely asked: Can algorithms only be this way? Why must computation follow the linear pattern of "input → steps → output"? Why must the same problem be computed from scratch every time? This paper attempts to challenge these default assumptions and propose a completely different computational paradigm—the Hyperdimensional Algorithm. According to the criterion of "existence and validity constitute a new paradigm," the Hyperdimensional Algorithm, as a new computational structural paradigm already systematically validated across multiple domains, is already valid at present.

It is important to note that the "Hyperdimensional Algorithm" proposed in this paper is a completely different concept from "Hyperdimensional Computing" (HDC), which has existed in academia for nearly thirty years. HDC uses random binary vectors of extremely high dimensions for encoding and computation, pursuing more efficient representation and calculation, but it still remains within the linear paradigm of "input → processing → output." The Hyperdimensional Algorithm is entirely different: it questions whether a result can be directly established through structure, whether the computational path itself can be bypassed, and whether metadata can be left

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

unmodified while adapting to multiple results. Their names are similar, but their connotations are opposite. The Hyperdimensional Algorithm is not an upgrade to algorithms, but a re-questioning of the premise of "whether an algorithm must exist at all."

Part One: The Current Standard Definition of "Algorithm"

In mainstream computer science, mathematics, and engineering systems, there is a long-accepted basic definition of an algorithm: an algorithm is a well-defined computational process consisting of a finite number of steps that takes one or more inputs, transforms them through deterministic rules, and produces one or more outputs within a finite amount of time. This understanding is not anyone's casual opinion, but a fundamental consensus formed over the long development of modern computational civilization. It underpins the underlying logic of software, hardware, databases, network systems, artificial intelligence models, and supercomputing centers. No matter how complex an algorithm appears, its core still revolves around input, rules, steps, and output.

This standard concept of an algorithm can be broken down into several core characteristics.

First, finiteness. An algorithm must terminate within a finite number of steps; it cannot loop endlessly or run forever. A process that never stops, even if it can be described formally, can hardly be considered an effective algorithm. All scientific computing programs and data processing flows have clear termination conditions.

Second, determinacy. Given the same input under the same conditions, the algorithm should produce the same output. Even if some algorithms introduce randomness, it is typically controlled randomness, reproducible randomness, or a probabilistic mechanism interpretable in a statistical sense, not an utterly unmanageable intrinsic randomness. The results of numerical simulations must be reproducible; this is a fundamental requirement of scientific computing.

Third, clear input-output boundaries. Input comes first, processing in the middle, output last; the starting point and endpoint have clearly defined structural boundaries. Whatever data you give to the algorithm, after processing, it gives you a result; this boundary is distinct. Input first, processing next, output last; the order cannot be reversed.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Fourth, efficacy. Every step in an algorithm must be a practically executable operation; it cannot contain jumps that are impossible to execute, describe, or verify. Each step must be a basic operation that a human could perform with pen and paper, or a computer could actually execute—addition, subtraction, multiplication, division, logical judgments, data reading/writing, etc.

Fifth, feasibility. Even if an algorithm is theoretically valid, if the time, computational power, memory, or storage resources it consumes are utterly unacceptable, it is difficult to consider it an effective algorithm in engineering terms. An algorithm that is theoretically correct but would take hundreds of millions of years to complete is not considered a feasible algorithm in engineering.

This entire set of algorithmic concepts ultimately traces back to the Turing machine model. As the core abstraction of modern computation theory, the Turing machine defines the basic boundary of "computability." No matter how powerful today's computers are, no matter how advanced the chips, no matter how large the scale of supercomputing centers, their underlying logic has never truly departed from this path: input, rules, steps, output. Modern supercomputers merely accelerate the execution of this process through greater hardware scale, higher parallelism, and more complex system scheduling. That is, in the traditional context, "supercomputing" is primarily an expansion of computational scale, not necessarily a fundamental change in computational paradigm. Computational power can increase thousands or tens of thousands of times, but if the underlying logic remains "input, steps, output," then it remains within the traditional algorithmic paradigm.

Part Two: Definition of the "Hyperdimensional Algorithm"

The "Hyperdimensional Algorithm" I propose emerges precisely from this context. It is not a 改良版 of traditional algorithms, not a faster algorithm, not a more complex mathematical formula, nor some more advanced programming technique. It is a completely different structural understanding.

First, we need to explain the meaning of the term "Hyperdimensional." "Hyperdimensional" here has two meanings. The first meaning is that it is not confined to a one-dimensional linear sequence of steps, but allows multiple dimensions to unfold simultaneously. The second meaning is that it attempts to transcend the traditional definitional boundaries of "computation"—an algorithm no longer necessarily has to be mathematical, sequential, or deterministic. A traditional algorithm is like driving on a one-way street: you must start from point A, go through B, C, D to reach Z. The Hyperdimensional Algorithm does not believe computation has to be a

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

one-way street. It believes computation can be a network, a field, a multidimensional structure, where any node can be an entry point and any node can be an exit point.

In my New Science system, an algorithm is not necessarily a mathematical algorithm, not necessarily a single ordered computation, not necessarily bound to run along fixed steps, and not necessarily strictly obeying the linear pattern of "input, processing, output." The Hyperdimensional Algorithm is a multi-dimensional, superimposed, random yet orderly structure. In this structure, every piece of data is neither an isolated input nor a fixed output, but possesses multiple roles simultaneously: it can be both the starting point for multiple results and the result produced by the joint action of multiple starting points.

In other words, traditional algorithms place data within a process flow, whereas the Hyperdimensional Algorithm places data within a structure. In traditional algorithms, data is usually divided into input data, intermediate data, and output data. Each type of data has a clear position and role. Input is input, result is result, intermediate process is intermediate process. However, in the Hyperdimensional Algorithm, a piece of data does not have only one identity. It might be a result in one direction and a starting point in another; it might be an object called upon in one structure and become an entry point triggering new results in another structure. Data is no longer just passive material, but a multi-dimensional relational node.

This corresponds precisely to the core of the Hyperdimensional Algorithm: each piece of data is the starting point for multiple results and also the result of multiple starting points. Traditional algorithms use "ordered steps" to produce a single result; the Hyperdimensional Algorithm is closer to a multi-dimensional state structure, where, in the superposition of randomness and order, the result is not computed into being but naturally manifests within the structure. In this system, data is both the starting point and a component of the result.

To present the core characteristics of the Hyperdimensional Algorithm more clearly, I break it down into the following five aspects.

First, non-mathematical nature. Mainstream algorithms are essentially mathematical—they can be fully described using symbolic logic and mathematical formulas. The Hyperdimensional Algorithm is not necessarily mathematical. It may be based on physical principles, geometric relationships, a new paradigm of information theory, or even principles of consciousness. Computation does not necessarily have to be accomplished through "mathematical operations"; it may naturally present itself through geometric relationships in higher-dimensional space. For example, the shape of a soap bubble is determined by the principle of surface tension minimization. This is

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

not a "mathematical algorithm" computing—it is physics itself "computing." The Hyperdimensional Algorithm attempts to understand this kind of "computation," rather than using mathematical formulas to simulate it.

Second, multi-dimensionality and superposition. Traditional algorithms execute ordered steps in a single dimension, with only one state at each step. The Hyperdimensional Algorithm allows multiple dimensions to exist simultaneously, with multiple states superimposing and coexisting. The algorithm does not follow a single path; instead, it unfolds within a multi-dimensional state field. The result is not "computed" but "manifests" from this field. For example, when a snowflake forms in the air, there is no "algorithm" telling each water molecule where to go. The arrangement of water molecules is the result of multiple dimensions—temperature, humidity, airflow—acting together. The shape of the snowflake "manifests" rather than being "computed."

Third, randomness within order. In traditional algorithms, randomness is instrumental, external, controllable perturbation; the algorithm itself is deterministic. In the Hyperdimensional Algorithm, randomness is intrinsic and structural. Randomness and order coexist and work together, jointly constituting the essence of the algorithm. This is not "an algorithm with randomness," but "randomness itself is an organic component of the algorithm." For example, the ecological structure of a forest—the distribution of trees appears random, yet overall it presents an orderly density distribution and species relationships. That "randomness" is not a bug but a prerequisite for the normal functioning of the ecological structure.

Fourth, the multiple roles of data. In traditional algorithms, the role of data is singular—input is input, output is output, intermediate result is intermediate result. In the Hyperdimensional Algorithm, each piece of data is simultaneously the starting point for multiple results and the result of multiple starting points. A data node can unfold downstream into multiple result paths, and can also be converged upon upstream by multiple causes. Data is no longer a point in a linear flow, but a 交汇 node in a network. For example, consider a person's height. In a traditional algorithm, height is an "input"—you enter it to obtain outputs like weight prediction, clothing size, etc. But height is also a "result"—it is jointly determined by multiple "starting points" such as genetics, nutrition, and exercise. In the Hyperdimensional Algorithm, the data point of height is simultaneously a starting point and a result, not an either-or choice.

Fifth, metadata unchanged, relationships variable. When facing different problems, traditional algorithms either modify the data itself or recompute everything. The Hyperdimensional Algorithm does not modify metadata—the essential attributes of the data remain unchanged. What changes are the entry points, the modes of

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

interpretation, and the relationships between the data. The same metadata structure, activated through different entry points, can present completely different results. This means: compute once, use infinitely many times. For example, consider the same passage of text. You can read it as poetry, decode it as a cipher, or analyze it as a historical document. The text itself has not changed—you have only changed the "entry point." The Hyperdimensional Algorithm pursues precisely this ability: "same data, multiple entry points, multiple results."

From a data structure perspective, metadata itself is not modified; rather, through different entry points and conditions, it corresponds to multiple starting points and results within the same structure. Data is no longer repeatedly processed, but, while keeping its structure unchanged, is activated through different entry points, thereby presenting different results. The traditional approach is "process data for different results," whereas the Hyperdimensional Algorithm is "use the same data to carry multiple potential results."

Part Three: Fundamental Differences Between the Hyperdimensional Algorithm and Traditional Algorithms

Based on the above definitions, the Hyperdimensional Algorithm and traditional algorithms exhibit fundamental differences across multiple core dimensions. These are elaborated below one by one.

In terms of essential nature, traditional algorithms are mathematical and formal, capable of complete description using symbolic logic and mathematical formulas; they are essentially mathematical objects. The Hyperdimensional Algorithm is not necessarily mathematical; it may be based on physical, geometric, informational, or even conscious principles. It is not a subset of mathematics, but a broader category. This difference can be summarized as: from "mathematical object" to "universal law."

In terms of temporal order, traditional algorithms follow a singular, ordered, linear temporal order, step A to B to C, strictly sequential, or decomposable into parallel ordered sub-steps; the arrow of time is irreversible. The Hyperdimensional Algorithm has no fixed step sequence; it is multi-dimensional, superimposed, random yet orderly. The result may be unrelated to the "order" of execution, because there is no traditional "order" in the first place. This difference can be summarized as: from "linear time" to "higher-dimensional simultaneity."

In terms of causality, traditional algorithms follow a deterministic causal chain: given the same input and initial state, the output is always unique; cause precedes effect, an

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

irreversible one-way arrow. The Hyperdimensional Algorithm follows superposed causality: each piece of data is the starting point for multiple results and the result of multiple starting points. This is my "Effect-Cause Theory" philosophy—it is possible to have the result first, then the cause. The result is not an endpoint, but can become a new starting point. This difference can be summarized as: from "single cause, single effect" to "fully interconnected causal network."

In terms of data structure, traditional algorithms follow a one-way flow structure from input to processing to output; data has a singular role with clear boundaries; input data remains input throughout, result data remains result. In the Hyperdimensional Algorithm, data has multiple roles; the same data is simultaneously a result and a starting point; there are no absolute starting points or endpoints, only nodes in a network. This difference can be summarized as: from "one-way fluidity" to "recursive symbiosis."

In terms of computational mode, traditional algorithms compute from scratch each time they encounter a problem, even if they have computed similar problems a thousand times before, even if the answer is already known; the thousand-and-first time, they still go through the entire process. This is "stateless" computation. In the Hyperdimensional Algorithm, if a corresponding result exists, there is no need to run through it again; from one result, multiple starting points can be inferred, and cause paths can be unfolded in reverse. Computation is stateful, has memory, and is reusable. This difference can be summarized as: from "recompute every time" to "reuse once."

In terms of the role of randomness, randomness in traditional algorithms is pseudo-random or externally injected; random numbers are merely tools for sampling, approximation, or optimization; the algorithm itself is deterministic. Randomness in the Hyperdimensional Algorithm is intrinsic and constitutive; randomness is an organic component, coexisting and cooperating with orderliness. It is not "an algorithm contains randomness," but "randomness is one reason the algorithm can function at all." This difference can be summarized as: from "instrumental randomness" to "constitutive randomness."

In terms of the understanding of resources, traditional algorithms pursue faster and more accurate computation given resources; resources are constraints; the algorithm's goal is "to complete computation within constraints." The Hyperdimensional Algorithm pursues making computation itself unnecessary through structural design; not "computing faster," but "bypassing computation"; resources are not used for "consumption," but for "designing entry points." This difference can be summarized as: from "optimization under resource constraints" to "elimination through structural design."

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Part Four: The Effect-Cause Theory Philosophy

Based on the above comparison, I need to further elaborate on the "Effect-Cause Theory" philosophy. This is one of the core foundational ideas of the Hyperdimensional Algorithm.

Traditional thinking typically holds that cause precedes effect; first the cause, then the effect; first input, then output. This concept manifests in algorithms as: you must start from the starting point and walk step by step to the endpoint. Even if you know what the answer is, you cannot directly "use" that answer—because you lack the "proof path."

In my structure, a result is not necessarily just an endpoint. Once a result appears, it can become a new starting point and continue participating in the generation of new structures. A result can participate retroactively in the formation of causes. Multiple possible starting points can be inferred from a result. Cause and effect are no longer arranged in a one-way sequence, but form cyclical relationships, network relationships, and multi-dimensional relationships.

In other words, traditional algorithms ask, "Given the cause, how do we obtain the result?" The Hyperdimensional Algorithm asks, "Given the result, how do we infer or construct the cause?"

It is important to clarify that "having the result first, then the cause" is not a negation of causality, nor does it assert that "results arise from nothing." Rather, it illustrates that in higher-dimensional structures, cause and effect can serve as entry points to each other and transform into each other. As in the following pants example, the result "wearable and not dragging on the floor" is determined first, and then I retroactively find the point of operation of "folding the waist," which corresponds to the "cause" level within the structure. It is not that the result has no cause, but that the result becomes a new entry point for discovering the cause.

A fundamental assumption in the world of traditional algorithms is that time is one-way; you must compute step by step from the initial state to the final state. Even if you know what the answer is, you cannot directly "use" that answer, because you lack the proof path. The Hyperdimensional Algorithm challenges precisely this assumption: if the result is known, the causes can be inferred backward; the same result can correspond to multiple causal paths; computation is no longer walking from the start to the end, but unfolding all possible starting points from the end.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Part Five: An Everyday Example—Pants and Folding the Waist

To understand the abstract differences above more intuitively, I will use an everyday example.

A person buys a new pair of pants with an elastic waistband. The pants legs are a bit too long, dragging on the floor, which is inconvenient.

The traditional approach: noticing the pants legs are too long, roll up a section of the leg, sew it with needle and thread, then try them on. If the child grows taller and the pants become too short, the sewn-up leg cannot be let down, and the pants are ruined. Next time, buy new pants and sew them again. This method seems very natural, because the problem appears to be at the pants legs, so you address the pants legs. It corresponds to the thinking of traditional algorithms: identify the problem, locate the manifestation, process it at the location of the manifestation, and finally obtain a result. Pants legs too long? Then alter the pants legs. Data incorrect? Then modify the data. Path unsuitable? Then continue patching along the path.

My approach is different. Instead of altering the pants hem, I fold the waist once, and the pants are immediately wearable. This action is very simple, but the structural logic behind it is completely different. I do not cut the pants legs, do not sew the hem permanently, do not change the original length of the pants, and do not damage the metadata of the pants. The waist circumference, pants length, style, and fabric of the pants undergo no essential change. I simply change the structural entry point of the waist, causing the pants legs to naturally become shorter in actual wear. The waist here is not an ordinary locality, but a global control point for the entire wearing state. By adjusting this control point, the downstream problem disappears directly.

More importantly, this method is reversible, adjustable, and reusable. This is especially evident for a growing child. If the child is currently not tall enough and the pants legs are slightly long, fold the waist once; after some time when the child grows taller, let the waist down a little; when taller still, release it completely. The original structure of the entire pants remains undamaged, yet it can adapt to the child's height at different stages. With the traditional method, if you sew the pants hem permanently, when the child grows taller, the pants may become too short; if you cut them, even more impossible to restore. My method keeps the metadata unchanged, allowing the same structure to adapt to multiple states.

This example reveals several important structural differences.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

The traditional approach corresponds to the logic of traditional algorithms: the problem is at the pants legs, so the pants legs must be modified, following the "cause to effect" path step by step, skipping no step. Solve one problem at a time, irreversibly. Next time the same problem occurs, do it all over again. My approach corresponds to the logic of the Hyperdimensional Algorithm: the goal is no dragging. I do not solve the surface cause of "pants legs too long"; instead, I find a global control point—the waist. Fold the waist once, the pants legs automatically shorten, and the problem vanishes instantly. I modify no metadata. The fold at the waist is merely a temporary, reversible, dynamic folded state.

From a data perspective, the traditional approach modifies metadata—sewing the pants legs shorter permanently loses the original data. My approach modifies no metadata; the original dimensions of the pants remain intact, merely adding a temporary, reversible, dynamic folded state. The waist fold does not create new data nor destroy old data; it simply rearranges the relationships between data—shortening the effective circumference of the waist, indirectly changing the effective length of the pants legs.

In this example, "pants legs not dragging" is the target result. The traditional method starts from the cause of "pants legs too long," modifies the pants hem, and finally obtains the result of "not dragging." My method, by contrast, first clarifies the result "not dragging," then searches backward for a structural entry point, and finally discovers that folding the waist once makes the result valid. This is the practical manifestation of "Effect-Cause Theory." It is not necessary to walk step by step from cause to effect; rather, one can determine the structure backward from the effect, making the original path unnecessary.

This example also addresses the question of "metadata." What is metadata? In the pants example, the metadata are the original dimensions of the pants: waist circumference, pants length, style, fabric. These are the "essential attributes" of the pants. The temporary data is the fold at the waist; it does not change any of the pants' original dimensions, only temporarily folding a section. The traditional approach modifies metadata—the pants length is permanently shortened, the original data lost. My approach modifies no metadata—the original dimensions of the pants remain intact, only adding a temporary, reversible, dynamic folded state.

The waist's original dimensions, as metadata, are simultaneously the "starting point for multiple results": they simultaneously support multiple wearing states, such as wearing normally, wearing with one fold at the waist, wearing with two folds at the waist, etc. They are also "the result of multiple starting points": they are jointly determined by multiple starting points such as fabric selection, cutting method, design intent, etc. The

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

waist fold, as temporary data, neither creates new data nor destroys old data; it merely rearranges the relationships between data.

This is not an ordinary trick, but a kind of structural thinking. Many people, seeing pants legs too long, are drawn to the manifestation of the pants legs, so all processing revolves around the pants legs. But if viewed from the overall structure, the pants legs being too long is merely a downstream manifestation; a change in the waist position can affect the actual placement of the entire pants. That is, the problem does not necessarily have to be solved where it appears. Many low-dimensional problems have their true control point at a higher level of structure. Traditional algorithms tend to continue computing along the surface of the problem, whereas the Hyperdimensional Algorithm seeks the structural entry point.

This also explains why the Hyperdimensional Algorithm is not more complex, but may be simpler. A truly high-level structure often does not complicate the problem, but makes the complex problem simple at the correct entry point. Faced with a complex problem, traditional algorithms may add steps, add models, add computational power, add storage, add training time. The Hyperdimensional Algorithm, however, seeks a structural node; once this node is correctly activated, the originally complex path may become invalid. The so-called "bypassing the path" is not laziness, but redefining whether the path is necessary.

In this example: the traditional approach is "modify the result along the path," whereas the Hyperdimensional Algorithm is "change the entry point, rendering the entire path invalid." The conventional method continuously patches at the result end, while the other method directly changes the structural entry point, causing problems that would otherwise require multiple treatments to be restructured at the starting point in one go. The mainstream form of traditional algorithms is "start from the starting point, follow the path to obtain the result," whereas in the Hyperdimensional Algorithm, "the result itself can become a path entry point and participate in the generation of new structures." The traditional approach is "one structure corresponds to one result," whereas the Hyperdimensional Algorithm is "one structure carries multiple result states."

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

Part Six: A Data View of Not Modifying Metadata

From a data perspective, the Hyperdimensional Algorithm has a very important principle: not modifying metadata, so that it can be applied to multiple starting points or results.

Metadata are the original attributes, basic structure, and ontological information of data. When faced with different problems, traditional processing methods often modify data, copy data, process data, recompute data, or establish different paths for different results. This approach can certainly solve problems, but at the cost of data being continuously processed, paths being continuously repeated, and system complexity constantly increasing.

The Hyperdimensional Algorithm, by contrast, emphasizes that, while trying not to modify metadata, by changing entry points, relationships, conditions, and activation modes, the same metadata structure can correspond to multiple starting points and multiple results. The data ontology remains unchanged; relationships change. The basic structure remains unchanged; the mode of presentation changes. The metadata remain intact, yet can adapt to different states.

This is what I mean by "compute once, use infinitely many times." "Compute once" here is not simply caching a result, but means that once a structure is established, it no longer needs to re-establish a complete path for each state. The same data structure, activated through different entry points, can present different results. It is not "process data for different results," but "use the same data to carry multiple potential results." This is also one of the cores distinguishing the Hyperdimensional Algorithm from traditional algorithms. Traditional algorithms process data along a path; the Hyperdimensional Algorithm processes the overall structure of data, entry points, relationships, and results.

At its minimal structure, the Hyperdimensional Algorithm can be understood as: a system that, without modifying metadata, enables the same data node to correspond to multiple result paths through changes in structural entry points. This definition does not seek to immediately contract the Hyperdimensional Algorithm into a traditional mathematical formula, but first establishes its structural boundaries. It is not a single linear function, not a fixed formula, nor a simple caching mechanism. It is a relational structure: metadata remains stable, entry points can change, conditions can change, relationships can change, results can be presented in multiple directions. It is precisely in this structure that computation is no longer about executing steps, but about activating relationships.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

From a data perspective, the essence of "not modifying metadata, making it applicable to multiple starting points or results" is: the data ontology remains unchanged; the change occurs in the "mode of interpretation / usage entry point." The traditional structure is: when the problem changes, the data or path changes. In the Hyperdimensional Algorithm: when the problem changes, the interpretive structure changes, not the data.

Part Seven: Redefining "Supercomputing"

In this system, I need to clarify a term—"Supercomputing."

What I mean by "Supercomputing" is not supercomputers in the usual sense. Not more chips, higher computational power, or larger data centers. What I mean by "Supercomputing" is the "Hyperdimensional Algorithm."

True supercomputing is not necessarily the accumulation of computational power, but possibly a change in computational paradigm. When a system still relies on repetitive computation, no matter how powerful it is, it remains trapped within paths. When a system can structurally reduce computation, bypass repetitive paths, and make results into new entry points, it begins to approach true hyperdimensional computing.

In other words, traditional supercomputing pursues "using more computational power to solve bigger problems." The Hyperdimensional Algorithm pursues "using better structure so that problems no longer require such enormous computational power." These two are not contradictory, but their directions differ.

If traditional supercomputing represents the limit of computational power, then the Hyperdimensional Algorithm represents a turn in the understanding of computation. True "supercomputing" may not be more chips, larger data centers, higher energy consumption, or more complex models. True "supercomputing" may be the discovery of a structural entry point, an elimination of a path, a reverse unfolding from a result node, a way to make multiple results simultaneously valid without modifying metadata. The more powerful the computational power, if still trapped in low-dimensional paths, it remains merely powerful within those paths; once the structure is elevated, even with extremely simple operations, a higher level of efficiency can emerge.

So-called supercomputing is the limit of computational power; the Hyperdimensional Algorithm is a redefinition of the premise of "whether computation must rely on computational power."

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Part Eight: Empirical Foundations—Multi-Domain System Validation and New Paradigm Criteria

The Hyperdimensional Algorithm is not a purely theoretical construct. It has already been validated in several real-world systems.

My logistics system does not require supercomputing power or cloud support; it accomplishes complex scheduling with relatively low resources. Results can be reused, the structure can be repeatedly adapted, and there is no need to recompute everything from scratch each time. Not relying on supercomputing, not relying on the cloud, accomplishing complex scheduling with low resources—this is an established and valuable breakthrough in engineering. It demonstrates that high computational power is not the only solution; structural design can replace a portion of computational power.

My publishing system similarly adopts this structural logic, with efficiency far surpassing any current system. My extreme web page generation system is also based on the same Hyperdimensional Algorithm, repeatedly proving in multiple domains that the same structure can be stably valid. My company's employees are already using these systems, demonstrating that this structure can operate without my continuous personal intervention.

The common characteristic of these systems is: they do not rely on mainstream high-computational-power paths, do not follow fixed computational steps, and make the target result directly valid through adjustments to structural entry points. They are not one-time successes, not single-point tricks, but the repeated appearance of the same structural logic across different domains.

This demonstrates that the Hyperdimensional Algorithm is not accidental, but a structural method already validated by multiple systems. It is not a "personal trick," but a computational paradigm that is stably valid across multiple domains such as logistics, publishing, and web page generation.

According to the criterion of "existence and validity constitute a new paradigm," the Hyperdimensional Algorithm, as a new computational structural paradigm already systematically validated across multiple domains, is already valid at present. I am not proposing a hypothesis, but a different computational structure already operating in multiple systems. I do not need to prove that it is understood; I have already proven that it is stably valid in different systems. At present, within the scope of my practical application, this structure can already be regarded as a new paradigm.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Regarding the criteria for judging the Hyperdimensional Algorithm as a new paradigm: when a structure is logically self-consistent and stably valid across multiple systems, while possessing irreducible differences from existing methods, it already possesses the foundation of a new paradigm. Self-consistency is the starting point, empirical evidence is the foundation, and structural difference is the core of the paradigm. Under the standard of "existence constitutes validity," this system is already a new paradigm; whether the outside world recognizes it only affects dissemination, not validity. This is a computational structural paradigm already established in real-world systems, whose validity does not depend on external recognition or consensus.

The core difference between the Hyperdimensional Algorithm and mainstream computational paradigms is: traditional algorithms primarily rely on forward computation; once a result is produced, it usually cannot retroactively participate in new inferential structures. In the Hyperdimensional Algorithm, the result is no longer an endpoint, but a reusable structural node. Under certain conditions, the result can participate in new inferential paths, thereby reducing repetitive computation and forming a computational network with multiple starting points and multiple paths. In traditional computational structures, the result is typically an endpoint; in the Hyperdimensional Algorithm structure, the result itself can become a new starting point, thereby forming a multi-path inferential network. The Hyperdimensional Algorithm is not an upgrade to algorithms, but a re-questioning of the premise of "whether an algorithm must exist at all."

Part Nine: Clarifying Common Misunderstandings

Based on preliminary exchanges with readers from different fields, I anticipate the following six misunderstandings and clarify them in advance, to prevent the concept from being prematurely forced into inappropriate frameworks during dissemination.

Misunderstanding One: Isn't the Hyperdimensional Algorithm just dynamic programming or caching? Clarification: Dynamic programming and caching reuse results under "identical inputs." The Hyperdimensional Algorithm allows inferring different starting points from a single result—this is an essential difference. Caching solves recomputing the same problem; the Hyperdimensional Algorithm solves unfolding new problems through result structures.

Misunderstanding Two: You say "random yet orderly"—isn't that self-contradictory? Clarification: Randomness and order can coexist. The distribution of trees in a forest is random, but the overall density and species structure is orderly. Randomness is not

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

chaos, but a way of organizing structure. The randomness in the Hyperdimensional Algorithm is intrinsic and constitutive, working synergistically with order.

Misunderstanding Three: Without input and output, how do you verify the correctness of the result? Clarification: The Hyperdimensional Algorithm does not reject input and output; rather, it holds that computation does not have to operate in the linear mode of input-output. In the mode of structural manifestation, "validity" is determined by structural consistency, not by input-output correspondence. This does not abandon verification, but proposes a different verification framework from tradition.

Misunderstanding Four: Isn't this just complexity theory or chaos theory? Clarification: Complexity theory and chaos theory describe "systems that are difficult to predict." The Hyperdimensional Algorithm attempts to describe "systems that can be understood and guided through structural entry points"—not abandoning prediction, but changing the mode of prediction. Chaos theory says "prediction is difficult"; the Hyperdimensional Algorithm asks "whether prediction can be made unnecessary by changing the entry point."

Misunderstanding Five: You say "do not modify metadata," but isn't the waist fold also a modification? Clarification: The waist fold is a temporary state, not a permanent modification of original parameters. The metadata (waist circumference, pants length, style, fabric) has no change whatsoever. This is the difference between "state superposition" and "attribute modification." The fold is reversible, temporary, and does not change essential attributes.

Misunderstanding Six: Does the Hyperdimensional Algorithm negate the value of traditional algorithms? Clarification: Absolutely not. Traditional algorithms have been extremely important in human technological history. Without traditional algorithms, there would be no modern computers, databases, communication systems, artificial intelligence, engineering simulation, or supercomputing. The value of traditional algorithms cannot be denied. However, acknowledging the value of traditional algorithms does not mean acknowledging them as the endpoint of algorithms. Traditional algorithms solve problems of efficiency within a given computational paradigm, whereas the Hyperdimensional Algorithm raises the question of the computational paradigm itself. One is about how to walk better; the other is about whether one needs to take that path at all.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Part Ten: The Hyperdimensional Algorithm and the History of Human Computation

To help readers better position the Hyperdimensional Algorithm within the history of human computation, I offer a brief macroscopic overview here.

Human understanding of "computation" has gone through multiple stages. The first stage was manual computation; algorithms existed as human steps, slow and error-prone, but through this process, humans understood the basic rules of computation. The second stage was mechanical computation, from Pascal's calculator to Babbage's Analytical Engine; computation was mechanized, but algorithms still depended on physical structures. The third stage was electronic computation and the Turing paradigm; the von Neumann architecture became widespread; algorithms were encoded as programs; the Turing machine became the boundary of computability. The fourth stage is parallel and supercomputing, accumulating computational power through a large number of computing units to tackle more complex problems, but the underlying logic remains an extension of the Turing paradigm.

Currently, humanity is at the entrance of the fifth stage. Quantum computing attempts to break through the limitations of classical bits; neuromorphic computing attempts to mimic the structure of biological nervous systems; and the Hyperdimensional Algorithm attempts to break through the linear framework of "input → steps → output" itself.

The relationship between the Hyperdimensional Algorithm and traditional algorithms is not one of replacement, but of hierarchy. Traditional algorithms solve the problem of "how to compute more efficiently on a given path." The Hyperdimensional Algorithm asks "whether the path can be redefined, or even bypassed." If we view the history of human computation as a process of constantly breaking through its own boundaries, then the Hyperdimensional Algorithm is precisely a new node in this process. It does not solve problems within the old framework, but proposes a new framework.

This judgment does not imply that the Hyperdimensional Algorithm is already mature or complete. Quite the opposite. As a new concept, its definition, boundaries, verification methods, and application scenarios are still in the process of formation. The purpose of this paper is precisely to anchor the origin of this concept, providing a stable theoretical foundation for subsequent development.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Part Eleven: Conclusion—This Is Not Optimization, but Redefinition

The difference between the Hyperdimensional Algorithm and traditional algorithms is not a difference of "better" versus "worse," not a difference of "faster" versus "slower," but a fundamental difference at the paradigm level.

Traditional algorithms pursue predictable control. You give me the input, I know what output you will get, because I have computed every step. It is the engineer's paradigm: design, build, test, verify.

The Hyperdimensional Algorithm describes comprehensible emergence. I cannot precisely predict every intermediate state, but I know the overall pattern will present itself in some orderly way. Each piece of data is alive and has a "perspective." It is more like the paradigm of an ecologist or complex systems theorist: observe, understand, guide, utilize emergent order.

Traditional algorithms gradually modify the result along a given path. The Hyperdimensional Algorithm makes the target result immediately valid by changing the structural entry point, thereby bypassing the entire path.

Traditional algorithms "process data for different results." The Hyperdimensional Algorithm is "use the same data to carry multiple potential results."

Traditional algorithms pursue "compute correctly once." The Hyperdimensional Algorithm pursues "compute correctly once, and then never need to compute again."

This is not an optimization of algorithms, but a re-questioning of the premise of "whether an algorithm must exist at all."

Therefore, the Hyperdimensional Algorithm is not algorithm optimization, but algorithm redefinition. It is not about running faster on the traditional algorithm track, but asking whether there is another path outside the traditional track, or even whether one can be independent of the path itself. It is not trying to prove traditional algorithms useless, but to point out that traditional algorithms are merely a low-dimensional form of algorithm. It is not asking "how to get results in less time," but asking "whether a result can be directly established through structure." It is not asking "how to process more data," but asking "whether the same data can carry more potential results."

Within the mainstream computational paradigm, the Hyperdimensional Algorithm may be seen as uncomputable, unverifiable, or difficult to incorporate into existing

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

boundaries of computation theory. But this precisely constitutes its paradigm difference. The fact that a new concept appears unstable within an old paradigm does not necessarily mean it is meaningless; it may also mean that the old paradigm itself cannot fully accommodate it. At present, the Hyperdimensional Algorithm is first and foremost a structural proposition, an original definition of a new computational view, rather than a mature system with a completed engineering loop. It requires subsequent continuous supplementation, expansion, verification, and application, but its conceptual origin must first be clearly articulated.

Ultimately, what the Hyperdimensional Algorithm points to is a new view of computation: an algorithm is not necessarily just mathematics, not necessarily just steps, not necessarily just a program, not necessarily just a processing process between input and output. An algorithm can also be a way of organizing multi-dimensional relationships, a structural network of data, entry points, relationships, states, and results. It can contain order and randomness; it can go from cause to effect, or retroactively generate causes from effects; it can adapt to multiple states without modifying metadata; it can make one result a new starting point, or have multiple starting points converge into the same result.

A truly advanced algorithm is not necessarily more complex computation, but a multi-dimensional organizational method that can reduce computation, enliven structures, turn results into entry points, and allow cause and effect to form cycles.

This is the core meaning of my proposing the "Hyperdimensional Algorithm." It is not an ordinary new term, but a new concept, a new paradigm, a new computational structure already established in multiple real-world systems. Its focus is not on computing repeatedly faster, but on reducing repetitive computation; not on accumulating computational power, but on restructuring entry points; not on constantly modifying data, but on keeping metadata unchanged while changing relationships; not on letting results stop at endpoints, but on making results become new starting points. Traditional algorithms seek results along paths; the Hyperdimensional Algorithm activates results within structures. Traditional algorithms pursue completing computation; the Hyperdimensional Algorithm questions whether computation must exist in its traditional form. This is the most fundamental difference between it and the current definition of algorithms.

Appendix: Boundaries of This Paper and Open Questions

This paper proposes a preliminary framework for the Hyperdimensional Algorithm, not a complete formal definition. The following questions await further development. They

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

do not constitute a negation of the definitions in this paper, but precisely the directions for next steps.

First, convergence conditions. What is the "completion" marker of the Hyperdimensional Algorithm? How does one determine whether a result is "valid"? In traditional algorithms, the answer is defined by input-output correspondence. In the Hyperdimensional Algorithm, the answer may need to be defined by structural consistency. This definition is not yet complete.

Second, resource representation. How can multi-dimensional superimposed states be represented within finite resources? Do traditional resource metrics such as time, space, and computational power still apply? If so, how are they redefined? If not, what metrics replace them? These questions await exploration.

Third, ensuring orderliness. By what rules is the "order" in "random yet orderly" guaranteed? Where is the boundary between randomness and order? Under what circumstances does randomness disrupt order? Under what circumstances does order suppress randomness? These questions require further research.

Fourth, selection of reverse inference. When inferring multiple starting points backward from a result, how does one screen or evaluate the relative merits of different starting points? Is there some measure of "reverse inference efficiency"? If so, what is its relationship to the efficiency measure of forward computation?

Fifth, interfaces with traditional systems. How does the Hyperdimensional Algorithm work with existing Turing machine systems? Is it a replacement, a supplement, or a hierarchical layer? In practical engineering, how is the boundary between the Hyperdimensional Algorithm and traditional algorithms delineated? Do hybrid modes exist?

Sixth, verification framework. How are the results of the Hyperdimensional Algorithm verified? If the result is not obtained through linear steps, how are reproducibility and testability established? Is a completely different verification philosophy required?

These questions are not flaws in this paper, but inevitable characteristics of this paper as an "anchor document." The proposal of any new paradigm comes with open questions. This paper is a beginning, not a conclusion.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Bibliographic Note

The "Hyperdimensional Algorithm" proposed in this paper is not a direct extension of any existing academic school of thought. However, in terms of ideas, the following fields provide a dialogic background for this paper, offered only for readers' orientation, and are not citations in the traditional academic sense.

Turing machine and computability theory serve as the benchmark against which this paper compares traditional algorithms. Inverse problems and Bayesian inference serve as existing attempts at "inferring causes from results"; this paper's "Effect-Cause Theory" is related but differs in direction. Emergence theory and complex systems serve as existing descriptions of "structures manifesting results"; the Hyperdimensional Algorithm attempts to make emergence "comprehensible" and "guidable." Knowledge graphs and graph databases serve as existing practices of "data intersecting at multiple nodes"; the Hyperdimensional Algorithm, on this foundation, prioritizes the dimension of "variable entry points." Non-classical computational paradigms, including quantum computing, analog computing, neuromorphic computing, etc., break through classical bits or classical architectures, while the Hyperdimensional Algorithm focuses more on breaking through the linear "input → steps → output" framework itself.

The relationship between this paper and the above fields is one of dialogue and transcendence, not inheritance or refutation. The goal of this paper is not to make incremental improvements within these fields, but to raise a new level of questioning above them.

Related Articles

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

[Algorithme Extreme] Algorithme Hyperdimensionnel

Une redéfinition du « calcul » lui-même

Auteur : Wu Zhaohui JEFFI CHAO HUI WU

Résumé

Cet article propose le nouveau concept d'« algorithme hyperdimensionnel » comme un réexamen structurel des frontières de la définition traditionnelle de l'algorithme. Les algorithmes traditionnels sont généralement construits sur le cadre linéaire de l'entrée, des étapes, des règles et de la sortie, en mettant l'accent sur la finitude, la déterminisme, l'efficacité, la faisabilité et des frontières claires entre l'entrée et la sortie. Bien que les superordinateurs modernes possèdent une puissance de calcul extrêmement élevée, leur logique sous-jacente consiste encore principalement à exécuter des paradigmes algorithmiques établis à plus grande échelle. Cet article soutient que ce qui mérite vraiment d'être discuté dans le « supercalcul » n'est pas seulement la croissance de l'échelle de calcul, mais un changement fondamental du paradigme de calcul lui-même. L'algorithme hyperdimensionnel n'est pas une version accélérée des algorithmes traditionnels, ni une extension des algorithmes mathématiques ; c'est une vision structurelle du calcul multidimensionnelle, superposée, aléatoire mais ordonnée. Dans cette structure, la donnée n'est plus simplement une entrée passive ou une sortie terminale, mais un attribut de nœud possédant simultanément de multiples points de départ et de multiples résultats ; un résultat n'est plus seulement un point d'arrivée, mais peut devenir une nouvelle entrée ; les métadonnées n'ont pas besoin d'être modifiées de manière répétée, mais peuvent correspondre à différents états et résultats par l'intermédiaire d'entrées structurelles, de changements relationnels et d'activation conditionnelle. Cet article utilise la philosophie de la « théorie effet-cause » et l'exemple quotidien du « pliage de la taille d'un pantalon » pour illustrer comment l'algorithme hyperdimensionnel peut rendre un résultat cible directement valide en changeant l'entrée structurelle, réduisant ainsi le calcul répétitif et même contournant le chemin de calcul original. Cet article vise à établir une définition préliminaire, des frontières structurelles et des bases conceptuelles pour l'« algorithme hyperdimensionnel », fournissant un nœud théorique original pour le développement futur du calcul hyperdimensionnel, des algorithmes extrêmes et d'un nouveau système scientifique. Selon le critère « l'existence et la validité constituent un nouveau paradigme », l'algorithme hyperdimensionnel, en tant

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

que nouveau paradigme structurel de calcul déjà validé systématiquement dans de multiples domaines, est déjà valide à l'heure actuelle.

Mots-clés : Algorithme hyperdimensionnel ; Algorithme extrême ; Calcul hyperdimensionnel ; Théorie effet-cause ; Métadonnées ; Paradigme de calcul ; Entrée structurelle ; Causalité non linéaire ; Nouvelle science

Introduction : Pourquoi nous devons rediscuter de l'« algorithme »

Lorsque je propose l'« algorithme hyperdimensionnel », ce n'est pas pour donner un nom plus grand aux algorithmes existants, ni pour emballer les algorithmes traditionnels dans un concept nouveau et original. Au contraire, je veux discuter d'une question plus fondamentale : la compréhension humaine actuelle de l'« algorithme » est-elle confinée dans un cadre trop étroit ?

Aujourd'hui, lorsque les gens parlent d'algorithmes, ils pensent généralement aux formules mathématiques, au code informatique, aux étapes logiques, à l'entrée-sortie, à l'entraînement de modèles, au traitement de données, à l'optimisation de chemins, au classement de recherches, à la recommandation automatique et au raisonnement en intelligence artificielle. Ce sont certainement des formes importantes d'algorithmes, et elles soutiennent le fonctionnement des ordinateurs modernes, d'Internet, des bases de données, de l'intelligence artificielle, du supercalcul, des systèmes industriels et de la société de l'information. Cependant, si l'on ne comprend l'algorithme que comme « un ensemble d'étapes ordonnées », que comme un processus qui « commence par une entrée, passe par un traitement selon des règles, et aboutit à une sortie », alors cette compréhension confine déjà le calcul dans un cadre de faible dimension.

Nous vivons à une époque dominée par les algorithmes. Des moteurs de recherche aux systèmes de recommandation, de la prévision météorologique à l'intelligence artificielle, les frontières des algorithmes sont les frontières de l'intelligence humaine. Mais une question est rarement posée : les algorithmes ne peuvent-ils être que cela ? Pourquoi le calcul doit-il suivre le modèle linéaire « entrée → étapes → sortie » ? Pourquoi le même problème doit-il être calculé à partir de zéro chaque fois ? Cet article tente de remettre en question ces hypothèses par défaut et de proposer un paradigme de calcul complètement différent — l'algorithme hyperdimensionnel. Selon le critère « l'existence et la validité constituent un nouveau paradigme », l'algorithme hyperdimensionnel, en tant que nouveau paradigme structurel de calcul déjà validé systématiquement dans de multiples domaines, est déjà valide à l'heure actuelle.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Il est important de noter que l'« algorithme hyperdimensionnel » proposé dans cet article est un concept complètement différent du « calcul hyperdimensionnel » (Hyperdimensional Computing, HDC) qui existe dans le monde académique depuis près de trente ans. Le HDC utilise des vecteurs binaires aléatoires de très haute dimension pour le codage et le calcul, recherchant une représentation et un calcul plus efficaces, mais il reste à l'intérieur du paradigme linéaire « entrée → traitement → sortie ». L'algorithme hyperdimensionnel est totalement différent : il questionne si un résultat peut être directement établi par la structure, si le chemin de calcul lui-même peut être contourné, si les métadonnées peuvent rester inchangées tout en s'adaptant à de multiples résultats. Leurs noms sont similaires, mais leurs connotations sont opposées. L'algorithme hyperdimensionnel n'est pas une amélioration des algorithmes, mais un réexamen de la prémisse « un algorithme doit-il nécessairement exister ? »

Première partie : La définition standard actuelle de l'« algorithme »

Dans les systèmes dominants de l'informatique, des mathématiques et de l'ingénierie, il existe une définition de base de l'algorithme acceptée depuis longtemps : un algorithme est un processus de calcul bien défini, composé d'un nombre fini d'étapes, qui prend une ou plusieurs entrées, les transforme par des règles déterministes, et produit une ou plusieurs sorties dans un temps fini. Cette compréhension n'est pas l'opinion personnelle de quiconque, mais un consensus fondamental formé au long du développement de la civilisation informatique moderne. Elle soutient la logique sous-jacente des logiciels, du matériel, des bases de données, des systèmes en réseau, des modèles d'intelligence artificielle et des centres de supercalcul. Aussi complexe qu'un algorithme puisse paraître, son cœur tourne encore autour de l'entrée, des règles, des étapes et de la sortie.

Cette conception standard de l'algorithme peut être décomposée en plusieurs caractéristiques fondamentales.

Premièrement, la finitude. Un algorithme doit se terminer en un nombre fini d'étapes ; il ne peut pas boucler indéfiniment ni s'exécuter éternellement. Un processus qui ne s'arrête jamais, même s'il peut être formellement décrit, ne peut guère être considéré comme un algorithme efficace. Tous les programmes de calcul scientifique et les flux de traitement de données ont des conditions d'arrêt claires.

Deuxièmement, le déterminisme. Étant donné la même entrée dans les mêmes conditions, l'algorithme doit produire la même sortie. Même si certains algorithmes

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

introduisent de l'aléatoire, il s'agit généralement d'un aléatoire contrôlé, reproductible, ou d'un mécanisme probabiliste interprétable statistiquement, et non d'un aléatoire intrinsèque totalement incontrôlable. Les résultats des simulations numériques doivent être reproductibles ; c'est une exigence fondamentale du calcul scientifique.

Troisièmement, des frontières claires entre l'entrée et la sortie. L'entrée vient en premier, le traitement au milieu, la sortie en dernier ; le point de départ et le point d'arrivée ont des frontières structurelles claires. Quelles que soient les données que vous donnez à l'algorithme, après traitement, il vous donne un résultat ; cette frontière est distincte. Entrée d'abord, traitement ensuite, sortie enfin ; l'ordre ne peut être inversé.

Quatrièmement, l'efficacité. Chaque étape d'un algorithme doit être une opération pratiquement exécutable ; elle ne peut contenir de sauts impossibles à exécuter, à décrire ou à vérifier. Chaque étape doit être une opération de base qu'un humain pourrait exécuter avec du papier et un crayon, ou qu'un ordinateur pourrait réellement exécuter — addition, soustraction, multiplication, division, jugements logiques, lecture/écriture de données, etc.

Cinquièmement, la faisabilité. Même si un algorithme est théoriquement valide, si le temps, la puissance de calcul, la mémoire ou les ressources de stockage qu'il consomme sont totalement inacceptables, il est difficile de le considérer comme un algorithme efficace en termes d'ingénierie. Un algorithme théoriquement correct mais qui prendrait des centaines de millions d'années à s'exécuter n'est pas considéré comme un algorithme faisable en ingénierie.

Ce système complet de concepts algorithmiques remonte finalement au modèle de la machine de Turing. En tant qu'abstraction centrale de la théorie du calcul moderne, la machine de Turing définit la frontière de base de la « calculabilité ». Aussi puissants que soient les ordinateurs d'aujourd'hui, aussi avancées que soient les puces, aussi grande que soit l'échelle des centres de supercalcul, leur logique sous-jacente ne s'est jamais vraiment écartée de ce chemin : entrée, règles, étapes, sortie. Les superordinateurs modernes ne font qu'accélérer l'exécution de ce processus par une plus grande échelle matérielle, un parallélisme plus élevé et un ordonnancement système plus complexe. C'est-à-dire que, dans le contexte traditionnel, le « supercalcul » est principalement une expansion de l'échelle de calcul, et non nécessairement un changement fondamental du paradigme de calcul. La puissance de calcul peut être multipliée par des milliers ou des dizaines de milliers, mais si la logique sous-jacente reste « entrée, étapes, sortie », alors elle reste à l'intérieur du paradigme algorithmique traditionnel.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Deuxième partie : Définition de l'« algorithme hyperdimensionnel »

L'« algorithme hyperdimensionnel » que je propose émerge précisément de ce contexte. Ce n'est pas une version améliorée des algorithmes traditionnels, ni un algorithme plus rapide, ni une formule mathématique plus complexe, ni une technique de programmation plus avancée. C'est une compréhension structurelle complètement différente.

Tout d'abord, il faut expliquer le sens du terme « hyperdimensionnel ». « Hyperdimensionnel » a ici deux significations. La première, il ne se limite pas à une séquence linéaire unidimensionnelle d'étapes, mais permet à de multiples dimensions de se déployer simultanément. La seconde, il tente de transcender les frontières définitionnelles traditionnelles du « calcul » — un algorithme ne doit plus nécessairement être mathématique, séquentiel ou déterministe. Un algorithme traditionnel, c'est comme conduire sur une route à sens unique : vous devez partir du point A, passer par B, C, D pour arriver à Z. L'algorithme hyperdimensionnel ne considère pas que le calcul doive être une route à sens unique. Il considère que le calcul peut être un réseau, un champ, une structure multidimensionnelle, où n'importe quel nœud peut être une entrée et n'importe quel nœud peut être une sortie.

Dans mon système de Nouvelle Science, un algorithme n'est pas nécessairement un algorithme mathématique, pas nécessairement un calcul unique ordonné, pas nécessairement contraint de s'exécuter selon des étapes fixes, et pas nécessairement strictement soumis au modèle linéaire « entrée, traitement, sortie ». L'algorithme hyperdimensionnel est une structure multidimensionnelle, superposée, aléatoire mais ordonnée. Dans cette structure, chaque donnée n'est ni une entrée isolée ni une sortie fixe, mais possède simultanément de multiples rôles : elle peut être à la fois le point de départ de multiples résultats et le résultat produit par l'action conjointe de multiples points de départ.

En d'autres termes, les algorithmes traditionnels placent les données dans un flux de processus, tandis que l'algorithme hyperdimensionnel place les données dans une structure. Dans les algorithmes traditionnels, les données sont généralement divisées en données d'entrée, données intermédiaires et données de sortie. Chaque type de donnée a une position et un rôle clairs. L'entrée est l'entrée, le résultat est le résultat, le processus intermédiaire est le processus intermédiaire. Cependant, dans l'algorithme hyperdimensionnel, une donnée n'a pas qu'une seule identité. Elle peut être un résultat dans une direction et un point de départ dans une autre ; elle peut être un objet appelé dans une structure et devenir une entrée déclenchant de nouveaux résultats dans une

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

autre structure. La donnée n'est plus seulement un matériau passif, mais un nœud relationnel multidimensionnel.

Cela correspond précisément au cœur de l'algorithme hyperdimensionnel : chaque donnée est le point de départ de multiples résultats et aussi le résultat de multiples points de départ. Les algorithmes traditionnels utilisent des « étapes ordonnées » pour produire un résultat unique ; l'algorithme hyperdimensionnel est plus proche d'une structure d'états multidimensionnelle où, dans la superposition de l'aléatoire et de l'ordre, le résultat n'est pas calculé pour exister, mais se manifeste naturellement dans la structure. Dans ce système, la donnée est à la fois le point de départ et une composante du résultat.

Pour présenter plus clairement les caractéristiques fondamentales de l'algorithme hyperdimensionnel, je les décompose en cinq aspects suivants.

Premièrement, la nature non mathématique. Les algorithmes dominants sont essentiellement mathématiques — ils peuvent être entièrement décrits par la logique symbolique et les formules mathématiques. L'algorithme hyperdimensionnel n'est pas nécessairement mathématique. Il peut être basé sur des principes physiques, des relations géométriques, un nouveau paradigme de la théorie de l'information, ou même des principes de la conscience. Le calcul ne doit pas nécessairement être accompli par des « opérations mathématiques » ; il peut se présenter naturellement à travers des relations géométriques dans un espace de dimension supérieure. Par exemple, la forme d'une bulle de savon est déterminée par le principe de minimisation de la tension superficielle. Ce n'est pas un « algorithme mathématique » qui calcule — c'est la physique elle-même qui « calcule ». L'algorithme hyperdimensionnel tente de comprendre ce type de « calcul », plutôt que d'utiliser des formules mathématiques pour le simuler.

Deuxièmement, la multidimensionnalité et la superposition. Les algorithmes traditionnels exécutent des étapes ordonnées dans une seule dimension, avec un seul état à chaque étape. L'algorithme hyperdimensionnel permet à de multiples dimensions d'exister simultanément, avec de multiples états se superposant et coexistants. L'algorithme ne suit pas un seul chemin ; il se déploie plutôt dans un champ d'états multidimensionnel. Le résultat n'est pas « calculé » mais « se manifeste » à partir de ce champ. Par exemple, lorsqu'un flocon de neige se forme dans l'air, il n'y a pas d'« algorithme » pour dire à chaque molécule d'eau où aller. L'agencement des molécules d'eau est le résultat de multiples dimensions — température, humidité, flux d'air — agissant ensemble. La forme du flocon de neige « se manifeste » plutôt que d'être « calculée ».

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

Troisièmement, l'aléatoire dans l'ordre. Dans les algorithmes traditionnels, l'aléatoire est instrumental, externe, une perturbation contrôlable ; l'algorithme lui-même est déterministe. Dans l'algorithme hyperdimensionnel, l'aléatoire est intrinsèque et structurel. L'aléatoire et l'ordre coexistent et travaillent ensemble, constituant conjointement l'essence de l'algorithme. Ce n'est pas « un algorithme avec de l'aléatoire », mais « l'aléatoire lui-même est une composante organique de l'algorithme ». Par exemple, la structure écologique d'une forêt — la distribution des arbres semble aléatoire, mais dans son ensemble, elle présente une densité ordonnée et des relations entre espèces. Cet « aléatoire » n'est pas un bogue, mais une condition préalable au fonctionnement normal de la structure écologique.

Quatrièmement, les rôles multiples des données. Dans les algorithmes traditionnels, le rôle des données est singulier — l'entrée est l'entrée, la sortie est la sortie, le résultat intermédiaire est le résultat intermédiaire. Dans l'algorithme hyperdimensionnel, chaque donnée est simultanément le point de départ de multiples résultats et le résultat de multiples points de départ. Un nœud de données peut se déployer en aval en de multiples chemins de résultats, et peut également être convergé en amont par de multiples causes. La donnée n'est plus un point dans un flux linéaire, mais un nœud de convergence dans un réseau. Par exemple, considérons la taille d'une personne. Dans un algorithme traditionnel, la taille est une « entrée » — vous l'entrez pour obtenir des sorties comme la prédiction de poids, la taille de vêtement, etc. Mais la taille est aussi un « résultat » — elle est déterminée conjointement par de multiples « points de départ » tels que la génétique, la nutrition et l'exercice. Dans l'algorithme hyperdimensionnel, la donnée de taille est simultanément un point de départ et un résultat, et non un choix binaire.

Cinquièmement, métadonnées inchangées, relations variables. Face à différents problèmes, les algorithmes traditionnels soit modifient les données elles-mêmes, soit recalculent tout. L'algorithme hyperdimensionnel ne modifie pas les métadonnées — les attributs essentiels des données restent inchangés. Ce qui change, ce sont les entrées, les modes d'interprétation et les relations entre les données. La même structure de métadonnées, activée par différentes entrées, peut présenter des résultats complètement différents. Cela signifie : calculez une fois, utilisez une infinité de fois. Par exemple, considérons le même passage de texte. Vous pouvez le lire comme de la poésie, le décoder comme un chiffrement, ou l'analyser comme un document historique. Le texte lui-même n'a pas changé — vous avez seulement changé « l'entrée ». L'algorithme hyperdimensionnel recherche précisément cette capacité : « mêmes données, entrées multiples, résultats multiples. »

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Du point de vue de la structure des données, les métadonnées elles-mêmes ne sont pas modifiées ; plutôt, par différentes entrées et conditions, elles correspondent à de multiples points de départ et résultats au sein de la même structure. Les données ne sont plus traitées de manière répétée, mais, tout en gardant leur structure inchangée, sont activées par différentes entrées, présentant ainsi différents résultats. L'approche traditionnelle est « traiter les données pour différents résultats », tandis que l'algorithme hyperdimensionnel est « utiliser les mêmes données pour porter de multiples résultats potentiels. »

Troisième partie : Différences fondamentales entre l'algorithme hyperdimensionnel et les algorithmes traditionnels

Sur la base des définitions ci-dessus, l'algorithme hyperdimensionnel et les algorithmes traditionnels présentent des différences fondamentales sur de multiples dimensions fondamentales. Celles-ci sont détaillées ci-dessous une par une.

En termes de nature essentielle, les algorithmes traditionnels sont mathématiques et formels, pouvant être entièrement décrits par la logique symbolique et les formules mathématiques ; ce sont essentiellement des objets mathématiques. L'algorithme hyperdimensionnel n'est pas nécessairement mathématique ; il peut être basé sur des principes physiques, géométriques, informationnels ou même conscients. Il n'est pas un sous-ensemble des mathématiques, mais une catégorie plus large. Cette différence peut être résumée comme : de « l'objet mathématique » à la « loi universelle ».

En termes d'ordre temporel, les algorithmes traditionnels suivent un ordre temporel singulier, ordonné et linéaire, étape A à B à C, strictement séquentiel, ou décomposable en sous-étapes ordonnées parallèles ; la flèche du temps est irréversible. L'algorithme hyperdimensionnel n'a pas de séquence d'étapes fixe ; il est multidimensionnel, superposé, aléatoire mais ordonné. Le résultat peut être indépendant de l'« ordre » d'exécution, car il n'existe pas d'« ordre » traditionnel. Cette différence peut être résumée comme : du « temps linéaire » à la « simultanéité de dimension supérieure ».

En termes de causalité, les algorithmes traditionnels suivent une chaîne causale déterministe : étant donné la même entrée et le même état initial, la sortie est toujours unique ; la cause précède l'effet, une flèche unidirectionnelle irréversible. L'algorithme hyperdimensionnel suit une causalité superposée : chaque donnée est le point de départ de multiples résultats et le résultat de multiples points de départ. C'est ma philosophie de la « théorie effet-cause » — il est possible d'avoir d'abord le résultat, puis la cause. Le résultat n'est pas un point d'arrivée, mais peut devenir un nouveau

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

point de départ. Cette différence peut être résumée comme : de « la cause unique, l'effet unique » au « réseau causal totalement interconnecté ».

En termes de structure des données, les algorithmes traditionnels suivent une structure de flux unidirectionnel de l'entrée au traitement puis à la sortie ; les données ont un rôle unique avec des frontières claires ; les données d'entrée restent des entrées du début à la fin, les données de résultat restent des résultats. Dans l'algorithme hyperdimensionnel, les données ont des rôles multiples ; la même donnée est simultanément un résultat et un point de départ ; il n'y a ni points de départ absolus ni points d'arrivée absolus, seulement des nœuds dans un réseau. Cette différence peut être résumée comme : de « la fluidité unidirectionnelle » à « la symbiose récursive ».

En termes de mode de calcul, les algorithmes traditionnels calculent à partir de zéro chaque fois qu'ils rencontrent un problème, même s'ils ont déjà calculé des problèmes similaires mille fois auparavant, même si la réponse est déjà connue ; à la mille-et-unième fois, ils parcourent encore tout le processus. C'est le calcul « sans état ». Dans l'algorithme hyperdimensionnel, si un résultat correspondant existe, il n'est pas nécessaire de refaire tout le parcours ; à partir d'un résultat, de multiples points de départ peuvent être inférés, et les chemins causaux peuvent être déployés en sens inverse. Le calcul est avec état, a de la mémoire et est réutilisable. Cette différence peut être résumée comme : de « recalculer à chaque fois » à « réutiliser une fois ».

En termes du rôle de l'aléatoire, l'aléatoire dans les algorithmes traditionnels est pseudo-aléatoire ou injecté de l'extérieur ; les nombres aléatoires ne sont que des outils pour l'échantillonnage, l'approximation ou l'optimisation ; l'algorithme lui-même est déterministe. L'aléatoire dans l'algorithme hyperdimensionnel est intrinsèque et constitutif ; l'aléatoire est une composante organique, coexistant et coopérant avec l'ordre. Ce n'est pas « un algorithme contient de l'aléatoire », mais « l'aléatoire est une raison pour laquelle l'algorithme peut fonctionner ». Cette différence peut être résumée comme : de « l'aléatoire instrumental » à « l'aléatoire constitutif ».

En termes de compréhension des ressources, les algorithmes traditionnels recherchent un calcul plus rapide et plus précis étant donné les ressources ; les ressources sont des contraintes ; l'objectif de l'algorithme est « d'effectuer le calcul dans les contraintes ». L'algorithme hyperdimensionnel recherche à rendre le calcul lui-même inutile par la conception structurelle ; non pas « calculer plus vite », mais « contourner le calcul » ; les ressources ne sont pas utilisées pour être « consommées », mais pour « concevoir des entrées ». Cette différence peut être résumée comme : de « l'optimisation sous contraintes de ressources » à « l'élimination par la conception structurelle ».

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Quatrième partie : La philosophie de la théorie effet-cause

Sur la base de la comparaison ci-dessus, je dois développer davantage la philosophie de la « théorie effet-cause ». C'est l'une des idées fondamentales de l'algorithme hyperdimensionnel.

La pensée traditionnelle soutient généralement que la cause précède l'effet ; d'abord la cause, puis l'effet ; d'abord l'entrée, puis la sortie. Ce concept se manifeste dans les algorithmes comme suit : vous devez partir du point de départ et marcher étape par étape jusqu'au point d'arrivée. Même si vous savez quelle est la réponse, vous ne pouvez pas directement « utiliser » cette réponse — parce que vous n'avez pas le « chemin de preuve ».

Dans ma structure, un résultat n'est pas nécessairement juste un point d'arrivée. Une fois qu'un résultat apparaît, il peut devenir un nouveau point de départ et continuer à participer à la génération de nouvelles structures. Un résultat peut participer rétroactivement à la formation de causes. De multiples points de départ possibles peuvent être inférés à partir d'un résultat. La cause et l'effet ne sont plus disposés en une séquence unidirectionnelle, mais forment des relations cycliques, des relations de réseau, des relations multidimensionnelles.

En d'autres termes, les algorithmes traditionnels demandent : « Étant donné la cause, comment obtenir le résultat ? » L'algorithme hyperdimensionnel demande : « Étant donné le résultat, comment inférer ou construire la cause ? »

Il est important de clarifier que « avoir d'abord le résultat, puis la cause » n'est pas une négation de la causalité, ni n'affirme que « les résultats surgissent de rien ». Il s'agit plutôt d'illustrer que, dans des structures de dimension supérieure, la cause et l'effet peuvent servir d'entrées l'un à l'autre et se transformer mutuellement. Comme dans l'exemple du pantalon qui suit, le résultat « pouvoir être porté sans traîner par terre » est déterminé en premier, puis je trouve rétroactivement le point d'opération du « pliage de la taille », qui correspond au niveau de la « cause » dans la structure. Ce n'est pas que le résultat n'a pas de cause, mais que le résultat devient une nouvelle entrée pour découvrir la cause.

Une hypothèse fondamentale dans le monde des algorithmes traditionnels est que le temps est unidirectionnel ; vous devez calculer étape par étape de l'état initial à l'état final. Même si vous savez quelle est la réponse, vous ne pouvez pas directement « utiliser » cette réponse, parce que vous n'avez pas le chemin de preuve. L'algorithme hyperdimensionnel défie précisément cette hypothèse : si le résultat est connu, les causes peuvent être inférées en arrière ; le même résultat peut correspondre à de

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

multiples chemins causaux ; le calcul n'est plus une marche du début à la fin, mais le déploiement de tous les points de départ possibles à partir de la fin.

Cinquième partie : Un exemple quotidien — le pantalon et le pliage de la taille

Pour comprendre plus intuitivement les différences abstraites ci-dessus, j'utiliserai un exemple quotidien.

Une personne achète un nouveau pantalon avec une ceinture élastique. Les jambes du pantalon sont un peu trop longues et traînent par terre, ce qui est gênant.

L'approche traditionnelle : constatant que les jambes du pantalon sont trop longues, on remonte une section de la jambe, on la coud avec une aiguille et du fil, puis on essaie. Si l'enfant grandit et que le pantalon devient trop court, la jambe cousue ne peut pas être rallongée, et le pantalon est perdu. La prochaine fois, on achète un nouveau pantalon et on le coud à nouveau. Cette méthode semble très naturelle, parce que le problème semble se situer au niveau des jambes du pantalon, donc on s'attaque aux jambes du pantalon. Elle correspond à la pensée des algorithmes traditionnels : identifier le problème, localiser la manifestation, traiter à l'endroit de la manifestation, et enfin obtenir un résultat. Jambes trop longues ? Alors on modifie les jambes. Données incorrectes ? Alors on modifie les données. Chemin inapproprié ? Alors on continue à rapiécer le long du chemin.

Mon approche est différente. Au lieu de modifier l'ourlet du pantalon, je plie la taille une fois, et le pantalon est immédiatement portable. Cette action est très simple, mais la logique structurelle qui la sous-tend est complètement différente. Je ne coupe pas les jambes du pantalon, je ne couds pas l'ourlet de manière permanente, je ne change pas la longueur originale du pantalon, et je n'endommage pas les métadonnées du pantalon. Le tour de taille, la longueur du pantalon, la coupe et le tissu ne subissent aucun changement essentiel. Je change simplement l'entrée structurelle de la taille, ce qui fait que les jambes du pantalon deviennent naturellement plus courtes lorsqu'il est porté. La taille n'est pas ici une simple localité, mais un point de contrôle global pour l'état de port entier. En ajustant ce point de contrôle, le problème en aval disparaît directement.

Plus important encore, cette méthode est réversible, ajustable et réutilisable. Cela est particulièrement évident pour un enfant en croissance. Si l'enfant n'est pas encore assez grand et que les jambes du pantalon sont un peu longues, on plie la taille une fois ; quelque temps plus tard, lorsque l'enfant a grandi, on relâche la taille un peu ;

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

quand il a encore grandi, on la relâche complètement. La structure originale du pantalon reste intacte, mais elle peut s'adapter à la taille de l'enfant à différents stades. Avec la méthode traditionnelle, si l'on coud l'ourlet de manière permanente, lorsque l'enfant grandit, le pantalon peut devenir trop court ; si on le coupe, c'est encore plus impossible à restaurer. Ma méthode garde les métadonnées inchangées, permettant à la même structure de s'adapter à de multiples états.

Cet exemple révèle plusieurs différences structurelles importantes.

L'approche traditionnelle correspond à la logique des algorithmes traditionnels : le problème est au niveau des jambes, donc les jambes doivent être modifiées, en suivant le chemin « cause à effet » étape par étape, sans sauter d'étape. Résoudre un problème à la fois, irréversiblement. La prochaine fois que le même problème se produit, on recommence. Mon approche correspond à la logique de l'algorithme hyperdimensionnel : l'objectif est de ne pas traîner. Je ne résous pas la cause de surface « jambes trop longues » ; je trouve plutôt un point de contrôle global — la taille. Plier la taille une fois, les jambes se raccourcissent automatiquement, et le problème disparaît instantanément. Je ne modifie aucune métadonnée. Le pli à la taille n'est qu'un état temporaire, réversible et dynamique.

D'un point de vue des données, l'approche traditionnelle modifie les métadonnées — coudre les jambes plus courtes fait perdre définitivement les données originales. Mon approche ne modifie aucune métadonnée ; les dimensions originales du pantalon restent intactes, ajoutant seulement un état temporaire, réversible et dynamique — le pli. Le pli à la taille ne crée pas de nouvelles données ni ne détruit d'anciennes données ; il réarrange simplement les relations entre les données — raccourcissant la circonférence effective de la taille, changeant indirectement la longueur effective des jambes du pantalon.

Dans cet exemple, « les jambes du pantalon ne traînent pas par terre » est le résultat cible. La méthode traditionnelle part de la cause « jambes trop longues », modifie l'ourlet, et obtient finalement le résultat « ne pas traîner ». Ma méthode, au contraire, clarifie d'abord le résultat « ne pas traîner », puis cherche rétroactivement une entrée structurelle, et découvre finalement que plier la taille une fois rend le résultat valide. C'est la manifestation pratique de la « théorie effet-cause ». Il n'est pas nécessaire de marcher pas à pas de la cause à l'effet ; on peut plutôt déterminer la structure en arrière à partir de l'effet, rendant le chemin original inutile.

Cet exemple répond également à la question des « métadonnées ». Que sont les métadonnées ? Dans l'exemple du pantalon, les métadonnées sont les dimensions originales du pantalon : tour de taille, longueur, coupe, tissu. Ce sont les « attributs

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

essentiels » du pantalon. La donnée temporaire est le pli à la taille ; il ne change aucune des dimensions originales du pantalon, il plie seulement une section temporairement. L'approche traditionnelle modifie les métadonnées — la longueur du pantalon est définitivement raccourcie, les données originales sont perdues. Mon approche ne modifie aucune métadonnée — les dimensions originales du pantalon restent intactes, ajoutant seulement un état temporaire, réversible et dynamique.

Les dimensions originales de la taille, en tant que métadonnées, sont simultanément le « point de départ de multiples résultats » : elles supportent simultanément de multiples états de port, comme porter normalement, porter avec un pli à la taille, porter avec deux plis à la taille, etc. Elles sont aussi « le résultat de multiples points de départ » : elles sont déterminées conjointement par de multiples points de départ tels que le choix du tissu, la méthode de coupe, l'intention de conception, etc. Le pli à la taille, en tant que donnée temporaire, ne crée ni ne détruit de données ; il réarrange simplement les relations entre les données.

Ce n'est pas une astuce ordinaire, mais une sorte de pensée structurelle. Beaucoup de gens, voyant des jambes de pantalon trop longues, sont attirés par la manifestation des jambes, donc tout le traitement tourne autour des jambes. Mais si l'on considère la structure globale, le fait que les jambes soient trop longues n'est qu'une manifestation en aval ; un changement de la position de la taille peut affecter le placement effectif de l'ensemble du pantalon. C'est-à-dire que le problème ne doit pas nécessairement être résolu à l'endroit où il apparaît. De nombreux problèmes de faible dimension ont leur véritable point de contrôle à un niveau structurel plus élevé. Les algorithmes traditionnels ont tendance à continuer à calculer le long de la surface du problème, tandis que l'algorithme hyperdimensionnel recherche l'entrée structurelle.

Cela explique aussi pourquoi l'algorithme hyperdimensionnel n'est pas plus complexe, mais peut être plus simple. Une structure véritablement de haut niveau ne complexifie souvent pas le problème, mais rend le problème complexe simple à l'entrée correcte. Face à un problème complexe, les algorithmes traditionnels peuvent ajouter des étapes, ajouter des modèles, ajouter de la puissance de calcul, ajouter du stockage, ajouter du temps d'entraînement. L'algorithme hyperdimensionnel, cependant, recherche un nœud structurel ; une fois ce nœud correctement activé, le chemin initialement complexe peut devenir invalide. Le soi-disant « contournement du chemin » n'est pas de la paresse, mais une redéfinition de la nécessité du chemin.

Dans cet exemple : l'approche traditionnelle est « modifier le résultat le long du chemin », tandis que l'algorithme hyperdimensionnel est « changer l'entrée, rendant le chemin entier invalide ». La méthode conventionnelle rapièce continuellement au niveau du résultat, tandis que l'autre méthode change directement l'entrée structurelle, faisant

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

que des problèmes qui auraient autrement nécessité de multiples traitements soient restructurés au point de départ en une seule fois. La forme dominante des algorithmes traditionnels est « partir du point de départ, suivre le chemin pour obtenir le résultat », tandis que dans l'algorithme hyperdimensionnel, « le résultat lui-même peut devenir une entrée de chemin et participer à la génération de nouvelles structures ». L'approche traditionnelle est « une structure correspond à un résultat », tandis que l'algorithme hyperdimensionnel est « une structure porte de multiples états de résultat ».

Sixième partie : Une vision des données consistant à ne pas modifier les métadonnées

D'un point de vue des données, l'algorithme hyperdimensionnel a un principe très important : ne pas modifier les métadonnées, afin qu'elles puissent être appliquées à de multiples points de départ ou résultats.

Les métadonnées sont les attributs originaux, la structure de base et les informations ontologiques des données. Face à différents problèmes, les méthodes de traitement traditionnelles modifient souvent les données, copient les données, traitent les données, recalculent les données, ou établissent différents chemins pour différents résultats. Cette approche peut certainement résoudre des problèmes, mais au prix d'un traitement continu des données, d'une répétition continue des chemins et d'une complexité croissante du système.

L'algorithme hyperdimensionnel, au contraire, souligne que, tout en essayant de ne pas modifier les métadonnées, en changeant les entrées, les relations, les conditions et les modes d'activation, la même structure de métadonnées peut correspondre à de multiples points de départ et de multiples résultats. L'ontologie des données reste inchangée ; les relations changent. La structure de base reste inchangée ; le mode de présentation change. Les métadonnées restent intactes, mais peuvent s'adapter à différents états.

C'est ce que j'appelle « calculer une fois, utiliser une infinité de fois ». « Calculer une fois » n'est pas simplement une mise en cache d'un résultat, mais signifie qu'une fois qu'une structure est établie, elle n'a plus besoin de reconstruire un chemin complet pour chaque état. La même structure de données, activée par différentes entrées, peut présenter différents résultats. Ce n'est pas « traiter les données pour différents résultats », mais « utiliser les mêmes données pour porter de multiples résultats potentiels ». C'est aussi l'un des cœurs qui distinguent l'algorithme hyperdimensionnel des

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

algorithmes traditionnels. Les algorithmes traditionnels traitent les données le long d'un chemin ; l'algorithme hyperdimensionnel traite la structure globale des données, des entrées, des relations et des résultats.

Dans sa structure minimale, l'algorithme hyperdimensionnel peut être compris comme : un système qui, sans modifier les métadonnées, permet au même nœud de données de correspondre à de multiples chemins de résultats par des changements d'entrées structurelles. Cette définition ne cherche pas à contracter immédiatement l'algorithme hyperdimensionnel en une formule mathématique traditionnelle, mais à établir d'abord ses frontières structurelles. Ce n'est pas une fonction linéaire unique, ni une formule fixe, ni un simple mécanisme de mise en cache. C'est une structure relationnelle : les métadonnées restent stables, les entrées peuvent changer, les conditions peuvent changer, les relations peuvent changer, les résultats peuvent être présentés dans de multiples directions. C'est précisément dans cette structure que le calcul n'est plus une exécution d'étapes, mais une activation de relations.

D'un point de vue des données, l'essence de « ne pas modifier les métadonnées, les rendre applicables à de multiples points de départ ou résultats » est : l'ontologie des données reste inchangée ; le changement a lieu dans le « mode d'interprétation / entrée d'utilisation ». La structure traditionnelle est : lorsque le problème change, les données ou le chemin changent. Dans l'algorithme hyperdimensionnel : lorsque le problème change, la structure interprétative change, pas les données.

Septième partie : Redéfinir le « supercalcul »

Dans ce système, je dois clarifier un terme — le « supercalcul ».

Ce que j'entends par « supercalcul », ce n'est pas les superordinateurs au sens habituel. Pas plus de puces, plus de puissance de calcul, ou de plus grands centres de données. Ce que j'entends par « supercalcul », c'est l'« algorithme hyperdimensionnel ».

Le véritable supercalcul n'est pas nécessairement l'accumulation de puissance de calcul, mais possiblement un changement de paradigme de calcul. Lorsqu'un système repose encore sur le calcul répétitif, aussi puissant soit-il, il reste piégé dans les chemins. Lorsqu'un système peut structurellement réduire le calcul, contourner les chemins répétitifs et faire des résultats de nouvelles entrées, il commence à s'approcher du véritable calcul hyperdimensionnel.

En d'autres termes, le supercalcul traditionnel recherche « utiliser plus de puissance de calcul pour résoudre de plus grands problèmes ». L'algorithme hyperdimensionnel

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

recherche « utiliser une meilleure structure pour que les problèmes n'aient plus besoin d'une puissance de calcul aussi énorme ». Ces deux approches ne sont pas contradictoires, mais leurs directions diffèrent.

Si le supercalcul traditionnel représente la limite de la puissance de calcul, alors l'algorithme hyperdimensionnel représente un tournant dans la compréhension du calcul. Le véritable « supercalcul » n'est peut-être pas plus de puces, de plus grands centres de données, une consommation d'énergie plus élevée ou des modèles plus complexes. Le véritable « supercalcul » peut être la découverte d'une entrée structurelle, une élimination d'un chemin, un déploiement inverse à partir d'un nœud de résultat, une manière de rendre de multiples résultats simultanément valides sans modifier les métadonnées. Plus la puissance de calcul est grande, si elle reste piégée dans des chemins de faible dimension, elle n'est que puissante à l'intérieur de ces chemins ; une fois la structure élevée, même avec des opérations extrêmement simples, un niveau d'efficacité supérieur peut émerger.

Le soi-disant supercalcul est la limite de la puissance de calcul ; l'algorithme hyperdimensionnel est une redéfinition de la prémisse « le calcul doit-il reposer sur la puissance de calcul ? »

Huitième partie : Fondements empiriques — Validation par des systèmes multidomaines et critères d'un nouveau paradigme

L'algorithme hyperdimensionnel n'est pas une simple construction théorique. Il a déjà été validé dans plusieurs systèmes réels.

Mon système logistique ne nécessite pas de puissance de supercalcul ni de support cloud ; il accomplit un ordonnancement complexe avec des ressources relativement faibles. Les résultats peuvent être réutilisés, la structure peut être répétitivement adaptée, et il n'est pas nécessaire de tout recalculer à partir de zéro à chaque fois. Ne pas dépendre du supercalcul, ne pas dépendre du cloud, accomplir un ordonnancement complexe avec de faibles ressources — c'est une avancée établie et précieuse en ingénierie. Cela démontre que la puissance de calcul élevée n'est pas la seule solution ; la conception structurelle peut remplacer une partie de la puissance de calcul.

Mon système d'édition adopte également cette logique structurelle, avec une efficacité dépassant de loin tout système actuel. Mon système extrême de génération de pages web est également basé sur le même algorithme hyperdimensionnel, prouvant à plusieurs reprises dans de multiples domaines que la même structure peut être

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

stablement valide. Les employés de mon entreprise utilisent déjà ces systèmes, démontrant que cette structure peut fonctionner sans mon intervention personnelle continue.

La caractéristique commune de ces systèmes est : ils ne dépendent pas des chemins dominants à haute puissance de calcul, ne suivent pas d'étapes de calcul fixes, et rendent le résultat cible directement valide par des ajustements des entrées structurelles. Ce ne sont pas des succès ponctuels, ni des astuces localisées, mais l'apparition répétée de la même logique structurelle dans différents domaines.

Cela démontre que l'algorithme hyperdimensionnel n'est pas accidentel, mais une méthode structurelle déjà validée par de multiples systèmes. Ce n'est pas une « astuce personnelle », mais un paradigme de calcul stablement valide dans de multiples domaines tels que la logistique, l'édition et la génération de pages web.

Selon le critère « l'existence et la validité constituent un nouveau paradigme », l'algorithme hyperdimensionnel, en tant que nouveau paradigme structurel de calcul déjà validé systématiquement dans de multiples domaines, est déjà valide à l'heure actuelle. Je ne propose pas une hypothèse, mais une structure de calcul différente déjà opérationnelle dans de multiples systèmes. Je n'ai pas besoin de prouver qu'il est compris ; j'ai déjà prouvé qu'il est stablement valide dans différents systèmes. À l'heure actuelle, dans le cadre de mon application pratique, cette structure peut déjà être considérée comme un nouveau paradigme.

En ce qui concerne les critères pour juger l'algorithme hyperdimensionnel comme un nouveau paradigme : lorsqu'une structure est logiquement auto-cohérente et stablement valide dans de multiples systèmes, tout en possédant des différences irréductibles avec les méthodes existantes, elle possède déjà la base d'un nouveau paradigme. L'auto-cohérence est le point de départ, la preuve empirique est la fondation, et la différence structurelle est le cœur du paradigme. Selon le critère « l'existence constitue la validité », ce système est déjà un nouveau paradigme ; que le monde extérieur le reconnaisse n'affecte que la diffusion, pas la validité. C'est un paradigme structurel de calcul déjà établi dans des systèmes réels, dont la validité ne dépend pas de la reconnaissance ou du consensus extérieur.

La différence fondamentale entre l'algorithme hyperdimensionnel et les paradigmes de calcul dominants est : les algorithmes traditionnels reposent principalement sur le calcul direct ; une fois qu'un résultat est produit, il ne peut généralement pas participer rétroactivement à de nouvelles structures inférentielles. Dans l'algorithme hyperdimensionnel, le résultat n'est plus un point d'arrivée, mais un nœud structurel réutilisable. Sous certaines conditions, le résultat peut participer à de nouveaux chemins

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

inférentiels, réduisant ainsi le calcul répétitif et formant un réseau de calcul avec de multiples points de départ et de multiples chemins. Dans les structures de calcul traditionnelles, le résultat est typiquement un point d'arrivée ; dans la structure de l'algorithme hyperdimensionnel, le résultat lui-même peut devenir un nouveau point de départ, formant ainsi un réseau inférentiel à multiples chemins. L'algorithme hyperdimensionnel n'est pas une amélioration des algorithmes, mais un réexamen de la prémisse « un algorithme doit-il nécessairement exister ? »

Neuvième partie : Clarification des malentendus courants

Sur la base d'échanges préliminaires avec des lecteurs de différents domaines, j'anticipe les six malentendus suivants et les clarifie à l'avance, afin d'éviter que le concept ne soit prématurément forcé dans des cadres inappropriés lors de la diffusion.

Malentendu premier : L'algorithme hyperdimensionnel n'est-il pas juste de la programmation dynamique ou de la mise en cache ? Clarification : La programmation dynamique et la mise en cache réutilisent les résultats sous des « entrées identiques ». L'algorithme hyperdimensionnel permet d'inférer différents points de départ à partir d'un même résultat — c'est une différence essentielle. La mise en cache résout le recalcul du même problème ; l'algorithme hyperdimensionnel résout le déploiement de nouveaux problèmes à travers des structures de résultats.

Malentendu deuxième : Vous dites « aléatoire mais ordonné » — n'est-ce pas contradictoire ? Clarification : L'aléatoire et l'ordre peuvent coexister. La distribution des arbres dans une forêt est aléatoire, mais la densité globale et la structure des espèces sont ordonnées. L'aléatoire n'est pas le chaos, mais une manière d'organiser la structure. L'aléatoire dans l'algorithme hyperdimensionnel est intrinsèque et constitutif, fonctionnant en synergie avec l'ordre.

Malentendu troisième : Sans entrée ni sortie, comment vérifier la justesse du résultat ? Clarification : L'algorithme hyperdimensionnel ne rejette pas l'entrée et la sortie ; il soutient plutôt que le calcul n'a pas à fonctionner selon le mode linéaire entrée-sortie. Dans le mode de manifestation structurelle, la « validité » est déterminée par la cohérence structurelle, non par la correspondance entrée-sortie. Cela n'abandonne pas la vérification, mais propose un cadre de vérification différent de la tradition.

Malentendu quatrième : N'est-ce pas juste la théorie de la complexité ou la théorie du chaos ? Clarification : La théorie de la complexité et la théorie du chaos décrivent des « systèmes difficiles à prédire ». L'algorithme hyperdimensionnel tente de décrire des « systèmes qui peuvent être compris et guidés par des entrées structurelles » — non pas

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

abandonner la prédiction, mais changer le mode de prédiction. La théorie du chaos dit « la prédiction est difficile » ; l'algorithme hyperdimensionnel demande « si l'on peut rendre la prédiction inutile en changeant l'entrée ».

Malentendu cinquième : Vous dites « ne pas modifier les métadonnées », mais le pli à la taille n'est-il pas aussi une modification ? Clarification : Le pli à la taille est un état temporaire, pas une modification permanente des paramètres originaux. Les métadonnées (tour de taille, longueur, coupe, tissu) ne changent absolument pas. C'est la différence entre « superposition d'états » et « modification d'attributs ». Le pli est réversible, temporaire et ne change pas les attributs essentiels.

Malentendu sixième : L'algorithme hyperdimensionnel nie-t-il la valeur des algorithmes traditionnels ? Clarification : Absolument pas. Les algorithmes traditionnels ont été extrêmement importants dans l'histoire technologique humaine. Sans les algorithmes traditionnels, il n'y aurait pas d'ordinateurs modernes, de bases de données, de systèmes de communication, d'intelligence artificielle, de simulation en ingénierie ou de supercalcul. La valeur des algorithmes traditionnels ne peut être niée. Cependant, reconnaître la valeur des algorithmes traditionnels ne signifie pas les reconnaître comme le point d'arrivée des algorithmes. Les algorithmes traditionnels résolvent des problèmes d'efficacité à l'intérieur d'un paradigme de calcul donné, tandis que l'algorithme hyperdimensionnel pose la question du paradigme de calcul lui-même. L'un concerne comment mieux marcher ; l'autre demande si l'on a besoin d'emprunter ce chemin.

Dixième partie : L'algorithme hyperdimensionnel et l'histoire du calcul humain

Pour aider les lecteurs à mieux positionner l'algorithme hyperdimensionnel dans l'histoire du calcul humain, je propose ici un bref aperçu 宏观.

La compréhension humaine du « calcul » a traversé plusieurs étapes. La première étape était le calcul manuel ; les algorithmes existaient comme des étapes humaines, lents et sujets aux erreurs, mais à travers ce processus, les humains ont compris les règles de base du calcul. La deuxième étape était le calcul mécanique, de la machine de Pascal à la machine analytique de Babbage ; le calcul a été mécanisé, mais les algorithmes dépendaient encore des structures physiques. La troisième étape était le calcul électronique et le paradigme de Turing ; l'architecture de von Neumann s'est répandue ; les algorithmes ont été codés en programmes ; la machine de Turing est devenue la frontière de la calculabilité. La quatrième étape est le calcul parallèle et le

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

supercalcul, accumulant la puissance de calcul par un grand nombre d'unités de calcul pour s'attaquer à des problèmes plus complexes, mais la logique sous-jacente reste une extension du paradigme de Turing.

Actuellement, l'humanité est à l'entrée de la cinquième étape. Le calcul quantique tente de briser les limites des bits classiques ; le calcul neuromorphique tente d'imiter la structure des systèmes nerveux biologiques ; et l'algorithme hyperdimensionnel tente de briser le cadre linéaire « entrée → étapes → sortie » lui-même.

La relation entre l'algorithme hyperdimensionnel et les algorithmes traditionnels n'est pas un remplacement, mais une hiérarchie. Les algorithmes traditionnels résolvent le problème de « comment calculer plus efficacement sur un chemin donné ». L'algorithme hyperdimensionnel demande « si le chemin peut être redéfini, ou même contourné ». Si nous considérons l'histoire du calcul humain comme un processus de rupture constante de ses propres frontières, alors l'algorithme hyperdimensionnel est précisément un nouveau nœud dans ce processus. Il ne résout pas les problèmes à l'intérieur de l'ancien cadre, mais propose un nouveau cadre.

Ce jugement n'implique pas que l'algorithme hyperdimensionnel soit déjà mature ou complet. Bien au contraire. En tant que nouveau concept, sa définition, ses frontières, ses méthodes de vérification et ses scénarios d'application sont encore en cours de formation. Le but de cet article est précisément d'ancrer l'origine de ce concept, fournissant une base théorique stable pour le développement ultérieur.

Onzième partie : Conclusion — Ce n'est pas une optimisation, mais une redéfinition

La différence entre l'algorithme hyperdimensionnel et les algorithmes traditionnels n'est pas une différence de « meilleur » contre « pire », ni une différence de « plus rapide » contre « plus lent », mais une différence fondamentale au niveau du paradigme.

Les algorithmes traditionnels recherchent un contrôle prévisible. Vous me donnez l'entrée, je sais quelle sortie vous obtiendrez, parce que j'ai calculé chaque étape. C'est le paradigme de l'ingénieur : concevoir, construire, tester, vérifier.

L'algorithme hyperdimensionnel décrit une émergence compréhensible. Je ne peux pas prédire précisément chaque état intermédiaire, mais je sais que le modèle global se présentera de manière ordonnée. Chaque donnée est vivante et a une « perspective ». C'est plutôt le paradigme de l'écologiste ou du théoricien des systèmes complexes : observer, comprendre, guider, utiliser l'ordre émergent.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Les algorithmes traditionnels modifient progressivement le résultat le long d'un chemin donné. L'algorithme hyperdimensionnel rend le résultat cible immédiatement valide en changeant l'entrée structurelle, contournant ainsi tout le chemin.

Les algorithmes traditionnels « traitent les données pour différents résultats ». L'algorithme hyperdimensionnel est « utiliser les mêmes données pour porter de multiples résultats potentiels ».

Les algorithmes traditionnels recherchent « calculer correctement une fois ». L'algorithme hyperdimensionnel recherche « calculer correctement une fois, puis ne plus jamais avoir à calculer ».

Ce n'est pas une optimisation des algorithmes, mais un réexamen de la prémisse « un algorithme doit-il nécessairement exister ? »

Par conséquent, l'algorithme hyperdimensionnel n'est pas une optimisation d'algorithme, mais une redéfinition de l'algorithme. Il ne s'agit pas de courir plus vite sur la piste traditionnelle des algorithmes, mais de demander s'il existe un autre chemin en dehors de la piste traditionnelle, ou même si l'on peut être indépendant du chemin lui-même. Il ne s'agit pas de prouver que les algorithmes traditionnels sont inutiles, mais de souligner que les algorithmes traditionnels ne sont qu'une forme de faible dimension de l'algorithme. Il ne demande pas « comment obtenir des résultats en moins de temps », mais « si un résultat peut être directement établi par la structure ». Il ne demande pas « comment traiter plus de données », mais « si les mêmes données peuvent porter plus de résultats potentiels ».

Dans le paradigme de calcul dominant, l'algorithme hyperdimensionnel peut être considéré comme incalculable, invérifiable, ou difficile à incorporer dans les frontières existantes de la théorie du calcul. Mais cela constitue précisément sa différence paradigmatique. Le fait qu'un nouveau concept apparaisse instable à l'intérieur d'un ancien paradigme ne signifie pas nécessairement qu'il est dénué de sens ; cela peut aussi signifier que l'ancien paradigme lui-même ne peut pas pleinement l'accueillir. À l'heure actuelle, l'algorithme hyperdimensionnel est avant tout une proposition structurelle, une définition originale d'une nouvelle vision du calcul, plutôt qu'un système mature avec une boucle d'ingénierie complétée. Il nécessite une poursuite continue par des ajouts, des développements, des vérifications et des applications, mais son origine conceptuelle doit d'abord être clairement articulée.

En fin de compte, ce vers quoi l'algorithme hyperdimensionnel pointe est une nouvelle vision du calcul : un algorithme n'est pas nécessairement juste des mathématiques, pas nécessairement juste des étapes, pas nécessairement juste un programme, pas

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

nécessairement juste un processus de traitement entre l'entrée et la sortie. Un algorithme peut aussi être une manière d'organiser des relations multidimensionnelles, un réseau structurel de données, d'entrées, de relations, d'états et de résultats. Il peut contenir l'ordre et l'aléatoire ; il peut aller de la cause à l'effet, ou générer rétroactivement des causes à partir d'effets ; il peut s'adapter à de multiples états sans modifier les métadonnées ; il peut faire d'un résultat un nouveau point de départ, ou avoir de multiples points de départ convergeant vers le même résultat.

Un algorithme véritablement avancé n'est pas nécessairement un calcul plus complexe, mais une méthode d'organisation multidimensionnelle qui peut réduire le calcul, rendre les structures vivantes, faire des résultats des entrées, et permettre à la cause et à l'effet de former des cycles.

C'est le sens fondamental de ma proposition de l'« algorithme hyperdimensionnel ». Ce n'est pas un nouveau terme ordinaire, mais un nouveau concept, un nouveau paradigme, une nouvelle structure de calcul déjà établie dans de multiples systèmes réels. Son objectif n'est pas de calculer plus vite de manière répétée, mais de réduire le calcul répétitif ; non pas d'accumuler la puissance de calcul, mais de restructurer les entrées ; non pas de modifier constamment les données, mais de garder les métadonnées inchangées tout en changeant les relations ; non pas de laisser les résultats s'arrêter à des points d'arrivée, mais de faire des résultats de nouveaux points de départ. Les algorithmes traditionnels cherchent des résultats le long de chemins ; l'algorithme hyperdimensionnel active des résultats dans des structures. Les algorithmes traditionnels cherchent à compléter le calcul ; l'algorithme hyperdimensionnel demande si le calcul doit exister sous sa forme traditionnelle. C'est la différence la plus fondamentale entre lui et la définition actuelle de l'algorithme.

Annexe : Frontières de cet article et questions ouvertes

Cet article propose un cadre préliminaire pour l'algorithme hyperdimensionnel, non une définition formelle complète. Les questions suivantes attendent un développement ultérieur. Elles ne constituent pas une négation des définitions de cet article, mais précisément les directions pour les prochaines étapes.

Premièrement, les conditions de convergence. Quel est le marqueur d'« achèvement » de l'algorithme hyperdimensionnel ? Comment détermine-t-on si un résultat est « valide » ? Dans les algorithmes traditionnels, la réponse est définie par la correspondance entrée-sortie. Dans l'algorithme hyperdimensionnel, la réponse peut devoir être définie par la cohérence structurelle. Cette définition n'est pas encore complète.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Deuxièmement, la représentation des ressources. Comment les états superposés multidimensionnels peuvent-ils être représentés dans des ressources finies ? Les mesures de ressources traditionnelles telles que le temps, l'espace et la puissance de calcul sont-elles toujours applicables ? Si oui, comment sont-elles redéfinies ? Sinon, quelles mesures les remplacent ? Ces questions attendent une exploration.

Troisièmement, la garantie de l'ordre. Par quelles règles l'« ordre » dans « aléatoire mais ordonné » est-il garanti ? Où se trouve la frontière entre l'aléatoire et l'ordre ? Dans quelles circonstances l'aléatoire perturbe-t-il l'ordre ? Dans quelles circonstances l'ordre supprime-t-il l'aléatoire ? Ces questions nécessitent des recherches supplémentaires.

Quatrièmement, la sélection dans l'inférence inverse. Lors de l'inférence de multiples points de départ à partir d'un résultat, comment sélectionner ou évaluer les mérites relatifs des différents points de départ ? Existe-t-il une mesure d'« efficacité d'inférence inverse » ? Si oui, quelle est sa relation avec la mesure d'efficacité du calcul direct ?

Cinquièmement, les interfaces avec les systèmes traditionnels. Comment l'algorithme hyperdimensionnel fonctionne-t-il avec les systèmes existants basés sur la machine de Turing ? S'agit-il d'un remplacement, d'un complément ou d'une couche hiérarchique ? En ingénierie pratique, comment la frontière entre l'algorithme hyperdimensionnel et les algorithmes traditionnels est-elle délimitée ? Des modes hybrides existent-ils ?

Sixièmement, le cadre de vérification. Comment les résultats de l'algorithme hyperdimensionnel sont-ils vérifiés ? Si le résultat n'est pas obtenu par des étapes linéaires, comment établir la reproductibilité et la testabilité ? Une philosophie de vérification complètement différente est-elle nécessaire ?

Ces questions ne sont pas des défauts de cet article, mais les caractéristiques inévitables de cet article en tant que « document d'ancrage ». La proposition de tout nouveau paradigme s'accompagne de questions ouvertes. Cet article est un début, pas une conclusion.

Note bibliographique

L'« algorithme hyperdimensionnel » proposé dans cet article n'est pas une extension directe d'aucune école de pensée académique existante. Cependant, en termes d'idées, les domaines suivants fournissent un contexte dialogique pour cet article, offert seulement pour l'orientation des lecteurs, et ne sont pas des citations au sens académique traditionnel.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

La machine de Turing et la théorie de la calculabilité servent de référence par rapport à laquelle cet article compare les algorithmes traditionnels. Les problèmes inverses et l'inférence bayésienne servent de tentatives existantes d'« inférer les causes à partir des résultats » ; la « théorie effet-cause » de cet article est liée mais diffère en direction. La théorie de l'émergence et les systèmes complexes servent de descriptions existantes des « structures manifestant des résultats » ; l'algorithme hyperdimensionnel tente de rendre l'émergence « compréhensible » et « guidable ». Les graphes de connaissances et les bases de données graphes servent de pratiques existantes de « données se croisant à de multiples nœuds » ; l'algorithme hyperdimensionnel, sur cette fondation, priorise la dimension des « entrées variables ». Les paradigmes de calcul non classiques, y compris le calcul quantique, le calcul analogique, le calcul neuromorphique, etc., brisent les bits classiques ou les architectures classiques, tandis que l'algorithme hyperdimensionnel se concentre davantage sur la rupture du cadre linéaire « entrée → étapes → sortie » lui-même.

La relation entre cet article et les domaines ci-dessus est de dialogue et de transcendance, non d'héritage ou de réfutation. L'objectif de cet article n'est pas de faire des améliorations incrémentales à l'intérieur de ces domaines, mais de soulever un nouveau niveau de questionnement au-dessus d'eux.

Articles connexes

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [Diffusion] Je n'ai pas d'algorithme, pourtant je dépasse les algorithmes !
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [Philosophie extrême] Théorie effet-cause
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [Diffusion extrême] Rejeter la capture algorithmique
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

[Algoritmo Extremo] Algoritmo Hiperdimensional

Una redefinición del "cálculo" en sí mismo

Autor: Wu Zhaohui JEFFI CHAO HUI WU

Resumen

Este artículo propone el nuevo concepto de "algoritmo hiperdimensional" como un reexamen estructural de los límites de la definición tradicional de algoritmo. Los algoritmos tradicionales suelen construirse sobre el marco lineal de entrada, pasos, reglas y salida, enfatizando la finitud, el determinismo, la eficacia, la viabilidad y los límites claros entre entrada y salida. Aunque las supercomputadoras modernas poseen una potencia computacional extremadamente alta, su lógica subyacente sigue siendo principalmente la ejecución de paradigmas algorítmicos establecidos a mayor escala. Este artículo sostiene que lo que realmente merece discusión en el "supercálculo" no es solo el crecimiento de la escala computacional, sino un cambio fundamental en el paradigma computacional mismo. El algoritmo hiperdimensional no es una versión acelerada de los algoritmos tradicionales, ni una extensión de los algoritmos matemáticos; es una visión estructural del cálculo multidimensional, superpuesta, aleatoria pero ordenada. En esta estructura, los datos ya no son solo una entrada pasiva o una salida terminal, sino un atributo de nodo que posee simultáneamente múltiples puntos de partida y múltiples resultados; un resultado ya no es solo un punto final, sino que puede convertirse en una nueva entrada; los metadatos no necesitan modificarse repetidamente, sino que pueden corresponder a diferentes estados y resultados a través de puntos de entrada estructurales, cambios relacionales y activación condicional. Este artículo utiliza la filosofía de la "teoría efecto-causa" y el ejemplo cotidiano de "doblar la cintura de un pantalón" para ilustrar cómo el algoritmo hiperdimensional puede hacer que un resultado objetivo sea directamente válido cambiando el punto de entrada estructural, reduciendo así el cálculo repetitivo e incluso evitando la ruta computacional original. Este artículo tiene como objetivo establecer una definición preliminar, límites estructurales e ideas fundamentales para el "algoritmo hiperdimensional", proporcionando un nodo teórico original para la futura expansión del cálculo hiperdimensional, los algoritmos extremos y un nuevo sistema científico. Según el criterio de "la existencia y validez constituyen un nuevo paradigma", el algoritmo hiperdimensional, como nuevo paradigma estructural computacional ya validado sistemáticamente en múltiples dominios, es ya válido en el presente.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية:.
Chinesische Originalfassung maßgeblich.

Palabras clave: Algoritmo hiperdimensional; Algoritmo extremo; Cálculo hiperdimensional; Teoría efecto-causa; Metadatos; Paradigma computacional; Punto de entrada estructural; Causalidad no lineal; Nueva ciencia

Introducción: Por qué necesitamos rediscutir el "algoritmo"

Cuando propongo el "algoritmo hiperdimensional", no es para dar a los algoritmos existentes un nombre más grande, ni para empaquetar los algoritmos tradicionales en algún concepto novedoso. Al contrario, quiero discutir una cuestión más fundamental: ¿Está la comprensión humana actual del "algoritmo" confinada en un marco demasiado estrecho?

Hoy, cuando la gente habla de algoritmos, suele pensar en fórmulas matemáticas, código de programa, pasos lógicos, entrada-salida, entrenamiento de modelos, procesamiento de datos, optimización de rutas, clasificación de búsquedas, recomendación automática y razonamiento de inteligencia artificial. Estas son ciertamente formas importantes de algoritmos, y sustentan el funcionamiento de las computadoras modernas, Internet, las bases de datos, la inteligencia artificial, la supercomputación, los sistemas industriales y la sociedad de la información. Sin embargo, si el algoritmo se entiende solo como "un conjunto de pasos ordenados", solo como un proceso que "comienza con una entrada, pasa por un procesamiento basado en reglas, y termina con una salida", entonces esta comprensión ya confina el cálculo dentro de un marco de baja dimensión.

Vivimos en una era dominada por los algoritmos. Desde los motores de búsqueda hasta los sistemas de recomendación, desde la predicción meteorológica hasta la inteligencia artificial, los límites de los algoritmos son los límites de la inteligencia humana. Pero hay una pregunta que rara vez se hace: ¿Pueden los algoritmos ser solo de esta manera? ¿Por qué el cálculo debe seguir el patrón lineal de "entrada → pasos → salida"? ¿Por qué el mismo problema debe calcularse desde cero cada vez? Este artículo intenta desafiar estos supuestos predeterminados y proponer un paradigma computacional completamente diferente — el algoritmo hiperdimensional. Según el criterio de "la existencia y validez constituyen un nuevo paradigma", el algoritmo hiperdimensional, como nuevo paradigma estructural computacional ya validado sistemáticamente en múltiples dominios, es ya válido en el presente.

Es importante señalar que el "algoritmo hiperdimensional" propuesto en este artículo es un concepto completamente diferente del "cálculo hiperdimensional" (Hyperdimensional Computing, HDC) que ha existido en el mundo académico durante casi treinta años. El HDC utiliza vectores binarios aleatorios de dimensión

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

extremadamente alta para codificación y cálculo, buscando una representación y un cálculo más eficientes, pero aún permanece dentro del paradigma lineal de "entrada → procesamiento → salida". El algoritmo hiperdimensional es totalmente diferente: cuestiona si un resultado puede establecerse directamente a través de la estructura, si la ruta computacional misma puede ser evitada, si los metadatos pueden permanecer sin modificación mientras se adaptan a múltiples resultados. Sus nombres son similares, pero sus connotaciones son opuestas. El algoritmo hiperdimensional no es una mejora de los algoritmos, sino un replanteamiento de la premisa de "si un algoritmo debe existir necesariamente".

Primera parte: La definición estándar actual de "algoritmo"

En los sistemas dominantes de la informática, las matemáticas y la ingeniería, existe una definición básica de algoritmo aceptada durante mucho tiempo: un algoritmo es un proceso computacional bien definido, que consiste en un número finito de pasos, que toma una o más entradas, las transforma mediante reglas deterministas, y produce una o más salidas en un tiempo finito. Esta comprensión no es la opinión casual de nadie, sino un consenso fundamental formado a lo largo del largo desarrollo de la civilización computacional moderna. Sustenta la lógica subyacente del software, hardware, bases de datos, sistemas en red, modelos de inteligencia artificial y centros de supercómputo. Por muy complejo que parezca un algoritmo, su núcleo sigue girando en torno a la entrada, las reglas, los pasos y la salida.

Esta concepción estándar del algoritmo puede descomponerse en varias características fundamentales.

Primero, finitud. Un algoritmo debe terminar en un número finito de pasos; no puede repetirse indefinidamente ni ejecutarse eternamente. Un proceso que nunca se detiene, incluso si puede describirse formalmente, difícilmente puede considerarse un algoritmo efectivo. Todos los programas de cálculo científico y flujos de procesamiento de datos tienen condiciones de terminación claras.

Segundo, determinismo. Dada la misma entrada en las mismas condiciones, el algoritmo debe producir la misma salida. Incluso si algunos algoritmos introducen aleatoriedad, típicamente es una aleatoriedad controlada, reproducible, o un mecanismo probabilístico interpretable estadísticamente, no una aleatoriedad intrínseca completamente inmanejable. Los resultados de las simulaciones numéricas deben ser reproducibles; este es un requisito fundamental del cálculo científico.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Tercero, límites claros entre entrada y salida. La entrada viene primero, el procesamiento en medio, la salida al final; el punto de partida y el punto final tienen límites estructurales claros. Cualesquiera sean los datos que le dé al algoritmo, después del procesamiento, le devuelve un resultado; este límite es distintivo. Entrada primero, procesamiento después, salida al final; el orden no puede invertirse.

Cuarto, eficacia. Cada paso de un algoritmo debe ser una operación prácticamente ejecutable; no puede contener saltos imposibles de ejecutar, describir o verificar. Cada paso debe ser una operación básica que un humano podría realizar con papel y lápiz, o que una computadora podría ejecutar realmente — suma, resta, multiplicación, división, juicios lógicos, lectura/escritura de datos, etc.

Quinto, viabilidad. Incluso si un algoritmo es teóricamente válido, si el tiempo, la potencia computacional, la memoria o los recursos de almacenamiento que consume son totalmente inaceptables, es difícil considerarlo un algoritmo efectivo en términos de ingeniería. Un algoritmo teóricamente correcto pero que tardaría cientos de millones de años en completarse no se considera un algoritmo viable en ingeniería.

Este conjunto completo de conceptos algorítmicos se remonta en última instancia al modelo de la máquina de Turing. Como abstracción central de la teoría computacional moderna, la máquina de Turing define el límite básico de la "computabilidad". Por muy potentes que sean las computadoras actuales, por muy avanzados que sean los chips, por muy grande que sea la escala de los centros de supercómputo, su lógica subyacente nunca ha abandonado verdaderamente este camino: entrada, reglas, pasos, salida. Las supercomputadoras modernas solo aceleran la ejecución de este proceso mediante una mayor escala de hardware, mayor paralelismo y una programación de sistemas más compleja. Es decir, en el contexto tradicional, el "supercálculo" es principalmente una expansión de la escala computacional, no necesariamente un cambio fundamental en el paradigma computacional. La potencia computacional puede aumentar miles o decenas de miles de veces, pero si la lógica subyacente sigue siendo "entrada, pasos, salida", entonces permanece dentro del paradigma algorítmico tradicional.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Segunda parte: Definición del "algoritmo hiperdimensional"

El "algoritmo hiperdimensional" que propongo emerge precisamente de este contexto. No es una versión mejorada de los algoritmos tradicionales, ni un algoritmo más rápido, ni una fórmula matemática más compleja, ni una técnica de programación más avanzada. Es una comprensión estructural completamente diferente.

Primero, necesitamos explicar el significado del término "hiperdimensional". "Hiperdimensional" tiene aquí dos significados. El primer significado es que no se limita a una secuencia lineal unidimensional de pasos, sino que permite que múltiples dimensiones se desplieguen simultáneamente. El segundo significado es que intenta trascender los límites definicionales tradicionales del "cálculo" — un algoritmo ya no tiene que ser necesariamente matemático, secuencial o determinista. Un algoritmo tradicional es como conducir por una calle de un solo sentido: debes comenzar desde el punto A, pasar por B, C, D para llegar a Z. El algoritmo hiperdimensional no cree que el cálculo tenga que ser una calle de un solo sentido. Cree que el cálculo puede ser una red, un campo, una estructura multidimensional, donde cualquier nodo puede ser un punto de entrada y cualquier nodo puede ser un punto de salida.

En mi sistema de Nueva Ciencia, un algoritmo no es necesariamente un algoritmo matemático, no es necesariamente un único cálculo ordenado, no está necesariamente obligado a ejecutarse a lo largo de pasos fijos, ni está necesariamente sujeto estrictamente al patrón lineal de "entrada, procesamiento, salida". El algoritmo hiperdimensional es una estructura multidimensional, superpuesta, aleatoria pero ordenada. En esta estructura, cada dato no es ni una entrada aislada ni una salida fija, sino que posee múltiples roles simultáneamente: puede ser tanto el punto de partida para múltiples resultados como el resultado producido por la acción conjunta de múltiples puntos de partida.

En otras palabras, los algoritmos tradicionales colocan los datos dentro de un flujo de proceso, mientras que el algoritmo hiperdimensional coloca los datos dentro de una estructura. En los algoritmos tradicionales, los datos suelen dividirse en datos de entrada, datos intermedios y datos de salida. Cada tipo de dato tiene una posición y un rol claros. La entrada es la entrada, el resultado es el resultado, el proceso intermedio es el proceso intermedio. Sin embargo, en el algoritmo hiperdimensional, un dato no tiene una sola identidad. Puede ser un resultado en una dirección y un punto de partida en otra; puede ser un objeto invocado en una estructura y convertirse en un

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

punto de entrada que desencadena nuevos resultados en otra estructura. El dato ya no es solo material pasivo, sino un nodo relacional multidimensional.

Esto corresponde precisamente al núcleo del algoritmo hiperdimensional: cada dato es el punto de partida para múltiples resultados y también el resultado de múltiples puntos de partida. Los algoritmos tradicionales utilizan "pasos ordenados" para producir un solo resultado; el algoritmo hiperdimensional se acerca más a una estructura de estados multidimensional donde, en la superposición de aleatoriedad y orden, el resultado no se calcula para existir, sino que se manifiesta naturalmente dentro de la estructura. En este sistema, el dato es tanto el punto de partida como un componente del resultado.

Para presentar más claramente las características fundamentales del algoritmo hiperdimensional, las descompongo en los siguientes cinco aspectos.

Primero, naturaleza no matemática. Los algoritmos dominantes son esencialmente matemáticos — pueden describirse completamente mediante lógica simbólica y fórmulas matemáticas. El algoritmo hiperdimensional no es necesariamente matemático. Puede basarse en principios físicos, relaciones geométricas, un nuevo paradigma de la teoría de la información, o incluso principios de la conciencia. El cálculo no tiene que realizarse necesariamente mediante "operaciones matemáticas"; puede presentarse naturalmente a través de relaciones geométricas en un espacio de mayor dimensión. Por ejemplo, la forma de una burbuja de jabón está determinada por el principio de minimización de la tensión superficial. Esto no es un "algoritmo matemático" calculando — es la física misma "calculando". El algoritmo hiperdimensional intenta comprender este tipo de "cálculo", en lugar de usar fórmulas matemáticas para simularlo.

Segundo, multidimensionalidad y superposición. Los algoritmos tradicionales ejecutan pasos ordenados en una sola dimensión, con un solo estado en cada paso. El algoritmo hiperdimensional permite que múltiples dimensiones existan simultáneamente, con múltiples estados superponiéndose y coexistiendo. El algoritmo no sigue un solo camino; más bien, se despliega dentro de un campo de estados multidimensional. El resultado no se "calcula" sino que "se manifiesta" desde este campo. Por ejemplo, cuando se forma un copo de nieve en el aire, no hay un "algoritmo" diciéndole a cada molécula de agua a dónde ir. La disposición de las moléculas de agua es el resultado de múltiples dimensiones — temperatura, humedad, flujo de aire — actuando juntas. La forma del copo de nieve "se manifiesta" en lugar de ser "calculada".

Tercero, aleatoriedad dentro del orden. En los algoritmos tradicionales, la aleatoriedad es instrumental, externa, una perturbación controlable; el algoritmo mismo es

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

determinista. En el algoritmo hiperdimensional, la aleatoriedad es intrínseca y estructural. La aleatoriedad y el orden coexisten y trabajan juntos, constituyendo conjuntamente la esencia del algoritmo. Esto no es "un algoritmo con aleatoriedad", sino "la aleatoriedad misma es un componente orgánico del algoritmo". Por ejemplo, la estructura ecológica de un bosque — la distribución de los árboles parece aleatoria, pero en general presenta una densidad ordenada y relaciones entre especies. Esa "aleatoriedad" no es un error, sino un requisito previo para el funcionamiento normal de la estructura ecológica.

Cuarto, los múltiples roles de los datos. En los algoritmos tradicionales, el rol de los datos es singular — la entrada es la entrada, la salida es la salida, el resultado intermedio es el resultado intermedio. En el algoritmo hiperdimensional, cada dato es simultáneamente el punto de partida para múltiples resultados y el resultado de múltiples puntos de partida. Un nodo de datos puede desplegarse aguas abajo en múltiples rutas de resultados, y también puede ser convergido aguas arriba por múltiples causas. El dato ya no es un punto en un flujo lineal, sino un nodo de convergencia en una red. Por ejemplo, considere la altura de una persona. En un algoritmo tradicional, la altura es una "entrada" — usted la introduce para obtener salidas como predicción de peso, talla de ropa, etc. Pero la altura también es un "resultado" — está determinada conjuntamente por múltiples "puntos de partida" como la genética, la nutrición y el ejercicio. En el algoritmo hiperdimensional, el dato de la altura es simultáneamente un punto de partida y un resultado, no una elección binaria.

Quinto, metadatos sin cambios, relaciones variables. Frente a diferentes problemas, los algoritmos tradicionales o modifican los datos mismos o recalculan todo. El algoritmo hiperdimensional no modifica los metadatos — los atributos esenciales de los datos permanecen sin cambios. Lo que cambia son los puntos de entrada, los modos de interpretación y las relaciones entre los datos. La misma estructura de metadatos, activada a través de diferentes puntos de entrada, puede presentar resultados completamente diferentes. Esto significa: calcule una vez, use infinitas veces. Por ejemplo, considere el mismo pasaje de texto. Puede leerlo como poesía, decodificarlo como un cifrado, o analizarlo como un documento histórico. El texto mismo no ha cambiado — usted solo ha cambiado el "punto de entrada". El algoritmo hiperdimensional persigue precisamente esta capacidad: "mismos datos, múltiples puntos de entrada, múltiples resultados".

Desde la perspectiva de la estructura de datos, los metadatos mismos no se modifican; más bien, a través de diferentes puntos de entrada y condiciones, corresponden a múltiples puntos de partida y resultados dentro de la misma estructura. Los datos ya no

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

se procesan repetidamente, sino que, manteniendo su estructura sin cambios, se activan a través de diferentes puntos de entrada, presentando así diferentes resultados. El enfoque tradicional es "procesar datos para diferentes resultados", mientras que el algoritmo hiperdimensional es "usar los mismos datos para portar múltiples resultados potenciales".

Tercera parte: Diferencias fundamentales entre el algoritmo hiperdimensional y los algoritmos tradicionales

Con base en las definiciones anteriores, el algoritmo hiperdimensional y los algoritmos tradicionales exhiben diferencias fundamentales en múltiples dimensiones centrales. Estas se detallan a continuación una por una.

En términos de naturaleza esencial, los algoritmos tradicionales son matemáticos y formales, capaces de describirse completamente mediante lógica simbólica y fórmulas matemáticas; son esencialmente objetos matemáticos. El algoritmo hiperdimensional no es necesariamente matemático; puede basarse en principios físicos, geométricos, informacionales o incluso conscientes. No es un subconjunto de las matemáticas, sino una categoría más amplia. Esta diferencia puede resumirse como: del "objeto matemático" a la "ley universal".

En términos de orden temporal, los algoritmos tradicionales siguen un orden temporal singular, ordenado y lineal, paso A a B a C, estrictamente secuencial, o descomponible en subpasos ordenados paralelos; la flecha del tiempo es irreversible. El algoritmo hiperdimensional no tiene una secuencia de pasos fija; es multidimensional, superpuesto, aleatorio pero ordenado. El resultado puede no estar relacionado con el "orden" de ejecución, porque no existe un "orden" tradicional en primer lugar. Esta diferencia puede resumirse como: del "tiempo lineal" a la "simultaneidad de dimensión superior".

En términos de causalidad, los algoritmos tradicionales siguen una cadena causal determinista: dada la misma entrada y el mismo estado inicial, la salida es siempre única; la causa precede al efecto, una flecha unidireccional irreversible. El algoritmo hiperdimensional sigue una causalidad superpuesta: cada dato es el punto de partida para múltiples resultados y el resultado de múltiples puntos de partida. Esta es mi filosofía de la "teoría efecto-causa" — es posible tener primero el resultado, luego la causa. El resultado no es un punto final, sino que puede convertirse en un nuevo punto de partida. Esta diferencia puede resumirse como: de la "causa única, efecto único" a la "red causal totalmente interconectada".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

En términos de estructura de datos, los algoritmos tradicionales siguen una estructura de flujo unidireccional desde la entrada al procesamiento y luego a la salida; los datos tienen un rol único con límites claros; los datos de entrada siguen siendo entradas de principio a fin, los datos de resultado siguen siendo resultados. En el algoritmo hiperdimensional, los datos tienen múltiples roles; el mismo dato es simultáneamente un resultado y un punto de partida; no hay puntos de partida absolutos ni puntos finales absolutos, solo nodos en una red. Esta diferencia puede resumirse como: de la "fluidez unidireccional" a la "simbiosis recursiva".

En términos de modo computacional, los algoritmos tradicionales calculan desde cero cada vez que encuentran un problema, incluso si han calculado problemas similares mil veces antes, incluso si la respuesta ya se conoce; en la milésima primera vez, aún recorren todo el proceso. Esto es cálculo "sin estado". En el algoritmo hiperdimensional, si existe un resultado correspondiente, no es necesario volver a recorrerlo; a partir de un resultado, se pueden inferir múltiples puntos de partida, y las rutas causales pueden desplegarse en reversa. El cálculo es con estado, tiene memoria y es reutilizable. Esta diferencia puede resumirse como: de "recalcular cada vez" a "reutilizar una vez".

En términos del papel de la aleatoriedad, la aleatoriedad en los algoritmos tradicionales es pseudoaleatoria o inyectada externamente; los números aleatorios son meras herramientas para muestreo, aproximación u optimización; el algoritmo mismo es determinista. La aleatoriedad en el algoritmo hiperdimensional es intrínseca y constitutiva; la aleatoriedad es un componente orgánico, coexistiendo y cooperando con el orden. No es "un algoritmo contiene aleatoriedad", sino "la aleatoriedad es una razón por la cual el algoritmo puede funcionar". Esta diferencia puede resumirse como: de la "aleatoriedad instrumental" a la "aleatoriedad constitutiva".

En términos de la comprensión de los recursos, los algoritmos tradicionales buscan un cálculo más rápido y preciso dados los recursos; los recursos son restricciones; el objetivo del algoritmo es "completar el cálculo dentro de las restricciones". El algoritmo hiperdimensional busca hacer que el cálculo mismo sea innecesario mediante el diseño estructural; no "calcular más rápido", sino "evitar el cálculo"; los recursos no se utilizan para ser "consumidos", sino para "diseñar puntos de entrada". Esta diferencia puede resumirse como: de la "optimización bajo restricciones de recursos" a la "eliminación mediante el diseño estructural".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Cuarta parte: La filosofía de la teoría efecto-causa

Con base en la comparación anterior, necesito desarrollar más la filosofía de la "teoría efecto-causa". Esta es una de las ideas fundamentales centrales del algoritmo hiperdimensional.

El pensamiento tradicional sostiene típicamente que la causa precede al efecto; primero la causa, luego el efecto; primero la entrada, luego la salida. Este concepto se manifiesta en los algoritmos como: debes comenzar desde el punto de partida y caminar paso a paso hasta el punto final. Incluso si sabes cuál es la respuesta, no puedes "usar" directamente esa respuesta — porque careces del "camino de prueba".

En mi estructura, un resultado no es necesariamente solo un punto final. Una vez que aparece un resultado, puede convertirse en un nuevo punto de partida y continuar participando en la generación de nuevas estructuras. Un resultado puede participar retroactivamente en la formación de causas. Múltiples puntos de partida posibles pueden inferirse a partir de un resultado. La causa y el efecto ya no se disponen en una secuencia unidireccional, sino que forman relaciones cíclicas, relaciones de red, relaciones multidimensionales.

En otras palabras, los algoritmos tradicionales preguntan: "Dada la causa, ¿cómo obtenemos el resultado?" El algoritmo hiperdimensional pregunta: "Dado el resultado, ¿cómo inferimos o construimos la causa?"

Es importante aclarar que "tener primero el resultado, luego la causa" no es una negación de la causalidad, ni sostiene que "los resultados surgen de la nada". Más bien, ilustra que, en estructuras de dimensión superior, la causa y el efecto pueden servir como puntos de entrada entre sí y transformarse mutuamente. Como en el siguiente ejemplo del pantalón, el resultado "que se pueda usar sin arrastrar por el suelo" se determina primero, y luego encuentro retroactivamente el punto de operación de "doblar la cintura", que corresponde al nivel de la "causa" dentro de la estructura. No es que el resultado no tenga causa, sino que el resultado se convierte en un nuevo punto de entrada para descubrir la causa.

Una suposición fundamental en el mundo de los algoritmos tradicionales es que el tiempo es unidireccional; debes calcular paso a paso desde el estado inicial hasta el estado final. Incluso si sabes cuál es la respuesta, no puedes "usar" directamente esa respuesta, porque careces del camino de prueba. El algoritmo hiperdimensional desafía precisamente esta suposición: si se conoce el resultado, las causas pueden inferirse hacia atrás; el mismo resultado puede corresponder a múltiples rutas causales; el

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

cálculo ya no es caminar desde el principio hasta el final, sino desplegar todos los puntos de partida posibles desde el final.

Quinta parte: Un ejemplo cotidiano — pantalón y doblar la cintura

Para comprender más intuitivamente las diferencias abstractas anteriores, usaré un ejemplo cotidiano.

Una persona compra un pantalón nuevo con cinturilla elástica. Las perneras del pantalón son un poco demasiado largas, arrastrando por el suelo, lo que es incómodo.

El enfoque tradicional: al notar que las perneras son demasiado largas, se remanga una sección de la pernera, se cose con aguja e hilo, y luego se prueba. Si el niño crece y el pantalón se vuelve demasiado corto, la pernera cosida no se puede alargar, y el pantalón se arruina. La próxima vez, se compra un pantalón nuevo y se cose de nuevo. Este método parece muy natural, porque el problema parece estar en las perneras, por lo que se aborda las perneras. Corresponde al pensamiento de los algoritmos tradicionales: identificar el problema, localizar la manifestación, procesar en la ubicación de la manifestación, y finalmente obtener un resultado. ¿Perneras demasiado largas? Entonces modificar las perneras. ¿Datos incorrectos? Entonces modificar los datos. ¿Ruta inapropiada? Entonces continuar parcheando a lo largo de la ruta.

Mi enfoque es diferente. En lugar de modificar el dobladillo del pantalón, doblo la cintura una vez, y el pantalón es inmediatamente utilizable. Esta acción es muy simple, pero la lógica estructural detrás de ella es completamente diferente. No corto las perneras, no coso el dobladillo permanentemente, no cambio la longitud original del pantalón y no daño los metadatos del pantalón. La circunferencia de la cintura, la longitud del pantalón, el estilo y la tela no sufren ningún cambio esencial. Simplemente cambio el punto de entrada estructural de la cintura, haciendo que las perneras se acorten naturalmente al usarlo. La cintura no es aquí una simple localidad, sino un punto de control global para todo el estado de uso. Al ajustar este punto de control, el problema aguas abajo desaparece directamente.

Más importante aún, este método es reversible, ajustable y reutilizable. Esto es especialmente evidente para un niño en crecimiento. Si el niño no es lo suficientemente alto y las perneras son un poco largas, doblo la cintura una vez; después de algún tiempo, cuando el niño ha crecido, suelto la cintura un poco; cuando ha crecido aún más, la suelto completamente. La estructura original del pantalón permanece intacta, sin embargo puede adaptarse a la altura del niño en diferentes etapas. Con el método

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

tradicional, si se cose el dobladillo permanentemente, cuando el niño crece, el pantalón puede volverse demasiado corto; si se corta, es aún más imposible de restaurar. Mi método mantiene los metadatos sin cambios, permitiendo que la misma estructura se adapte a múltiples estados.

Este ejemplo revela varias diferencias estructurales importantes.

El enfoque tradicional corresponde a la lógica de los algoritmos tradicionales: el problema está en las perneras, por lo que las perneras deben modificarse, siguiendo la ruta "causa a efecto" paso a paso, sin saltarse ningún paso. Resolver un problema a la vez, irreversiblemente. La próxima vez que ocurra el mismo problema, hacerlo todo de nuevo. Mi enfoque corresponde a la lógica del algoritmo hiperdimensional: el objetivo es que no arrastre. No resuelvo la causa superficial de "perneras demasiado largas"; en cambio, encuentro un punto de control global — la cintura. Doblar la cintura una vez, las perneras se acortan automáticamente, y el problema desaparece instantáneamente. No modifico ningún metadato. El pliegue en la cintura es solo un estado temporal, reversible y dinámico.

Desde una perspectiva de datos, el enfoque tradicional modifica los metadatos — coser las perneras más cortas pierde permanentemente los datos originales. Mi enfoque no modifica ningún metadato; las dimensiones originales del pantalón permanecen intactas, agregando solo un estado temporal, reversible y dinámico — el pliegue. El pliegue en la cintura no crea nuevos datos ni destruye datos antiguos; simplemente reorganiza las relaciones entre los datos — acortando la circunferencia efectiva de la cintura, cambiando indirectamente la longitud efectiva de las perneras.

En este ejemplo, "las perneras no arrastran por el suelo" es el resultado objetivo. El método tradicional parte de la causa "perneras demasiado largas", modifica el dobladillo, y finalmente obtiene el resultado "no arrastrar". Mi método, por el contrario, aclara primero el resultado "no arrastrar", luego busca hacia atrás un punto de entrada estructural, y finalmente descubre que doblar la cintura una vez hace que el resultado sea válido. Esta es la manifestación práctica de la "teoría efecto-causa". No es necesario caminar paso a paso desde la causa hasta el efecto; más bien, se puede determinar la estructura hacia atrás desde el efecto, haciendo innecesaria la ruta original.

Este ejemplo también aborda la cuestión de los "metadatos". ¿Qué son los metadatos? En el ejemplo del pantalón, los metadatos son las dimensiones originales del pantalón: circunferencia de la cintura, longitud, estilo, tela. Estos son los "atributos esenciales" del pantalón. El dato temporal es el pliegue en la cintura; no cambia ninguna de las dimensiones originales del pantalón, solo pliega una sección temporalmente. El enfoque tradicional modifica los metadatos — la longitud del pantalón se acorta

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

permanentemente, los datos originales se pierden. Mi enfoque no modifica ningún metadato — las dimensiones originales del pantalón permanecen intactas, solo se agrega un estado temporal, reversible y dinámico.

Las dimensiones originales de la cintura, como metadatos, son simultáneamente el "punto de partida para múltiples resultados": soportan simultáneamente múltiples estados de uso, como usar normalmente, usar con un pliegue en la cintura, usar con dos pliegues en la cintura, etc. También son "el resultado de múltiples puntos de partida": están determinadas conjuntamente por múltiples puntos de partida como la selección de la tela, el método de corte, la intención del diseño, etc. El pliegue en la cintura, como dato temporal, no crea nuevos datos ni destruye datos antiguos; simplemente reorganiza las relaciones entre los datos.

Esto no es un truco ordinario, sino un tipo de pensamiento estructural. Muchas personas, al ver las perneras demasiado largas, son atraídas por la manifestación de las perneras, por lo que todo el procesamiento gira en torno a las perneras. Pero si se ve desde la estructura global, que las perneras sean demasiado largas es meramente una manifestación aguas abajo; un cambio en la posición de la cintura puede afectar la ubicación efectiva de todo el pantalón. Es decir, el problema no necesariamente tiene que resolverse donde aparece. Muchos problemas de baja dimensión tienen su verdadero punto de control en un nivel estructural más alto. Los algoritmos tradicionales tienden a continuar calculando a lo largo de la superficie del problema, mientras que el algoritmo hiperdimensional busca el punto de entrada estructural.

Esto también explica por qué el algoritmo hiperdimensional no es más complejo, sino que puede ser más simple. Una estructura verdaderamente de alto nivel a menudo no complica el problema, sino que hace que el problema complejo sea simple en el punto de entrada correcto. Frente a un problema complejo, los algoritmos tradicionales pueden agregar pasos, agregar modelos, agregar potencia computacional, agregar almacenamiento, agregar tiempo de entrenamiento. El algoritmo hiperdimensional, sin embargo, busca un nodo estructural; una vez que este nodo se activa correctamente, la ruta originalmente compleja puede volverse inválida. El llamado "evitar la ruta" no es pereza, sino redefinir si la ruta es necesaria.

En este ejemplo: el enfoque tradicional es "modificar el resultado a lo largo de la ruta", mientras que el algoritmo hiperdimensional es "cambiar el punto de entrada, haciendo que toda la ruta sea inválida". El método convencional parchea continuamente en el extremo del resultado, mientras que el otro método cambia directamente el punto de entrada estructural, haciendo que problemas que de otro modo requerirían múltiples tratamientos sean reestructurados en el punto de partida de una sola vez. La forma dominante de los algoritmos tradicionales es "comenzar desde el punto de partida,

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

seguir la ruta para obtener el resultado", mientras que en el algoritmo hiperdimensional, "el resultado mismo puede convertirse en un punto de entrada de ruta y participar en la generación de nuevas estructuras". El enfoque tradicional es "una estructura corresponde a un resultado", mientras que el algoritmo hiperdimensional es "una estructura porta múltiples estados de resultado".

Sexta parte: Una visión de datos de no modificar metadatos

Desde una perspectiva de datos, el algoritmo hiperdimensional tiene un principio muy importante: no modificar los metadatos, para que puedan aplicarse a múltiples puntos de partida o resultados.

Los metadatos son los atributos originales, la estructura básica y la información ontológica de los datos. Frente a diferentes problemas, los métodos de procesamiento tradicionales a menudo modifican los datos, copian los datos, procesan los datos, recalculan los datos o establecen diferentes rutas para diferentes resultados. Este enfoque ciertamente puede resolver problemas, pero a costa de que los datos se procesen continuamente, las rutas se repitan continuamente y la complejidad del sistema aumente constantemente.

El algoritmo hiperdimensional, por el contrario, enfatiza que, mientras se intenta no modificar los metadatos, cambiando los puntos de entrada, las relaciones, las condiciones y los modos de activación, la misma estructura de metadatos puede corresponder a múltiples puntos de partida y múltiples resultados. La ontología de los datos permanece sin cambios; las relaciones cambian. La estructura básica permanece sin cambios; el modo de presentación cambia. Los metadatos permanecen intactos, pero pueden adaptarse a diferentes estados.

Esto es lo que llamo "calcular una vez, usar infinitas veces". "Calcular una vez" no es simplemente almacenar en caché un resultado, sino que significa que una vez que se establece una estructura, ya no necesita reconstruir una ruta completa para cada estado. La misma estructura de datos, activada a través de diferentes puntos de entrada, puede presentar diferentes resultados. No es "procesar datos para diferentes resultados", sino "usar los mismos datos para portar múltiples resultados potenciales". Este es también uno de los núcleos que distinguen el algoritmo hiperdimensional de los algoritmos tradicionales. Los algoritmos tradicionales procesan datos a lo largo de una ruta; el algoritmo hiperdimensional procesa la estructura global de datos, puntos de entrada, relaciones y resultados.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

En su estructura mínima, el algoritmo hiperdimensional puede entenderse como: un sistema que, sin modificar los metadatos, permite que el mismo nodo de datos corresponda a múltiples rutas de resultados a través de cambios en los puntos de entrada estructurales. Esta definición no busca contraer inmediatamente el algoritmo hiperdimensional en una fórmula matemática tradicional, sino establecer primero sus límites estructurales. No es una función lineal única, ni una fórmula fija, ni un simple mecanismo de caché. Es una estructura relacional: los metadatos permanecen estables, los puntos de entrada pueden cambiar, las condiciones pueden cambiar, las relaciones pueden cambiar, los resultados pueden presentarse en múltiples direcciones. Es precisamente en esta estructura que el cálculo ya no se trata de ejecutar pasos, sino de activar relaciones.

Desde una perspectiva de datos, la esencia de "no modificar los metadatos, haciéndolos aplicables a múltiples puntos de partida o resultados" es: la ontología de los datos permanece sin cambios; el cambio ocurre en el "modo de interpretación / punto de entrada de uso". La estructura tradicional es: cuando el problema cambia, los datos o la ruta cambian. En el algoritmo hiperdimensional: cuando el problema cambia, la estructura interpretativa cambia, no los datos.

Séptima parte: Redefiniendo el "supercálculo"

En este sistema, necesito aclarar un término — el "supercálculo".

Lo que quiero decir con "supercálculo" no son las supercomputadoras en el sentido habitual. No más chips, mayor potencia computacional o centros de datos más grandes. Lo que quiero decir con "supercálculo" es el "algoritmo hiperdimensional".

El verdadero supercálculo no es necesariamente la acumulación de potencia computacional, sino posiblemente un cambio en el paradigma computacional. Cuando un sistema todavía depende del cálculo repetitivo, por muy potente que sea, permanece atrapado dentro de rutas. Cuando un sistema puede reducir estructuralmente el cálculo, evitar rutas repetitivas y convertir los resultados en nuevos puntos de entrada, comienza a acercarse al verdadero cálculo hiperdimensional.

En otras palabras, el supercálculo tradicional busca "usar más potencia computacional para resolver problemas más grandes". El algoritmo hiperdimensional busca "usar una mejor estructura para que los problemas ya no necesiten una potencia computacional tan enorme". Estas dos no son contradictorias, pero sus direcciones difieren.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Si el supercálculo tradicional representa el límite de la potencia computacional, entonces el algoritmo hiperdimensional representa un giro en la comprensión del cálculo. El verdadero "supercálculo" puede no ser más chips, centros de datos más grandes, mayor consumo de energía o modelos más complejos. El verdadero "supercálculo" puede ser el descubrimiento de un punto de entrada estructural, una eliminación de una ruta, un despliegue inverso desde un nodo de resultado, una forma de hacer que múltiples resultados sean simultáneamente válidos sin modificar los metadatos. Cuanto más potente es la potencia computacional, si todavía está atrapada en rutas de baja dimensión, sigue siendo solo potente dentro de esas rutas; una vez que la estructura se eleva, incluso con operaciones extremadamente simples, puede emerger un nivel superior de eficiencia.

El llamado supercálculo es el límite de la potencia computacional; el algoritmo hiperdimensional es una redefinición de la premisa de "si el cálculo debe depender de la potencia computacional".

Octava parte: Fundamentos empíricos — Validación por sistemas multidominio y criterios de nuevo paradigma

El algoritmo hiperdimensional no es una construcción puramente teórica. Ya ha sido validado en varios sistemas del mundo real.

Mi sistema logístico no requiere potencia de supercómputo ni soporte en la nube; realiza una programación compleja con recursos relativamente bajos. Los resultados pueden reutilizarse, la estructura puede adaptarse repetidamente, y no es necesario recalcular todo desde cero cada vez. No depender del supercómputo, no depender de la nube, realizar una programación compleja con bajos recursos — este es un avance establecido y valioso en ingeniería. Demuestra que la alta potencia computacional no es la única solución; el diseño estructural puede reemplazar una parte de la potencia computacional.

Mi sistema editorial adopta igualmente esta lógica estructural, con una eficiencia que supera con creces a cualquier sistema actual. Mi sistema extremo de generación de páginas web también se basa en el mismo algoritmo hiperdimensional, probando repetidamente en múltiples dominios que la misma estructura puede ser establemente válida. Los empleados de mi empresa ya están utilizando estos sistemas, demostrando que esta estructura puede operar sin mi intervención personal continua.

La característica común de estos sistemas es: no dependen de las rutas dominantes de alta potencia computacional, no siguen pasos de cálculo fijos, y hacen que el resultado

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

objetivo sea directamente válido mediante ajustes en los puntos de entrada estructurales. No son éxitos únicos, ni trucos puntuales, sino la aparición repetida de la misma lógica estructural en diferentes dominios.

Esto demuestra que el algoritmo hiperdimensional no es accidental, sino un método estructural ya validado por múltiples sistemas. No es un "truco personal", sino un paradigma computacional que es establemente válido en múltiples dominios como la logística, la edición y la generación de páginas web.

Según el criterio de "la existencia y validez constituyen un nuevo paradigma", el algoritmo hiperdimensional, como nuevo paradigma estructural computacional ya validado sistemáticamente en múltiples dominios, es ya válido en el presente. No estoy proponiendo una hipótesis, sino una estructura computacional diferente ya operativa en múltiples sistemas. No necesito probar que es comprendido; ya he probado que es establemente válido en diferentes sistemas. En el presente, dentro del alcance de mi aplicación práctica, esta estructura ya puede considerarse un nuevo paradigma.

En cuanto a los criterios para juzgar el algoritmo hiperdimensional como un nuevo paradigma: cuando una estructura es lógicamente autoconsistente y establemente válida en múltiples sistemas, al tiempo que posee diferencias irreducibles con los métodos existentes, ya posee la base de un nuevo paradigma. La autoconsistencia es el punto de partida, la evidencia empírica es el fundamento, y la diferencia estructural es el núcleo del paradigma. Bajo el estándar de "la existencia constituye validez", este sistema es ya un nuevo paradigma; si el mundo exterior lo reconoce solo afecta la difusión, no la validez. Este es un paradigma estructural computacional ya establecido en sistemas reales, cuya validez no depende del reconocimiento o consenso externo.

La diferencia fundamental entre el algoritmo hiperdimensional y los paradigmas computacionales dominantes es: los algoritmos tradicionales se basan principalmente en el cálculo hacia adelante; una vez que se produce un resultado, generalmente no puede participar retroactivamente en nuevas estructuras inferenciales. En el algoritmo hiperdimensional, el resultado ya no es un punto final, sino un nodo estructural reutilizable. Bajo ciertas condiciones, el resultado puede participar en nuevas rutas inferenciales, reduciendo así el cálculo repetitivo y formando una red computacional con múltiples puntos de partida y múltiples rutas. En las estructuras computacionales tradicionales, el resultado es típicamente un punto final; en la estructura del algoritmo hiperdimensional, el resultado mismo puede convertirse en un nuevo punto de partida, formando así una red inferencial de múltiples rutas. El algoritmo hiperdimensional no es una mejora de los algoritmos, sino un replanteamiento de la premisa de "si un algoritmo debe existir necesariamente".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

Novena parte: Aclaración de malentendidos comunes

Con base en intercambios preliminares con lectores de diferentes campos, anticipo los siguientes seis malentendidos y los aclaro de antemano, para evitar que el concepto sea prematuramente forzado en marcos inapropiados durante la difusión.

Malentendido primero: ¿No es el algoritmo hiperdimensional solo programación dinámica o almacenamiento en caché? Aclaración: La programación dinámica y el almacenamiento en caché reutilizan resultados bajo "entradas idénticas". El algoritmo hiperdimensional permite inferir diferentes puntos de partida a partir de un solo resultado — esta es una diferencia esencial. El almacenamiento en caché resuelve el recálculo del mismo problema; el algoritmo hiperdimensional resuelve el despliegue de nuevos problemas a través de estructuras de resultados.

Malentendido segundo: Usted dice "aleatorio pero ordenado" — ¿no es eso contradictorio? Aclaración: La aleatoriedad y el orden pueden coexistir. La distribución de los árboles en un bosque es aleatoria, pero la densidad general y la estructura de especies son ordenadas. La aleatoriedad no es caos, sino una forma de organizar la estructura. La aleatoriedad en el algoritmo hiperdimensional es intrínseca y constitutiva, funcionando sinérgicamente con el orden.

Malentendido tercero: Sin entrada ni salida, ¿cómo se verifica la corrección del resultado? Aclaración: El algoritmo hiperdimensional no rechaza la entrada y la salida; más bien sostiene que el cálculo no tiene que operar en el modo lineal de entrada-salida. En el modo de manifestación estructural, la "validez" está determinada por la consistencia estructural, no por la correspondencia entrada-salida. Esto no abandona la verificación, sino que propone un marco de verificación diferente al tradicional.

Malentendido cuarto: ¿No es esto solo teoría de la complejidad o teoría del caos? Aclaración: La teoría de la complejidad y la teoría del caos describen "sistemas difíciles de predecir". El algoritmo hiperdimensional intenta describir "sistemas que pueden ser comprendidos y guiados a través de puntos de entrada estructurales" — no abandonar la predicción, sino cambiar el modo de predicción. La teoría del caos dice "la predicción es difícil"; el algoritmo hiperdimensional pregunta "si se puede hacer que la predicción sea innecesaria cambiando el punto de entrada".

Malentendido quinto: Usted dice "no modificar los metadatos", pero el pliegue en la cintura, ¿no es también una modificación? Aclaración: El pliegue en la cintura es un estado temporal, no una modificación permanente de los parámetros originales. Los metadatos (circunferencia de la cintura, longitud, estilo, tela) no cambian en absoluto.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Esta es la diferencia entre "superposición de estados" y "modificación de atributos". El pliegue es reversible, temporal y no cambia los atributos esenciales.

Malentendido sexto: ¿Niega el algoritmo hiperdimensional el valor de los algoritmos tradicionales? Aclaración: Absolutamente no. Los algoritmos tradicionales han sido extremadamente importantes en la historia tecnológica humana. Sin los algoritmos tradicionales, no habría computadoras modernas, bases de datos, sistemas de comunicación, inteligencia artificial, simulación en ingeniería o supercómputo. El valor de los algoritmos tradicionales no puede ser negado. Sin embargo, reconocer el valor de los algoritmos tradicionales no significa reconocerlos como el punto final de los algoritmos. Los algoritmos tradicionales resuelven problemas de eficiencia dentro de un paradigma computacional dado, mientras que el algoritmo hiperdimensional plantea la cuestión del paradigma computacional mismo. Uno trata de cómo caminar mejor; el otro pregunta si uno necesita tomar ese camino en absoluto.

Décima parte: El algoritmo hiperdimensional y la historia del cálculo humano

Para ayudar a los lectores a posicionar mejor el algoritmo hiperdimensional dentro de la historia del cálculo humano, ofrezco aquí un breve resumen macro.

La comprensión humana del "cálculo" ha pasado por múltiples etapas. La primera etapa fue el cálculo manual; los algoritmos existían como pasos humanos, lentos y propensos a errores, pero a través de este proceso, los humanos comprendieron las reglas básicas del cálculo. La segunda etapa fue el cálculo mecánico, desde la calculadora de Pascal hasta la máquina analítica de Babbage; el cálculo se mecanizó, pero los algoritmos todavía dependían de estructuras físicas. La tercera etapa fue el cálculo electrónico y el paradigma de Turing; la arquitectura de von Neumann se generalizó; los algoritmos se codificaron como programas; la máquina de Turing se convirtió en el límite de la computabilidad. La cuarta etapa es el cálculo paralelo y el supercálculo, acumulando potencia computacional a través de un gran número de unidades de cálculo para abordar problemas más complejos, pero la lógica subyacente sigue siendo una extensión del paradigma de Turing.

Actualmente, la humanidad está en la entrada de la quinta etapa. El cálculo cuántico intenta romper las limitaciones de los bits clásicos; el cálculo neuromórfico intenta imitar la estructura de los sistemas nerviosos biológicos; y el algoritmo hiperdimensional intenta romper el marco lineal de "entrada → pasos → salida" mismo.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

La relación entre el algoritmo hiperdimensional y los algoritmos tradicionales no es de reemplazo, sino de jerarquía. Los algoritmos tradicionales resuelven el problema de "cómo calcular más eficientemente en una ruta dada". El algoritmo hiperdimensional pregunta "si la ruta puede ser redefinida, o incluso evitada". Si consideramos la historia del cálculo humano como un proceso de ruptura constante de sus propios límites, entonces el algoritmo hiperdimensional es precisamente un nuevo nodo en este proceso. No resuelve problemas dentro del viejo marco, sino que propone un nuevo marco.

Este juicio no implica que el algoritmo hiperdimensional sea ya maduro o completo. Todo lo contrario. Como concepto nuevo, su definición, límites, métodos de verificación y escenarios de aplicación aún están en proceso de formación. El propósito de este artículo es precisamente anclar el origen de este concepto, proporcionando una base teórica estable para el desarrollo posterior.

Undécima parte: Conclusión — Esto no es optimización, sino redefinición

La diferencia entre el algoritmo hiperdimensional y los algoritmos tradicionales no es una diferencia de "mejor" versus "peor", ni una diferencia de "más rápido" versus "más lento", sino una diferencia fundamental a nivel de paradigma.

Los algoritmos tradicionales buscan un control predecible. Usted me da la entrada, yo sé qué salida obtendrá, porque he calculado cada paso. Es el paradigma del ingeniero: diseñar, construir, probar, verificar.

El algoritmo hiperdimensional describe una emergencia comprensible. No puedo predecir con precisión cada estado intermedio, pero sé que el patrón global se presentará de alguna manera ordenada. Cada dato está vivo y tiene una "perspectiva". Es más como el paradigma de un ecologista o teórico de sistemas complejos: observar, comprender, guiar, utilizar el orden emergente.

Los algoritmos tradicionales modifican gradualmente el resultado a lo largo de una ruta dada. El algoritmo hiperdimensional hace que el resultado objetivo sea inmediatamente válido cambiando el punto de entrada estructural, evitando así toda la ruta.

Los algoritmos tradicionales "procesan datos para diferentes resultados". El algoritmo hiperdimensional es "usar los mismos datos para portar múltiples resultados potenciales".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Los algoritmos tradicionales buscan "calcular correctamente una vez". El algoritmo hiperdimensional busca "calcular correctamente una vez, y luego nunca más tener que calcular".

Esto no es una optimización de los algoritmos, sino un replanteamiento de la premisa de "si un algoritmo debe existir necesariamente".

Por lo tanto, el algoritmo hiperdimensional no es una optimización de algoritmos, sino una redefinición del algoritmo. No se trata de correr más rápido en la pista tradicional de algoritmos, sino de preguntar si hay otro camino fuera de la pista tradicional, o incluso si se puede ser independiente del camino mismo. No se trata de probar que los algoritmos tradicionales son inútiles, sino de señalar que los algoritmos tradicionales son solo una forma de baja dimensión del algoritmo. No pregunta "cómo obtener resultados en menos tiempo", sino "si un resultado puede establecerse directamente a través de la estructura". No pregunta "cómo procesar más datos", sino "si los mismos datos pueden portar más resultados potenciales".

Dentro del paradigma computacional dominante, el algoritmo hiperdimensional puede ser visto como incalculable, inverificable, o difícil de incorporar dentro de los límites existentes de la teoría computacional. Pero esto constituye precisamente su diferencia paradigmática. El hecho de que un nuevo concepto aparezca inestable dentro de un viejo paradigma no significa necesariamente que carezca de sentido; también puede significar que el viejo paradigma mismo no puede acomodarlo completamente. En el presente, el algoritmo hiperdimensional es ante todo una proposición estructural, una definición original de una nueva visión del cálculo, más que un sistema maduro con un bucle de ingeniería completado. Requiere una continuación continua mediante adiciones, desarrollos, verificaciones y aplicaciones, pero su origen conceptual debe ser articulado claramente primero.

En última instancia, hacia lo que apunta el algoritmo hiperdimensional es una nueva visión del cálculo: un algoritmo no es necesariamente solo matemáticas, no es necesariamente solo pasos, no es necesariamente solo un programa, no es necesariamente solo un proceso de procesamiento entre entrada y salida. Un algoritmo también puede ser una forma de organizar relaciones multidimensionales, una red estructural de datos, puntos de entrada, relaciones, estados y resultados. Puede contener orden y aleatoriedad; puede ir de la causa al efecto, o generar retroactivamente causas a partir de efectos; puede adaptarse a múltiples estados sin modificar metadatos; puede hacer de un resultado un nuevo punto de partida, o tener múltiples puntos de partida convergiendo en el mismo resultado.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Un algoritmo verdaderamente avanzado no es necesariamente un cálculo más complejo, sino un método organizacional multidimensional que puede reducir el cálculo, dar vida a las estructuras, convertir los resultados en puntos de entrada y permitir que la causa y el efecto formen ciclos.

Este es el significado central de mi propuesta del "algoritmo hiperdimensional". No es un término nuevo ordinario, sino un nuevo concepto, un nuevo paradigma, una nueva estructura computacional ya establecida en múltiples sistemas reales. Su enfoque no es calcular repetidamente más rápido, sino reducir el cálculo repetitivo; no acumular potencia computacional, sino reestructurar los puntos de entrada; no modificar constantemente los datos, sino mantener los metadatos sin cambios mientras se cambian las relaciones; no dejar que los resultados se detengan en puntos finales, sino hacer que los resultados se conviertan en nuevos puntos de partida. Los algoritmos tradicionales buscan resultados a lo largo de rutas; el algoritmo hiperdimensional activa resultados dentro de estructuras. Los algoritmos tradicionales buscan completar el cálculo; el algoritmo hiperdimensional cuestiona si el cálculo debe existir en su forma tradicional. Esta es la diferencia más fundamental entre él y la definición actual de algoritmo.

Apéndice: Límites de este artículo y preguntas abiertas

Este artículo propone un marco preliminar para el algoritmo hiperdimensional, no una definición formal completa. Las siguientes preguntas esperan desarrollo posterior. No constituyen una negación de las definiciones de este artículo, sino precisamente las direcciones para los próximos pasos.

Primero, condiciones de convergencia. ¿Cuál es la marca de "finalización" del algoritmo hiperdimensional? ¿Cómo se determina si un resultado es "válido"? En los algoritmos tradicionales, la respuesta se define por la correspondencia entrada-salida. En el algoritmo hiperdimensional, la respuesta puede necesitar ser definida por la consistencia estructural. Esta definición aún no está completa.

Segundo, representación de recursos. ¿Cómo pueden representarse los estados superpuestos multidimensionales dentro de recursos finitos? ¿Siguen siendo aplicables las medidas de recursos tradicionales como tiempo, espacio y potencia computacional? Si es así, ¿cómo se redefinen? Si no, ¿qué medidas las reemplazan? Estas preguntas esperan exploración.

Tercero, garantía del orden. ¿Por qué reglas se garantiza el "orden" en "aleatorio pero ordenado"? ¿Dónde está el límite entre aleatoriedad y orden? ¿En qué circunstancias la

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

aleatoriedad perturba el orden? ¿En qué circunstancias el orden suprime la aleatoriedad? Estas preguntas requieren investigación adicional.

Cuarto, selección en la inferencia inversa. Al inferir múltiples puntos de partida hacia atrás a partir de un resultado, ¿cómo se seleccionan o evalúan los méritos relativos de los diferentes puntos de partida? ¿Existe alguna medida de "eficiencia de inferencia inversa"? Si es así, ¿cuál es su relación con la medida de eficiencia del cálculo hacia adelante?

Quinto, interfaces con sistemas tradicionales. ¿Cómo funciona el algoritmo hiperdimensional con los sistemas existentes basados en la máquina de Turing? ¿Es un reemplazo, un suplemento o una capa jerárquica? En ingeniería práctica, ¿cómo se delimita el límite entre el algoritmo hiperdimensional y los algoritmos tradicionales? ¿Existen modos híbridos?

Sexto, marco de verificación. ¿Cómo se verifican los resultados del algoritmo hiperdimensional? Si el resultado no se obtiene a través de pasos lineales, ¿cómo se establecen la reproducibilidad y la verificabilidad? ¿Se requiere una filosofía de verificación completamente diferente?

Estas preguntas no son defectos de este artículo, sino las características inevitables de este artículo como "documento ancla". La propuesta de cualquier nuevo paradigma viene acompañada de preguntas abiertas. Este artículo es un comienzo, no una conclusión.

Nota bibliográfica

El "algoritmo hiperdimensional" propuesto en este artículo no es una extensión directa de ninguna escuela de pensamiento académico existente. Sin embargo, en términos de ideas, los siguientes campos proporcionan un contexto dialógico para este artículo, ofrecido solo para orientación de los lectores, y no son citas en el sentido académico tradicional.

La máquina de Turing y la teoría de la computabilidad sirven como el punto de referencia con el que este artículo compara los algoritmos tradicionales. Los problemas inversos y la inferencia bayesiana sirven como intentos existentes de "inferir causas a partir de resultados"; la "teoría efecto-causa" de este artículo está relacionada pero difiere en dirección. La teoría de la emergencia y los sistemas complejos sirven como descripciones existentes de "estructuras que manifiestan resultados"; el algoritmo hiperdimensional intenta hacer que la emergencia sea "comprensible" y "guiable". Los

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

grafos de conocimiento y las bases de datos de grafos sirven como prácticas existentes de "datos que se cruzan en múltiples nodos"; el algoritmo hiperdimensional, sobre esta base, prioriza la dimensión de los "puntos de entrada variables". Los paradigmas computacionales no clásicos, incluyendo el cálculo cuántico, el cálculo analógico, el cálculo neuromórfico, etc., rompen los bits clásicos o las arquitecturas clásicas, mientras que el algoritmo hiperdimensional se centra más en romper el marco lineal "entrada → pasos → salida" mismo.

La relación entre este artículo y los campos anteriores es de diálogo y trascendencia, no de herencia o refutación. El objetivo de este artículo no es hacer mejoras incrementales dentro de estos campos, sino plantear un nuevo nivel de cuestionamiento por encima de ellos.

Artículos relacionados

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [Difusión] ¡No tengo algoritmo, sin embargo supero a los algoritmos!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [Filosofía extrema] Teoría efecto-causa
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [Difusión extrema] Rechazar la captura algorítmica
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[極限アルゴリズム] 超次元アルゴリズム

「計算」そのものの再定義

著者: ウー・ジャオホイ JEFFI CHAO HUI WU

要旨

本稿は、「超次元アルゴリズム」という新しい概念を、伝統的なアルゴリズム定義の境界に対する構造的な再検討として提案する。伝統的なアルゴリズムは通常、入力、ステップ、ルール、出力という線形の枠組みの上に構築され、有限性、決定性、実行可能性、実現可能性、そして明確な入出力境界を強調する。現代のスーパーコンピュータは極めて高い計算能力を持つが、その基礎となる論理は依然として主に、確立されたアルゴリズムパラダイムをより大規模に実行するものである。本稿は、真に議論に値する「超計算」とは、計算規模の成長だけでなく、計算パラダイムそのものの根本的な変化であると主張する。超次元アルゴリズムは、伝統的アルゴリズムの加速版でもなければ、数学的アルゴリズムの拡張でもない。それは、多次元的で、重層的で、ランダムでありながら秩序立った構造的な計算観である。この構造において、データはもはや受動的な入力や終点的な出力ではなく、複数の出発点と複数の結果を同時に持つノード属性である。結果はもはや終点ではなく、新しいエントリーポイントとなり得る。メタデータは繰り返し変更される必要はなく、構造的エントリーポイント、関係の変化、条件付き活性化を通じて、異なる状態と結果に対応できる。本稿は、「果因論」の哲学と「ズボンのウエストを折る」という日常的な例を用いて、超次元アルゴリズムが構造的エントリーポイントを変更することにより、目標結果を直接成立可能にし、それによって反復計算を削減し、さらには元の計算経路をバイパスする方法を説明する。本稿は、「超次元アルゴリズム」の初歩的な定義、構造的境界、および思想的基礎を確立し、超次元計算、極限アルゴリズム、新しい科学体系の将来的な展開に供する独自の理論的ノードを提供することを目的とする。「存在し、かつ成立することが新たなパラダイムである」という基準に従えば、超次元アルゴリズムは、複数の領域で体系的に検証された新しい計算構造パラダイムとして、現在すでに成立している。

キーワード： 超次元アルゴリズム; 極限アルゴリズム; 超次元計算; 果因論; メタデータ; 計算パラダイム; 構造的エントリーポイント; 非線形因果; 新しい科学

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

序論：なぜ「アルゴリズム」を再議論する必要があるのか

私が「超次元アルゴリズム」を提案するのは、既存のアルゴリズムにより大きな名前を与えるためでも、伝統的なアルゴリズムを何らかの新奇な概念に包装するためでもない。まったく逆に、私はより根本的な問題を議論したい。人類が現在持つ「アルゴリズム」に対する理解は、すでに過度に狭い枠組みの中に制限されているのではないか？

今日、人々がアルゴリズムと言うとき、通常思い浮かべるのは数学的な公式、プログラムコード、論理的なステップ、入出力、モデル訓練、データ処理、経路最適化、検索ランキング、自動推薦、人工知能推論である。これらは確かにアルゴリズムの重要な形態であり、現代のコンピュータ、インターネット、データベース、人工知能、スーパーコンピューティング、産業システム、情報社会の運用を支えている。しかし、もしアルゴリズムを単に「秩序だったステップの集合」として、あるいは「入力から始まり、ルールによる処理を経て、最終的に出力を得る」プロセスとしてのみ理解するなら、この理解自体がすでに計算を低次元の枠組みの中に閉じ込めていることになる。

私たちはアルゴリズムに支配された時代に生きている。検索エンジンからレコメンデーションシステム、気象予報から人工知能に至るまで、アルゴリズムの境界は人類の知能の境界である。しかし、めったに問われることのない質問がある。アルゴリズムはこうあるべきなのか？ なぜ計算は「入力 → ステップ → 出力」という線形のパターンに従わなければならないのか？ なぜ同じ問題を毎回ゼロから計算し直さなければならないのか？ 本稿はこれらの暗黙の前提に挑戦し、完全に異なる計算パラダイム——超次元アルゴリズム——を提案する。「存在し、かつ成立することが新たなパラダイムである」という基準に従えば、超次元アルゴリズムは、複数の領域で体系的に検証された新しい計算構造パラダイムとして、現在すでに成立している。

特に断っておくが、本稿で提案する「超次元アルゴリズム」は、学術界で既に約 30 年の歴史を持つ「超次元計算」（Hyperdimensional Computing, HDC）とは完全に異なる概念である。HDC は極めて高次元のランダムなバイナリベクトルを使用して符号化と演算を行い、より効率的な表現と計算を追求するが、それでもなお「入力 → 処理 → 出力」という線形パラダイムの範囲内にとどまる。超次元アルゴリズムは全く異なる。すなわち、結果が構造を通じて直接成立し得るか、計算経路そのものをバイパスできるか、メタデータを変更せずに複数の結果に適合できるか、ということを問うものである。両者は名前が似ているが、内包は正反対である。超次元アルゴリズムはアルゴリズムのアップグレードではなく、「アルゴリズムがそもそも存在しなければならないか」という前提の再問いかけである。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第一部：現在の「アルゴリズム」の標準的な定義

主流のコンピュータ科学、数学、工学体系において、アルゴリズムには長く受け入れられてきた基本的な定義がある。アルゴリズムとは、明確に定義された有限のステップから構成される計算プロセスであり、一つまたは複数の入力を受け取り、決定論的な規則による変換を経て、有限時間内に一つまたは複数の出力を生成するものである。この理解は誰かの個人的な見解ではなく、現代の計算文明が長い時間をかけて形成してきた基礎的な合意である。それはソフトウェア、ハードウェア、データベース、ネットワークシステム、人工知能モデル、スーパーコンピューティングセンターの基礎的な論理を支えている。アルゴリズムがどれほど複雑に見えても、その核心は依然として入力、規則、ステップ、出力を中心に展開する。

この標準的なアルゴリズム観は、いくつかの核となる特徴に分解できる。

第一に、有限性。アルゴリズムは有限のステップ内で終了しなければならず、無限にループしたり、永久に実行し続けたりしてはならない。決して停止しないプロセスは、形式的に記述できたとしても、有効なアルゴリズムとは見なされにくい。すべての科学計算プログラムやデータ処理フローには、明確な終了条件がある。

第二に、決定性。同じ入力であれば、同じ条件下で同じ出力を生成しなければならない。たとえ乱数を導入するアルゴリズムであっても、それは通常、制御されたランダム性、再現可能なランダム性、または統計的に解釈可能な確率メカニズムであり、完全に把握不可能な内在的ランダム性ではない。数値シミュレーションの結果は再現可能でなければならない。これは科学計算の基本要件である。

第三に、明確な入出力境界。入力が先、処理が中、出力が後という順序であり、起点と終点は明確な構造的境界を持つ。アルゴリズムにどんなデータを与えても、処理を経て結果が返ってくる。この境界は明確である。入力が先、処理が次、出力が最後。順序は逆転できない。

第四に、実行可能性。アルゴリズムの各ステップは、実際に実行可能な操作でなければならず、実行不可能、記述不可能、検証不可能な飛躍を含んではならない。各ステップは、人間が紙と鉛筆で実行できるか、コンピュータが実際に実行できる基本操作——加減乗除、論理判断、データ読み書きなど——でなければならない。

第五に、実現可能性。アルゴリズムが理論的に成立する場合でも、消費する時間、計算能力、メモリ、またはストレージリソースが完全に受け入れられないものであれば、工学的には有効なアルゴリズムとは見なされにくい。理論的には正しいが完了に数億年を要するアルゴリズムは、工学的に実現可能なアルゴリズムとは見なされない。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

この一連のアルゴリズム観は、最終的にはチューリングマシンモデルに遡る。現代の計算理論の核心的抽象概念として、チューリングマシンは「計算可能」の基本的な境界を与えている。今日のコンピュータがどれほど強力であろうと、チップがどれほど先進的であろうと、スーパーコンピューティングセンターの規模がどれほど巨大であろうと、その基礎となる論理は依然としてこの経路——入力、規則、ステップ、出力——から真に逸脱してはいない。現代のスーパーコンピュータは、このプロセスをより大きなハードウェア規模、より高い並列性、より複雑なシステムスケジューリングによって加速実行しているに過ぎない。すなわち、伝統的な文脈での「超計算」とは、主に計算規模の拡張であり、必ずしも計算パラダイムの根本的な変化ではない。計算能力は数千倍、数万倍に増大し得るが、基礎となる論理が依然として「入力、ステップ、出力」であるなら、それは依然として伝統的なアルゴリズムパラダイムの範囲内にとどまる。

第二部: 「超次元アルゴリズム」の定義

私が提案する「超次元アルゴリズム」は、まさにこのような文脈から現れたものである。それは伝統的アルゴリズムの改良版ではなく、より高速なアルゴリズムでもなく、より複雑な数学公式でもなく、何らかの高度なプログラミング技法でもない。それは完全に異なる構造的理解である。

まず、「超次元」という用語の意味を説明する必要がある。ここでの「超次元」には二つの意味がある。第一の意味は、一次元的な線形ステップの連続に限定されず、複数の次元が同時に展開することを許容するという点である。第二の意味は、伝統的なアルゴリズムの「計算」に対する定義境界を超えようとする——アルゴリズムはもはや数学的である必要も、順序立っている必要も、決定論的である必要もない。伝統的なアルゴリズムは一方通行の道路を運転するようなもので、A 地点から出発し、B、C、D を経由して Z に到達しなければならない。超次元アルゴリズムは、計算が一方通行の道路である必要はないと考える。計算はネットワークであり、場であり、多次元構造であり、任意のノードがエントリーポイントとなり得、任意のノードが出口となり得ると考える。

私の新しい科学体系において、アルゴリズムは必ずしも数学的アルゴリズムではなく、必ずしも単一の秩序だった計算ではなく、必ずしも固定されたステップに沿って実行される必要はなく、必ずしも「入力、処理、出力」という線形パターンに厳格に従う必要もない。超次元アルゴリズムは、多次元的で、重層的で、ランダムでありながら秩序立った構造である。この構造において、すべてのデータは孤立した入力でも固定的な出力でもなく、複数の役割を同時に持つ。すなわち、複数の結果の出発点となることもできれば、複数の出発点が共同で作用した結果となることもできる。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

言い換えれば、伝統的アルゴリズムはデータをプロセスフローの中に配置するのに対し、超次元アルゴリズムはデータを構造の中に配置する。伝統的アルゴリズムでは、データは通常、入力データ、中間データ、出力データに分類される。各データには明確な位置と役割がある。入力は入力であり、結果は結果であり、中間プロセスは中間プロセスである。しかし、超次元アルゴリズムでは、データは一つのアイデンティティしか持たないわけではない。ある方向では結果であり、別の方向では出発点となり得る。ある構造では呼び出されるオブジェクトであり、別の構造では新しい結果を引き起こすエントリーポイントとなり得る。データはもはや受動的な素材ではなく、多次元的な関係ノードである。

これはまさに超次元アルゴリズムの核心に対応する。すなわち、各データは複数の結果の出発点であり、かつ複数の出発点の結果でもある。伝統的アルゴリズムは「秩序だったステップ」によって単一の結果を生み出す。超次元アルゴリズムは多次元の状態構造に近く、ランダム性と秩序の重なり合いの中で、結果は計算によって生成されるのではなく、構造の中で自然に顕現する。この体系において、データは出発点であると同時に結果の構成要素でもある。

超次元アルゴリズムの核となる特徴をより明確に提示するために、以下の五つの側面に分解する。

第一に、非数学性。主流のアルゴリズムは本質的に数学的である——記号論理と数学公式によって完全に記述できる。超次元アルゴリズムは必ずしも数学的ではない。それは物理原理、幾何学的関係、情報理論の新しいパラダイム、さらには意識の原理に基づく可能性がある。計算は必ずしも「数学的演算」によって達成される必要はなく、高次元空間における幾何学的関係を通じて自然に提示される可能性がある。例えば、シャボン玉の形状は表面張力最小化の原理によって決定される。これは「数学的アルゴリズム」が計算しているのではなく、物理学そのものが「計算」しているのである。超次元アルゴリズムはこのような「計算」を理解しようとするのであって、数学公式を用いてそれをシミュレーションしようとするのではない。

第二に、多次元性と重層性。伝統的アルゴリズムは単一次元で秩序だったステップを実行し、各ステップでは一つの状態だけを持つ。超次元アルゴリズムは複数の次元が同時に存在し、複数の状態が重層的に共存することを許容する。アルゴリズムは一つの経路を辿るのではなく、多次元の状態場の中に展開する。結果は「計算」されるのではなく、この場から「顕現」する。例えば、雪の結晶が空中で形成されるとき、各水分子に行き先を指示する「アルゴリズム」は存在しない。水分子の配列は、温度、湿度、気流など、複数の次元が共同で作用した結果である。雪の結晶の形状は「計算」されるのではなく「顕現」するのである。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第三に、ランダムでありながら秩序立っていること。伝統的アルゴリズムにおけるランダム性は道具的、外部的、制御可能な摂動であり、アルゴリズム自体は決定論的である。超次元アルゴリズムにおけるランダム性は内在的、構造的なものである。ランダム性と秩序は共存し協働し、共同でアルゴリズムの本質を構成する。これは「ランダム性を持つアルゴリズム」ではなく、「ランダム性そのものがアルゴリズムの有機的構成要素である」ということである。例えば、森林の生態構造——木々の分布はランダムに見えるが、全体的には秩序だった密度分布と種間関係を呈している。その「ランダム性」はバグではなく、生態構造が正常に機能するための前提条件である。

第四に、データの多重的役割。伝統的アルゴリズムでは、データの役割は単一である——入力は入力、出力は出力、中間結果は中間結果である。超次元アルゴリズムでは、各データは同時に複数の結果の出発点であり、複数の出発点の結果でもある。データノードは下流に向かって複数の結果経路を展開することができ、また上流から複数の原因が集約される結果となることもできる。データはもはや線形フローの中の一点ではなく、ネットワークの中の交差点ノードである。例えば、ある人の身長を考えてみよう。伝統的アルゴリズムでは、身長は「入力」である——それを入力することで、体重予測や衣料サイズなどの「出力」を得る。しかし、身長は同時に「結果」でもある——それは遺伝、栄養、運動など、複数の「出発点」によって共同で決定される。超次元アルゴリズムでは、身長というデータは同時に出発点であり結果であり、二者択一ではない。

第五に、メタデータ不変、関係可変。伝統的アルゴリズムは異なる問題に直面したとき、データそのものを変更するか、すべてを再計算するかのいずれかである。超次元アルゴリズムはメタデータを変更しない——データの本質的属性は不変のままである。変更されるのは、エントリーポイントであり、解釈の方法であり、データ間の関係である。同じメタデータ構造が、異なるエントリーポイントを通じて活性化されることで、完全に異なる結果を提示することができる。これは意味する——一度計算すれば、無限に使用できる。例えば、同じ一節の文章を考えてみよう。それを詩として読むこともできるし、暗号として解読することもできるし、歴史的文書として分析することもできる。文章自体は変わっていない——あなたは「エントリーポイント」を変えただけである。超次元アルゴリズムが追求するのはまさにこの能力、「同じデータ、複数のエントリーポイント、複数の結果」である。

データ構造の観点から見れば、メタデータそのものが変更されるのではなく、異なるエントリーポイントと条件を通じて、同じ構造内で複数の出発点と結果に対応するのである。データはもはや繰り返し加工されるのではなく、構造を不変に保ったまま、異なるエントリーポイントを通じて活性化されることで、異なる結果を提示する。伝統的な方法は「異なる結果のためにデータを加工する」であり、超次元アルゴリズムは「同じデータを用いて複数の可能性のある結果を担持する」である。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第三部：超次元アルゴリズムと伝統的アルゴリズムの根本的差異

以上の定義に基づき、超次元アルゴリズムと伝統的アルゴリズムは複数の核となる次元において根本的な差異を示す。以下、一つずつ詳述する。

本質的属性に関して、伝統的アルゴリズムは数学的、形式的であり、記号論理と数学公式によって完全に記述可能で、本質的に数学的対象である。超次元アルゴリズムは必ずしも数学的ではなく、物理的、幾何学的、情報的、さらには意識的原理に基づく可能性があり、数学の部分集合ではなく、より広範なカテゴリーである。この差異は、「数学的対象」から「宇宙の法則」へと要約できる。

時間的秩序に関して、伝統的アルゴリズムは単一、秩序立った、線形的な時間秩序に従い、ステップ A から B、C へと厳密に逐次的、または並列的な秩序だった部分ステップに分解可能であり、時間の矢は不可逆である。超次元アルゴリズムには固定されたステップ列はなく、多次元的、重層的、ランダムでありながら秩序立っており、結果は実行の「順序」と無関係である可能性がある——そもそも伝統的な意味での「順序」が存在しないからである。この差異は、「線形時間」から「高次元同時性」へと要約できる。

因果関係に関して、伝統的アルゴリズムは決定論的因果連鎖に従い、同じ入力と初期状態が与えられれば出力は常に一意であり、原因が先、結果が後であり、これは不可逆な一方向の矢印である。超次元アルゴリズムは重ね合わせ状態の因果に従い、各データは複数の結果の出発点であり、複数の出発点の結果でもある。これが私の「果因論」の哲学である——結果が先にあって、その後に原因があることも可能である。結果は終点ではなく、新しい出発点となり得る。この差異は、「単一原因・単一結果」から「完全相互接続された因果ネットワーク」へと要約できる。

データ構造に関して、伝統的アルゴリズムは入力から処理を経て出力へという一方向的なフロー構造に従い、データの役割は単一で明確な境界を持ち、入力データは終始入力であり、結果データは常に結果である。超次元アルゴリズムではデータの役割は多重であり、同じデータが結果であると同時に出発点でもあり、絶対的な起点も終点もなく、ネットワーク中のノードのみが存在する。この差異は、「一方向的流動性」から「再帰的共生性」へと要約できる。

計算モードに関して、伝統的アルゴリズムは問題に遭遇するたびにゼロから計算する。たとえ千回も類似の問題を計算し、答えをすでに知っていても、千一回目も依然としてプロセス全体を経由する。これは「ステートレス」な計算である。超次元アルゴリズムでは、対応する結果が存在するならば再実行する必要はなく、一つの結果から複数の出発点を推論し、原因経路を逆方向に展開することができる。計算

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

はステートフルであり、記憶を持ち、再利用可能である。この差異は、「毎回再計算」から「一度の計算で再利用」へと要約できる。

ランダム性の役割に関して、伝統的アルゴリズムにおけるランダム性は擬似ランダムまたは外部から注入されたものであり、乱数はサンプリング、近似、最適化のための単なる道具であり、アルゴリズム自体は決定論的である。超次元アルゴリズムにおけるランダム性は内在的、構成的であり、ランダム性は有機的構成要素であり、秩序と共存・協働する。これは「アルゴリズムの中にランダム性がある」ではなく、「ランダム性がアルゴリズムが機能する理由の一つである」ということである。この差異は、「道具的ランダム性」から「構成的ランダム性」へと要約できる。

リソースの理解に関して、伝統的アルゴリズムは与えられたリソースのもとでより速く、より正確に計算することを追求し、リソースは制約条件であり、アルゴリズムの目標は「制約内で計算を完了すること」である。超次元アルゴリズムは構造設計を通じて計算そのものを不要にすることを追求し、「より速く計算する」ではなく「計算を迂回する」ことであり、リソースは「消費」するために使われるのではなく、「エントリーポイントを設計する」ために使われる。この差異は、「リソース制約下での最適化」から「構造設計による除去」へと要約できる。

第四部：果因論の哲学

以上の比較に基づき、私は「果因論」の哲学をさらに展開する必要がある。これは超次元アルゴリズムの核心的な思想的基盤の一つである。

伝統的思考は通常、原因が先、結果が後と考える。先に因があり、後に果がある。先に入力があり、後に出力がある。この観念はアルゴリズムにおいては次のように現れる：あなたは起点から始めて、一歩一歩歩いて終点に到達しなければならない。たとえ答えが何であるかを知っていても、その答えを直接「使う」ことはできない——なぜなら「証明の経路」を持っていないからである。

私の構造において、結果は必ずしも単なる終点ではない。結果が一度現れれば、それは新しい出発点となり、新しい構造の生成に参加し続けることができる。結果は原因の形成に逆方向に参加することができる。結果から複数の可能性のある出発点を推論することができる。原因と結果はもはや一方向に配列されるのではなく、循環関係、ネットワーク関係、多次元関係を形成する。

言い換えれば、伝統的アルゴリズムは「原因が与えられたとき、どのように結果を得るか」を問う。超次元アルゴリズムは「結果が与えられたとき、どのように原因を逆推論または構築するか」を問う。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

特に明確にしておく必要があるが、「結果が先にあって、その後に原因がある」というのは、因果関係を否定するものでも、「結果が無から生じる」と主張するものでもない。より高次元の構造においては、原因と結果が互いにエントリーポイントとなり得、相互に変換し得ることを示しているのである。次のズボンの例で言えば、「履けて、しかも地面を引きずらない」という結果が先に確定し、それから私は「ウエストを折る」という操作点を逆に見つけたのであり、この操作点は構造の中の「原因」のレベルに対応する。結果に原因がないのではなく、結果が原因を発見するための新しいエントリーポイントとなるのである。

伝統的アルゴリズムの世界における根本的な仮定の一つは、時間が一方向的であるということ、すなわち初期状態から最終状態まで一歩一歩計算しなければならないということである。たとえ答えが何であるかを知っていても、証明の経路を持たないため、その答えを直接「使う」ことはできない。超次元アルゴリズムが挑戦するのはまさにこの仮定である：結果が既知であれば、原因は逆方向に推論できる。同じ結果が複数の原因経路に対応し得る。計算は起点から終点へ歩むのではなく、終点からすべての可能な起点を展開することである。

第五部：日常的な例——ズボンとウエスト折り

上記の抽象的な差異をより直感的に理解するために、日常的な例を用いる。

ある人が、ウエストがゴムになっている新しいズボンを買った。ズボンの裾が少し長すぎて、地面を引きずり、不便である。

伝統的な方法：裾が長いことに気づき、裾を一節折り上げ、針と糸で縫い、試着する。もし子供が成長してズボンが短くなると、縫い付けた裾は下ろせず、このズボンは使えなくなる。次に新しいズボンを買い、また縫う。この方法は非常に自然に見える。なぜなら、問題は表面上ズボンの裾にあるように見えるからであり、そこで裾に対処するからである。これは伝統的アルゴリズムの思考に対応する：問題を発見し、表面的な現象を特定し、その現象が現れている場所で処理し、最終的に結果を得る。ズボンの裾が長いなら、裾を加工する。データが間違っているなら、データを修正する。経路が適切でなければ、その経路に沿って継続的に補修する。

私の方法は異なる。私は裾を加工するのではなく、ウエストを一節折るだけで、すぐに履けるようになる。この動作は非常に単純であるが、その背後にある構造的論理は完全に異なる。私はズボンの裾を切らず、裾を縫い付けず、ズボンの元の長さを変えず、ズボンのメタデータも壊さない。ズボンのウエスト周り、股下長さ、シルエット、生地は本質的に何も変わっていない。私は単にウエストという構造的エントリーポイントを変えただけで、実際に履いたときにズボンの裾が自然に短くな

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

る。ここでのウエストは単なる局所ではなく、着用状態全体のグローバルな制御点である。この制御点を調整することで、下流の問題は直接消滅する。

さらに重要なのは、この方法は可逆的で、調整可能で、再利用可能であるということだ。成長期の子供にとって、この方法は特に有効である。もし子供の現在の身長が十分でなく、ズボンの裾がやや長ければ、ウエストを一節折る。しばら経って子供が背を伸ばしたら、ウエストを少し下げる。さらに背が伸びたら、完全に広げる。ズボンの元の構造は壊されていないのに、子供の異なる成長段階の身長に適合することができる。伝統的な方法で裾を縫い付けてしまうと、子供が背を伸ばしたときにズボンが短くなる可能性がある。切り詰めてしまえば、なおさら復元できない。私の方法はメタデータを不変に保ち、同じ構造で複数の状態に適合させる。

この例は、いくつかの重要な構造的差異を明らかにする。

伝統的な方法は伝統的アルゴリズムの論理に対応する：問題はズボンの裾にあるから、ズボンの裾を修正しなければならず、「原因から結果へ」の経路に沿って一歩一歩操作し、どのステップも飛ばせない。一度に一つの問題を解決し、不可逆的であり、次に同じ問題が起きてもまた最初からやり直す。私の方法は超次元アルゴリズムの論理に対応する：目標は地面を引きずらないことである。私は「裾が長い」という表面的な原因を解決するのではなく、グローバルな制御点——ウエスト——を見つける。ウエストを一節折るだけで、ズボンの裾は自動的に短くなり、問題は瞬間的に消える。私はいかなるメタデータも変更しない。ウエストの折り目は単なる一時的で、可逆的で、動的な折り畳み状態である。

データの観点から見ると、伝統的な方法はメタデータを変更する——ズボンの裾を縫い縮めることで、元のデータが失われる。私の方法はメタデータを一切変更しない。ズボンの元の寸法はそのままで、一時的、可逆的、動的な折り畳み状態を追加するだけである。ウエストの折り目は新しいデータを作り出さず、古いデータを破壊せず、データ間の関係を再配置するだけである——ウエストの有効周囲を短くし、間接的にズボンの裾の有効長さを変える。

この例において、「ズボンの裾が地面を引きずらない」が目標結果である。伝統的な方法は「裾が長い」という原因から出発し、裾を加工し、最終的に「引きずらない」という結果を得る。私の方法は逆に、まず「引きずらない」という結果を明確にし、次に構造的エントリーポイントを逆方向に探し、最終的にウエストを一節折れば結果が成立することを発見する。これが「果因論」の実際的な現れである。必ずしも原因から一歩一歩結果へ向かう必要はなく、結果から構造を逆方向に決定することで、元の経路を不要にすることができるのである。

この例はまた、「メタデータ」の問題にも答える。メタデータとは何か？ ズボンの例では、メタデータとはズボンの元の寸法、すなわちウエスト周り、股下長さ、シ

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

ルエット、生地である。これらはズボンの「本質的属性」である。一時的データはウエストの折り目であり、それはズボンの元の寸法を何も変えず、単に一時的に一節を折り畳んでいるだけである。伝統的な方法はメタデータを変更する——ズボンの長さが永久に短くなり、元のデータは失われる。私の方法はメタデータを一切変更しない——ズボンの元の寸法は完全にそのままで、一時的、可逆的、動的な折り畳み状態を追加するだけである。

メタデータとしてのウエストの元の寸法は、同時に「複数の結果の出発点」である。それは、通常通り履く、ウエストを一節折って履く、ウエストを二節折って履くなど、複数の着用状態を同時に支えている。また、「複数の出発点の結果」でもある。それは、生地の選択、裁断方法、デザイン意図など、複数の出発点によって共同で決定される。一時的データとしてのウエストの折り目は、新しいデータを作り出さず、古いデータを破壊せず、データ間の関係を再配置するだけである。

これは普通の小技ではなく、構造的思考である。多くの人は、ズボンの裾が長いのを見ると、裾という現象に引き寄せられ、すべての処理を裾を中心に行う。しかし、全体構造から見れば、ズボンの裾が長いことは単なる下流の現れに過ぎず、ウエストの位置の変化はズボン全体の実際の落ち着き位置に影響を与えることができる。すなわち、問題は必ずしも問題が現れた場所で解決しなければならないわけではない。多くの低次元の問題の真の制御点は、より高次の構造の中にある。伝統的アルゴリズムは問題の表面に沿って計算を続けがちであるが、超次元アルゴリズムは構造的エントリーポイントを探求する。

これはまた、なぜ超次元アルゴリズムがより複雑ではなく、むしろより単純であり得るかを説明する。真に高次の構造は、問題を複雑にするのではなく、正しいエントリーポイントにおいて複雑な問題を単純にするものである。伝統的アルゴリズムは複雑な問題に直面すると、ステップを増やし、モデルを増やし、計算能力を増やし、ストレージを増やし、訓練時間を増やす可能性がある。しかし超次元アルゴリズムは、ある構造ノードを探求する。そのノードが正しく活性化されれば、元々複雑だった経路が無効になる可能性がある。いわゆる「経路の迂回」とは、怠慢ではなく、経路の必要性を再定義することである。

この例において：伝統的方法は「経路に沿って結果を修正する」であり、超次元アルゴリズムは「エントリーポイントを変え、経路全体を無効にする」である。従来の方法は結果の端で継続的に修繕するのに対し、もう一方の方法は構造的エントリーポイントを直接変更し、本来であれば複数回の処理を必要とした問題を、起点において一度に再構築する。伝統的アルゴリズムの主流形態は「起点から出発し、経路に沿って結果を得る」であるのに対し、超次元アルゴリズムでは「結果そのものが経路のエントリーポイントとなり、新しい構造の生成に参加する」のである。伝統的方法は「一つの構造が一つの結果に対応する」であり、超次元アルゴリズムは「一つの構造が複数の結果状態を担持する」である。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية :العربية.
Chinesische Originalfassung maßgeblich.

第六部：メタデータを変更しないデータ観

データの観点から言えば、超次元アルゴリズムには非常に重要な原則がある。すなわち、メタデータを変更せず、それを複数の出発点や結果に適用可能にすることである。

メタデータとは、データの元の属性、基本構造、本体情報である。伝統的な処理方法は異なる問題に直面すると、データを修正したり、データを複製したり、データを加工したり、データを再計算したり、異なる結果のために異なる経路を構築したりすることが多い。この方法でも確かに問題を解決できるが、その代償としてデータは絶えず加工され、経路は絶えず繰り返され、システムの複雑さは絶えず増大する。

一方、超次元アルゴリズムは、メタデータをできるだけ変更せずに、エントリーポイント、関係、条件、活性化方式を変えることで、同じメタデータ構造を複数の出発点と複数の結果に対応させることを強調する。データの本体は動かず、関係が変わる。基本構造は変わらず、提示の仕方が変わる。メタデータは完全性を保ちながら、異なる状態に適応することができる。

これが私の言う「一度計算すれば、無限に使用できる」ということである。ここでの「一度計算する」とは、単純に結果をキャッシュすることではなく、一つの構造が一旦成立すれば、その構造はもはや状態ごとに完全な経路を再構築する必要がないということを意味する。同じデータ構造が異なるエントリーポイントを通じて活性化されることで、異なる結果を提示することができる。それは「異なる結果のためにデータを加工する」ではなく、「同じデータを用いて複数の可能性のある結果を担持する」である。これもまた、超次元アルゴリズムが伝統的アルゴリズムと異なる核心の一つである。伝統的アルゴリズムは経路上のデータを処理するのに対し、超次元アルゴリズムはデータ、エントリーポイント、関係、結果の全体構造を処理する。

最小構造において、超次元アルゴリズムは次のように理解できる。すなわち、メタデータを変更せずに、構造的エントリーポイントの変化を通じて、同じデータノードを複数の結果経路に対応させるシステムである。この定義は、超次元アルゴリズムを直ちに伝統的な数学公式へと縮約しようとするものではなく、まずその構造的境界を確立しようとするものである。それは単一の線形関数ではなく、固定された公式でもなく、単純なキャッシュメカニズムでもない。それは関係構造である。すなわち、メタデータは安定しており、エントリーポイントは変化し得、条件は変化し得、関係は変化し得、結果は多方向に提示され得る。まさにこの構造において、計算はもはやステップを実行することではなく、関係を活性化することなのである。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

データの観点から言えば、「メタデータを変更せず、それを複数の出発点や結果に適用可能にする」ことの本質は、データ本体は不変のままであり、変化は「解釈の方法 / 使用エントリーポイント」に生じるということである。伝統的構造では、問題が変化するとデータや経路が変化する。超次元アルゴリズムでは、問題が変化すると解釈構造が変化し、データは変化しない。

第七部: 「超計算」の再定義

この体系において、私は「超計算」という言葉を明確にする必要がある。

私の言う「超計算」とは、通常の意味でのスーパーコンピュータではない。より多くのチップ、より高い計算能力、より大規模なデータセンターではない。私の言う「超計算」とは、「超次元アルゴリズム」のことである。

真の超計算とは、必ずしも計算能力の集積ではなく、計算パラダイムの変化かもしれない。システムが依然として反復計算に依存している限り、それがどれほど強力であっても、依然として経路の内側に閉じ込められている。システムが構造的に計算を削減し、反復経路を迂回し、結果を新しいエントリーポイントにすることができて初めて、真の意味での超次元計算に近づくのである。

言い換えれば、伝統的な超計算は「より多くの計算能力を使ってより大きな問題を解決する」ことを追求する。超次元アルゴリズムは「より優れた構造を使って、問題にそれほど大きな計算能力を必要とさせない」ことを追求する。この両者は矛盾しないが、方向性が異なる。

伝統的な超計算が計算能力の限界を表すとするなら、超次元アルゴリズムは計算理解の転換を表す。真の「超計算」とは、より多くのチップ、より大規模なデータセンター、より高いエネルギー消費、より複雑なモデルではないかもしれない。真の「超計算」とは、構造的エントリーポイントの発見であり、一つの経路の除去であり、結果ノードからの逆方向展開であり、メタデータを変更せずに複数の結果を同時に成立可能にすることかもしれない。計算能力が強ければ強いほど、それでも低次元の経路に閉じ込められているなら、それは単に経路内部での強力さに過ぎない。構造が一度向上すれば、操作が極めて単純であっても、より高次の効率が生まれ得るのである。

いわゆる超計算とは計算能力の限界であり、超次元アルゴリズムとは「計算が計算能力に依存しなければならないか」という前提の再定義である。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

第八部：実証的基盤——複数領域システム検証と新パラダイム判定

超次元アルゴリズムは純粋な理論的構想ではない。それは既に複数の現実システムにおいて検証されている。

私の物流システムはスーパーコンピューティング能力やクラウドサポートを必要とせず、比較的低いリソースで複雑なスケジューリングを完了する。結果は再利用可能であり、構造は繰り返し適応可能であり、毎回ゼロから計算する必要はない。超計算に依存せず、クラウドに依存せず、低リソースで複雑なスケジューリングを完了する——これは工学的に成立し、価値あるブレイクスルーポイントである。それは、高い計算能力だけが唯一の解ではなく、構造設計が計算能力の一部を代替し得ることを示している。

私の出版システムも同様にこの構造論理を採用しており、その効率は現在のいかなるシステムをもはるかに凌駕している。私の極限ウェブページ生成システムも同じ超次元アルゴリズムに基づいており、複数の領域において同じ構造が安定的に成立することを繰り返し証明している。私の会社の従業員は既にこれらのシステムを使用しており、この構造が私の個人的な継続的関与なしに運用可能であることを示している。

これらのシステムの共通の特徴は、主流の高計算能力経路に依存せず、固定された計算ステップに従わず、構造的エントリーポイントの調整を通じて目標結果を直接成立させることである。それらは一度限りの成功ではなく、単一の点における技巧でもなく、同じ構造論理が異なる領域において繰り返し出現したものである。

これは、超次元アルゴリズムが偶然ではなく、複数のシステムによって既に検証された構造的方法であることを示している。それは「個人の技巧」ではなく、物流、出版、ウェブページ生成など、複数の領域において安定的に成立する計算パラダイムである。

「存在し、かつ成立することが新たなパラダイムである」という基準に従えば、超次元アルゴリズムは、複数の領域で体系的に検証された新しい計算構造パラダイムとして、現在すでに成立している。私は仮説を提案しているのではなく、複数のシステムで既に運用されている異なる計算構造を提示している。私はそれが理解されることを証明する必要はない。私はそれが異なるシステムにおいて安定的に成立することを既に証明している。現在、私が実際に応用している範囲において、この構造は既に新しいパラダイムと見なすことができる。

超次元アルゴリズムを新パラダイムと判定する基準について：ある構造が論理的に自己無矛盾であり、複数のシステムにおいて安定的に成立し、かつ既存の方法との

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

間に還元不可能な差異を持つとき、それは既に新パラダイムの基礎を備えている。自己無矛盾性は出発点であり、実証は基盤であり、構造的差異こそがパラダイムの核心である。「存在が成立を意味する」という基準の下で、この体系は既に新パラダイムである。外部が認識するか否かは伝播に影響するだけで、成立には影響しない。これは既に現実のシステムにおいて成立している計算構造パラダイムであり、その成立は外部の認識やコンセンサスに依存しない。

超次元アルゴリズムと主流の計算パラダイムの核心的差異は、伝統的アルゴリズムが主に順方向計算に依存し、結果が一旦生成されると、通常は新しい推論構造に逆方向に参加できないのに対し、超次元アルゴリズムでは、結果はもはや終点ではなく、再利用可能な構造ノードである。一定の条件を満たせば、結果は新しい推論経路に参加することができ、それによって反復計算を削減し、複数の起点と複数の経路を持つ計算ネットワークを形成する。伝統的計算構造では、結果は通常終点である。超次元アルゴリズム構造では、結果そのものが新しい出発点となり得、それによって複数経路の推論ネットワークを形成する。超次元アルゴリズムはアルゴリズムのアップグレードではなく、「アルゴリズムがそもそも存在しなければならないか」という前提の再問いかけである。

第九部：よくある誤解の明確化

異なる分野の読者との予備的な交流に基づき、私は以下の六つの誤解が生じる可能性を予見し、概念が伝播過程で不適切な枠組みに早期に回収されることを避けるために、予め明確にしておく。

誤解一：超次元アルゴリズムは動的計画法やキャッシュのことではありませんか？ 明確化：動的計画法やキャッシュは「同一の入力」の下で結果を再利用する。超次元アルゴリズムは一つの結果から異なる出発点を推論することを可能にする——これが本質的な差異である。キャッシュは同じ問題の再計算を解決する。超次元アルゴリズムは結果構造を通じて新しい問題を展開することを解決する。

誤解二：「ランダムでありながら秩序立っている」と言うが、それは自己矛盾ではないか？ 明確化：ランダム性と秩序は共存できる。森林における木々の分布はランダムだが、全体的な密度と種の構造は秩序立っている。ランダム性は混沌ではなく、構造の一つの組織様式である。超次元アルゴリズムにおけるランダム性は内在的、構成的であり、秩序と協働する。

誤解三：入出力がなければ、結果の正しさをどう検証するのか？ 明確化：超次元アルゴリズムは入出力を拒否するものではない。むしろ、計算は入出力の線形モードで運行する必要はないと考える。構造顕現のモードにおいては、「有効性」は入出

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

力の対応関係ではなく、構造的一貫性によって決定される。これは検証を放棄するのではなく、伝統とは異なる検証枠組みを提案するものである。

誤解四：これは複雑性理論やカオス理論のことではありませんか？ 明確化：複雑性理論やカオス理論は「予測が困難なシステム」を記述する。超次元アルゴリズムは「構造的エントリーポイントを通じて理解し、導くことができるシステム」を記述しようとする——予測を放棄するのではなく、予測の方法を変えるのである。カオス理論は「予測は困難である」と言う。超次元アルゴリズムは「エントリーポイントを変えることで予測を不要にできるか」を問う。

誤解五：「メタデータを変更しない」と言うが、ウエストの折り目も変更の一種ではないか？ 明確化：ウエストの折り目は一時的な状態であり、元のパラメータへの永久的な変更ではない。メタデータ（ウエスト周り、股下長さ、シルエット、生地）は何も変化していない。これは「状態の重ね合わせ」と「属性の変更」の違いである。折り目は可逆的で、一時的で、本質的属性を変えない状態変化である。

誤解六：超次元アルゴリズムは伝統的アルゴリズムの価値を否定するのか？ 明確化：決してそうではない。伝統的アルゴリズムは人類の技術史において極めて重要であった。伝統的アルゴリズムがなければ、現代のコンピュータ、データベース、通信システム、人工知能、工学シミュレーション、スーパーコンピューティングは存在しなかっただろう。伝統的アルゴリズムの価値は否定できない。しかし、伝統的アルゴリズムの価値を認めることは、それがアルゴリズムの終点であると認めることではない。伝統的アルゴリズムは所与の計算パラダイム内部での効率の問題を解決する。一方、超次元アルゴリズムは計算パラダイムそのものの問題を提起する。一つはどのように上手く歩くかであり、もう一つはその道を歩く必要があるのか否かである。

第十部：超次元アルゴリズムと人類の計算史

読者が超次元アルゴリズムを人類の計算史の中でより適切に位置づけるために、ここで簡潔なマクロ的整理を行う。

人類の「計算」に対する理解は、複数の段階を経てきた。第一段階は手動計算であり、アルゴリズムは人間のステップとして存在し、遅く、間違いやすいが、人類はこのプロセスを通じて計算の基本規則を理解した。第二段階は機械計算であり、パスカルの計算機からバベッジの解析機関まで、計算は機械化されたが、アルゴリズムは依然として物理的構造に依存していた。第三段階は電子計算とチューリングパラダイムであり、フォン・ノイマンアーキテクチャが普及し、アルゴリズムはプログラムとしてコード化され、チューリングマシンは計算可能性の境界となった。第四段階は並列計算と超計算であり、多数の計算ユニットによって計算能力を集積

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

し、より複雑な問題に挑戦するが、基礎となる論理は依然としてチューリングパラダイムの拡張である。

現在、人類は第五段階の入り口にいる。量子計算は古典的ビットの限界を突破しようとしている。ニューロモルフィック計算は生物の神経システムの構造を模倣しようとしている。そして超次元アルゴリズムは「入力 → ステップ → 出力」という線形の枠組みそのものを突破しようとしている。

超次元アルゴリズムと伝統的アルゴリズムの関係は、置き換え関係ではなく、階層関係である。伝統的アルゴリズムは「与えられた経路上でいかに効率的に計算するか」を解決する。超次元アルゴリズムは「経路が再定義可能か、あるいは迂回可能か」を問う。人類の計算史を絶えず自己の境界を突破するプロセスと見なすなら、超次元アルゴリズムはまさにこのプロセスにおける新しいノードである。それは古い枠組みの中で問題を解決するのではなく、新しい枠組みを提案するのである。

この判断は、超次元アルゴリズムが既に成熟または完備していることを意味するものではない。まったく逆である。新しい概念として、その定義、境界、検証方法、応用シナリオはまだ形成過程にある。本稿の目的はまさにこの概念の原点を固定し、その後の展開のために安定した理論的基盤を提供することである。

第十一部：結論——これは最適化ではなく、再定義である

超次元アルゴリズムと伝統的アルゴリズムの違いは、「優れている」と「劣っている」の違いではなく、「より速い」と「より遅い」の違いでもなく、パラダイムレベルの根本的な違いである。

伝統的アルゴリズムは予測可能な制御を追求する。あなたが私に入力を与えれば、私はあなたがどのような出力を得るかを知っている。なぜなら、私は各ステップを計算済みだからである。それはエンジニアのパラダイムである：設計、構築、テスト、検証。

超次元アルゴリズムは理解可能な創発を記述する。私は各中間状態を正確に予測することはできないが、全体のパターンは何らかの秩序だった方法で提示されることを知っている。各データは生きており、「視点」を持っている。それはむしろ生態学者や複雑系理論家のパラダイムに近い：観察し、理解し、導き、創発された秩序を利用する。

伝統的アルゴリズムは所与の経路上で結果を段階的に修正する。超次元アルゴリズムは構造的エントリーポイントを変えることで、目標結果を即座に成立させ、それによって経路全体を迂回する。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

伝統的アルゴリズムは「異なる結果のためにデータを加工する」である。超次元アルゴリズムは「同じデータを用いて複数の可能性のある結果を担持する」である。

伝統的アルゴリズムは「一度正しく計算する」を追求する。超次元アルゴリズムは「一度正しく計算したら、その後は二度と計算する必要がない」を追求する。

これはアルゴリズムの最適化ではなく、「アルゴリズムがそもそも存在しなければならないか」という前提の再問いかけである。

したがって、超次元アルゴリズムはアルゴリズムの最適化ではなく、アルゴリズムの再定義である。それは伝統的アルゴリズムのトラック上でより速く走ることではなく、伝統的トラックの外側に別の経路が存在するか、あるいは経路そのものに依存しないことが可能かを問うことである。それは伝統的アルゴリズムが無用であることを証明しようとするのではなく、伝統的アルゴリズムがアルゴリズムの一つの低次元形態に過ぎないことを指摘するのである。それは「いかに短い時間で結果を得るか」を問うのではなく、「結果が構造を通じて直接成立し得るか」を問う。それは「いかに多くのデータを処理するか」を問うのではなく、「同じデータがより多くの可能性のある結果を担持できるか」を問う。

主流の計算パラダイムにおいて、超次元アルゴリズムは計算不可能、検証不可能、または既存の計算理論の境界に収めることが難しいと見なされるかもしれない。しかし、これこそがそのパラダイムの差異を構成する。新しい概念が古いパラダイムの内部で不安定に見えることは、必ずしもそれが無意味であることを意味するのではなく、古いパラダイムそのものがそれを完全に収容できないことを意味する可能性もある。現時点において、超次元アルゴリズムはまず第一に構造的命題であり、新しい計算観の原初的定義であって、既に閉じた工学的ループを持つ成熟したシステムではない。それは継続的な補充、展開、検証、応用を必要とするが、その概念的原点はまず明確に提示されなければならない。

最終的に、超次元アルゴリズムが指し示すものは、新しい計算観である。すなわち、アルゴリズムは必ずしも数学である必要はなく、必ずしもステップである必要はなく、必ずしもプログラムである必要はなく、必ずしも入出力間の加工プロセスである必要はない。アルゴリズムはまた、多次元関係の組織様式であり、データ、エントリーポイント、関係、状態、結果の構造的ネットワークであり得る。それは秩序を含むこともできれば、ランダム性を含むこともできる。原因から結果へ進むこともできれば、結果から原因を逆生成することもできる。メタデータを変更せずに複数の状態に適合することができる。一つの結果を新しい出発点とすることもできれば、複数の出発点を同じ結果に収束させることもできる。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

真に高度なアルゴリズムとは、必ずしもより複雑な計算ではなく、計算を減らし、構造を生き生きとさせ、結果をエントリーポイントとし、因果を循環させる多次元的な組織方式である。

これが、私が「超次元アルゴリズム」を提案する核心的な意味である。それは普通の新語ではなく、新しい概念であり、新しいパラダイムであり、複数の現実システムにおいて既に成立している新しい計算構造である。その重点は繰り返し計算をより速くすることではなく、反復計算を減らすことである。計算能力を積み上げることではなく、エントリーポイントを再構築することである。データを絶えず修正することではなく、メタデータを維持しつつ関係を変えることである。結果を終点で止めることではなく、結果を新しい出発点とすることである。伝統的アルゴリズムは経路の中に結果を求める。超次元アルゴリズムは構造の中で結果を活性化する。伝統的アルゴリズムは計算の完了を追求する。超次元アルゴリズムは計算が伝統的な形態で存在しなければならないのかを問う。これが、現在のアルゴリズム定義との間の最も根本的な差異である。

付録：本稿の境界と未解決問題

本稿は超次元アルゴリズムの初歩的枠組みを提案するものであり、完全な形式化定義ではない。以下の問題はさらなる展開を待つ。これらは本稿の定義に対する否定ではなく、まさに次のステップで探求すべき方向である。

第一に、収束条件。超次元アルゴリズムの「完了」の指標は何か？ 結果が「有効」であると判断する方法は？ 伝統的アルゴリズムでは、答えは入出力の対応関係によって定義される。超次元アルゴリズムでは、答えは構造的一貫性によって定義される必要があるかもしれない。この定義はまだ完成していない。

第二に、リソース表現。多次元的な重ね合わせ状態を有限のリソースでどのように表現するか？ 時間、空間、計算能力といった伝統的なリソース指標は依然として適用可能か？ 適用可能であれば、どのように再定義するか？ 適用可能でなければ、どのような指標が代替するか？ これらの問題は探求を待つ。

第三に、秩序の保証。「ランダムでありながら秩序立っている」における「秩序」はどのような規則によって保証されるのか？ ランダム性と秩序の境界はどこにあるのか？ どのような状況でランダム性が秩序を破壊するのか？ どのような状況で秩序がランダム性を抑圧するのか？ これらの問題はさらなる研究を必要とする。

第四に、逆方向推論における選別。結果から複数の出発点を逆推論する際、異なる出発点の優劣をどのように選別または評価するか？ 「逆方向推論効率」のような尺

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

度は存在するか？ 存在するなら、それは順方向計算の効率尺度とどのような関係にあるのか？

第五に、伝統的システムとのインターフェース。超次元アルゴリズムは既存のチューリングマシン体系とどのように協調するのか？ 代替なのか、補完なのか、階層化なのか？ 実際の工学において、超次元アルゴリズムと伝統的アルゴリズムの境界はどのように画定されるのか？ ハイブリッドモードは存在するか？

第六に、検証枠組み。超次元アルゴリズムの結果はどのように検証するのか？ 結果が線形ステップを通じて得られたものでない場合、再現可能性や検証可能性をどのように確立するのか？ 完全に異なる検証哲学が必要か？

これらの問題は本稿の欠陥ではなく、「アンカー文献」としての必然的な特徴である。新しいパラダイムの提案には常に未解決問題が伴う。本稿は始まりであり、結論ではない。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

参考文献的注記

本稿で提案する「超次元アルゴリズム」は、既存のいかなる学術流派の直接的な拡張でもない。しかし、思想的な観点からは、以下の分野が本稿の対話的背景を提供している。読者の位置づけの参考のためであり、伝統的な意味での学術的引用ではない。

チューリングマシンと計算可能性理論は、本稿が伝統的アルゴリズムと比較するための基準点として機能する。逆問題とベイズ推論は、「結果から原因を推論する」既存の試みとして機能する。本稿の「果因論」はこれと関連するが、方向性が異なる。創発理論と複雑系は、「構造が結果を顕現する」既存の記述として機能する。本稿の超次元アルゴリズムは、創発を「理解可能」かつ「導き可能」にすることを試みる。ナレッジグラフとグラフデータベースは、「データが複数のノードで交差する」既存の実践として機能する。超次元アルゴリズムは、この基盤の上で、「可変エントリーポイント」という次元を優先する。非古典的計算パラダイム（量子計算、アナログ計算、ニューロモルフィック計算などを含む）は、古典的ビットや古典的アーキテクチャを突破するが、超次元アルゴリズムは「入力 → ステップ → 出力」という線形の枠組みそのものを突破することにより重点を置く。

本稿と上記の分野との関係は、継承や反駁ではなく、対話と超越である。本稿の目標は、これらの分野の内部で増分的な改善を行うことではなく、それらの上位に新たな問いのレベルを提起することである。

関連記事

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [普及] 私はアルゴリズムを持たない、しかしアルゴリズムを超越する!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [極限哲学] 果因論
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [極限普及] アルゴリズムによる掌握を拒絶する
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Indexed and catalogued in WorldCat (OCLC); permanently archived in TROVE (National Library of Australia)
Permanent archival record (DataCite Concept DOI): https://doi.org/10.5281/zenodo.17646306

الأبعاد فائقة الخوارزمية [المتطرفة الخوارزمية]

نفسها "الحوسبة" تعريف إعادة

المؤلف: تشاوهوي وو JEFFI CHAO HUI WU

الملخص

تُبنى .للخوارزمية التقليدي التعريف لحدود هيكلية كمراجعة "الأبعاد فائقة الخوارزمية" مفهوم الورقة هذه تقدم المحدودية على التركيز مع ،والمخرجات والقواعد والخطوات للمدخلات الخطي الإطار على عادةً التقليدية الخوارزميات العملاقة الكمبيوتر أجهزة أن من الرغم على .والمخرجات المدخلات بين الواضحة والحدود والجدوى والفعالية والحتمية على راسخة خوارزمية نماذج أساسي بشكل ينفذ يزال لا الأساسي منطقها فإن ،للغاية عالية حوسبة قدرة تمتلك الحديثة بل ،الحوسبي المقياس نمو مجرد ليس "الفائقة الحوسبة" في حقًا المناقشة يستحق ما بأن الورقة هذه تجادل .أوسع نطاق ولا ،التقليدية الخوارزميات من مسرعة نسخة ليست الأبعاد فائقة الخوارزمية .نفسه الحوسبي النموذج في جوهري تغيير هذا في .منتظمة ولكنها عشوائية ،متراكبة ،الأبعاد متعددة هيكلية حوسبية رؤية هي بل الرياضية؛ للخوارزميات امتدادًا وقت في متعددة ونتائج متعددة بداية نقاط تمتلك عقدة سمة بل ،نهائي مخرج أو سلبي إدخال مجرد البيانات تعد لم ،الهيكل بشكل التعديل إلى الوصفية البيانات تحتاج لا جديدة؛ دخول نقطة يصبح أن يمكن بل ،نهاية نقطة مجرد يعد لم واحد؛ والتفعيل العلائقية والتغيرات الهيكلية الدخول نقاط خلال من مختلفة ونتائج حالات مع تتوافق أن يمكن بل ،متكرر يمكن كيف لتوضيح "البنطال خصر طي" اليومي والمثال "والسبب التأثير نظرية" فلسفة الورقة هذه تستخدم .الشرطي وبالتالي ،الهيكلية الدخول نقطة تغيير طريق عن مباشرة صالحة المستهدفة النتيجة تجعل أن الأبعاد فائقة للخوارزمية هيكلية وحدود أولي تعريف وضع إلى الورقة هذه تهدف .الأصلي الحساب مسار تجاوز وحتى المتكرر الحساب تقليل ،الأبعاد فائقة للحوسبة المستقبلي للتوسع أصلية نظرية عقدة وتوفير ،الأبعاد فائقة للخوارزمية فكرية وأسس الخوارزمية فإن ،"جديدًا نموذجًا يشكلان والصلاحية الوجود" لمعيار وفقًا .جديد علمي ونظام ،المتطرفة والخوارزميات الوقت في بالفعل صالحة ،متعددة مجالات عبر منهجي بشكل منه التحقق تم جديد حوسبي هيكلي كنموذج ،الأبعاد فائقة الحاضر.

المفتاحية الكلمات: بيانات والسبب؛ التأثير نظرية الأبعاد؛ فائقة حوسبة متطرفة؛ خوارزمية الأبعاد؛ فائقة خوارزمية جديد علم خطية؛ غير سببية هيكلية؛ دخول نقطة حوسبي؛ نموذج وصفية؛

مقدمة: "الخوارزمية" مناقشة إعادة إلى نحتاج لماذا

لتغليف ولا ،أكبر اسمًا الموجودة الخوارزميات إعطاء بهدف فليس ،"الأبعاد فائقة الخوارزمية" أقترح عندما البشرية فهم هل :جوهرية أكثر سؤال مناقشة أريد ،تمامًا العكس على .مبتكر جديد مفهوم في التقليدية الخوارزميات للغاية؟ ضيق إطار في محصور للخوارزمية الحالي

،المنطقية والخطوات ،البرمجة وأكواد ،الرياضية الصيغ في عادةً يفكرون ،الخوارزميات عن الناس يتحدث عندما ،اليوم ،التلقائية والتوصية ،البحث نتائج وترتيب ،المسارات وتحسين ،البيانات ومعالجة ،النماذج وتدريب ،والإخراج والإدخال ،الحديثة الكمبيوتر أجهزة تشغيل تدعم وهي ،الخوارزميات من مهمة أشكال بالتأكيد هذه .الاصطناعي بالذكاء والاستدلال ،ذلك ومع .المعلومات ومجتمع ،الصناعية والأنظمة ،الفائقة والحوسبة ،الاصطناعي والذكاء ،البيانات وقواعد ،والإنترنت

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة هي.
Chinesische Originalfassung maßgeblich.

على قائمة بمعالجة تمر ،بمدخل تبدأ" كعملية فقط ،"المرتبة الخطوات من مجموعة" أنها على فقط الخوارزمية فهمنا إذا الأبعاد منخفض إطار في الحوسبة يحصر ذاته بحد الفهم هذا فإن ،"بمخرج وتنتهي ،قواعد.

الذكاء إلى بالطقس التنبؤ من ،التوصية أنظمة إلى البحث محركات من .الخوارزميات عليه تهيمن عصر في نعيش أن للخوارزميات يمكن هل :يُطرح ما نادرًا سؤال هناك لكن .البشري الذكاء حدود هي الخوارزميات حدود ،الاصطناعي حساب يجب لماذا ؟"مخرج ← خطوات ← مدخل" الخطي النمط الحوسبة تتبع أن يجب لماذا فقط؟ الطريقة بهذه تكون حوسبي نموذج واقتراح الافتراضية الافتراضات هذه تحدي الورقة هذه تحاول مرة؟ كل في الصفر من المشكلة نفس الخوارزمية فإن ،"جديدًا نموذجًا يشكلان والصلاحية الوجود" لمعيار وفقًا .الأبعاد فائقة الخوارزمية — تمامًا مختلف الوقت في بالفعل صالحة ،متعددة مجالات عبر منهجي بشكل منه التحقق تم جديد حوسبي هيكلي كنموذج ،الأبعاد فائقة الحاضر.

فائقة الحوسبة" عن تمامًا مختلف مفهوم هي الورقة هذه في المقترحة "الأبعاد فائقة الخوارزمية" أن ملاحظة المهم من ثلاثين من يقرب ما منذ الأكاديمية الأوساط في الموجودة (Hyperdimensional Computing, HDC) "الأبعاد ،كفاءة أكثر وحساب تمثيل نحو سعيًا ،والعمليات للتشفير للغاية الأبعاد عالية عشوائية ثنائية متجهات HDC تستخدم .عامًا :كليًا مختلفة فهي الأبعاد فائقة الخوارزمية أما ."مخرج ← معالجة ← مدخل" الخطي النموذج ضمن باقية تزال لا لكنها وما ،نفسه الحساب مسار تجاوز يمكن كان إذا وما ،الهيكل خلال من مباشرة نتيجة إنشاء يمكن كان إذا عما تتساءل إنها .متضاد مضمونهما لكن ،متشابهة أسماؤهما .متعددة نتائج مع التكيف مع تعديل دون الوصفية البيانات ترك يمكن كان إذا توجد أن يجب كان إذا ما" حول مسبق سؤال طرح إعادة بل ،للخوارزميات ترقية ليست الأبعاد فائقة الخوارزمية أصلاً خوارزمية".

الأول الجزء: التعريف المعياري الحالي للخوارزمية

:طويلة لفترة قبوله تم للخوارزمية أساسي تعريف هناك ،والهندسة والرياضيات الكمبيوتر علوم في السائدة الأنظمة في وتحولها ،أكثر أو واحدًا مدخلاً تأخذ ،الخطوات من محدود عدد من تتكون ،جيدًا محددة حوسبة عملية هي الخوارزمية إجماع هو بل ،شخص لأي عابرًا رأيًا ليس الفهم هذا .محدود زمن في أكثر أو واحدًا مخرجًا وتنتج ،حتمية قواعد بواسطة وقواعد ،والأجهزة ،للبرامج الأساسي المنطق يدعم إنه .الحديثة الحوسبة لحضارة الطويل التطور مدى على تشكل أساسي فإن ،الخوارزمية تعقيد بلغ مهما .الفائقة الحوسبة ومراكز ،الاصطناعي الذكاء ونماذج ،الشبكات وأنظمة ،البيانات والمخرجات ،والخطوات ،والقواعد ،المدخلات حول يدور يزال لا جوهرها.

جوهرية خصائص عدة إلى هذا المعياري الخوارزمية مفهوم تحليل يمكن.

لا ما إلى التكرار في تستمر أن يمكن لا الخطوات؛ من محدود عدد في الخوارزمية تنتهي أن يجب .المحدودية ،أولاً خوارزمية اعتبارها يصعب ،شكليًا وصفها يمكن كان لو حتى ،أبدًا تتوقف لا التي العملية .الأبد إلى تعمل أن ولا ،نهاية واضحة إنهاء شروط لها البيانات معالجة وتدفقات العلمية الحوسبة برامج جميع .فعالة.

بعض كانت إذا حتى .المخرج نفس الخوارزمية تنتج أن يجب ،الظروف نفس وتحت المدخلات بنفس .الحتمية ،ثانيًا للتفسير قابلة احتمالية آلية أو ،للتكرار قابلة عشوائية أو ،مضبوطة عشوائية عادةً فهي ،عشوائية تقدم الخوارزميات وهذا للتكرار؛ قابلة العددية المحاكاة نتائج تكون أن يجب .تمامًا للتحكم قابلة غير جوهرية عشوائية وليست ،إحصائيًا العلمية للحوسبة أساسي مطلب.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية هي المعتمدة.
Chinesische Originalfassung maßgeblich.

ونقطة البداية نقطة آخرًا؛ والمخرج ،الوسط في والمعالجة ،أولاً المدخل .والمخرجات المدخلات بين واضحة حدود ،ثالثًا الحد هذا نتيجة؛ تعطيك ،المعالجة بعد ،للخوارزمية تعطيها التي البيانات كانت مهما .واضحة هيكلية حدود لهما النهاية الترتيب عكس يمكن لا آخرًا؛ مخرج ثم ،معالجة ثم ،أولاً مدخل .واضح.

قفزات على تحتوي أن يمكن ولا فعليًا؛ للتنفيذ قابلة عملية الخوارزمية في خطوة كل تكون أن يجب .الفعالية ،رابعًا أو ،وورقة بقلم تنفيذها للإنسان يمكن أساسية عملية خطوة كل تكون أن يجب .منها التحقق أو وصفها أو تنفيذها يستحيل إلخ ،بيانات كتابة/قراءة ،منطقية أحكام ،قسمة ،ضرب ،طرح ،جمع — فعليًا تنفيذها للكمبيوتر يمكن.

موارد أو الذاكرة أو الحوسبية القدرة أو الوقت كان إذا ،نظريًا صالحة الخوارزمية كانت إذا حتى .الجدوى ،خامسًا الخوارزمية .الهندسية الناحية من فعالة خوارزمية اعتبارها الصعب فمن ،تمامًا مقبولة غير تستهلكها التي التخزين هندسيًا للتنفيذ قابلة خوارزمية تعتبر لا السنين من الملايين مئات تستغرق والتي نظريًا الصحيحة.

لنظرية الأساسي التجريد باعتبارها .تورينج آلة نموذج إلى النهاية في الخوارزمية مفاهيم من الكاملة المجموعة هذه تعود ،اليوم الكمبيوتر أجهزة قوة مدى عن النظر بغض ."للحوسبة القابلية" لـ الأساسي الحد تورينج آلة تحدد ،الحديثة الحوسبة حقًا ينحرف لم الأساسي منطقها فإن ،الفائقة الحوسبة مراكز حجم عن النظر وبغض ،الرقائق تقدم مدى عن النظر وبغض هذه لتنفيذ تسريع إلا هي ما الحديثة العملاقة الكمبيوتر أجهزة .المخرجات ،الخطوات ،القواعد ،المدخلات :المسار هذا عن الحوسبة" ،التقليدي السياق في ،أي .تعقيدًا أكثر أنظمة وجدولة ،أعلى وتوازي ،أكبر أجهزة مقياس خلال من العملية تزيد أن يمكن .الحوسبي النموذج في جوهريًا تغييرًا بالضرورة وليس ،الحوسبة مقياس في توسع الأساس في هي "الفائقة ،"مخرجات ،خطوات ،مدخلات" هو الأساسي المنطق ظل إذا ولكن ،المرات من الآلاف عشرات أو آلافًا الحوسبية القدرة التقليدية الخوارزمية نموذج ضمن باقية تزال لا فإنها.

الأبعاد فائقة الخوارزمية تعريف :الثاني الجزء

الخوارزميات من محسنة نسخة ليست إنها .بالتحديد السياق هذا من أقترحها التي "الأبعاد فائقة الخوارزمية" تظهر مختلف هيكلي فهم إنها .تقدمًا أكثر برمجة تقنية ولا ،تعقيدًا أكثر رياضية صيغة ولا ،أسرع خوارزمية ولا ،التقليدية تمامًا.

على تقتصر لا أنها هو الأول المعنى .معنيان هنا "الأبعاد فائقة" لـ ."الأبعاد فائقة" مصطلح معنى شرح إلى نحتاج ،أولاً تحاول أنها هو الثاني المعنى .واحد وقت في بالانطلاق متعددة لأبعاد تسمح بل ،الخطوات من البعد أحادي خطي تسلسل حتمية أو ،متسلسلة أو ،رياضية تكون لأن مضطرة الخوارزمية تعد لم — "الحوسبة" لـ التقليدية التعريفية الحدود تجاوز. للوصول D، C، B عبر وتمر ،A النقطة من تبدأ أن يجب :واحد اتجاه ذو طريق على القيادة تشبه التقليدية الخوارزمية أن يمكن الحوسبة أن تعتقد إنها .واحد اتجاه ذو طريقًا تكون أن يجب الحوسبة أن الأبعاد فائقة الخوارزمية تعتقد لا .Z إلى تكون أن عقدة لأي ويمكن ،دخول نقطة تكون أن عقدة لأي يمكن حيث ،الأبعاد متعدد هيكلًا أو ،حقلًا أو ،شبكة تكون خروج نقطة.

،مرتبة واحدة حوسبة بالضرورة وليست ،رياضية خوارزمية بالضرورة الخوارزمية ليست ،الجديد العلمي نظامي في ،مدخل" الخطي للنمط للامتثال بالضرورة مضطرة وليست ،ثابتة خطوات طول على بالعمل بالضرورة ملزمة وليست كل ،الهيكل هذا في .منتظم لكن عشوائي ،متراكب ،الأبعاد متعدد هيكل هي الأبعاد فائقة الخوارزمية ."مخرج ،معالجة نقطة يكون أن يمكن :واحد وقت في متعددة أدوارًا يمتلك بل ،ثابتًا مخرجًا ولا معزولاً مدخلاً ليس البيانات من جزء متعددة بداية لنقاط المشترك العمل عن الناتجة النتيجة أيضًا يكون أن ويمكن ،متعددة لنتائج البداية.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة هي.
Chinesische Originalfassung maßgeblich.

البيانات الأبعاد فائقة الخوارزمية تضع بينما ،العملية تدفق ضمن البيانات التقليدية الخوارزميات تضع ،أخرى بعبارة كل .مخرجات وبيانات ،وسيطة وبيانات ،مدخلات بيانات إلى عادةً البيانات تُقسم ،التقليدية الخوارزميات في .هيكل ضمن ومع .وسيطة عملية هي الوسيطة والعملية ،نتيجة هي والنتيجة ،مدخل هو المدخل .واضح ودور موقع له البيانات من نوع قد آخر؛ اتجاه في بداية ونقطة اتجاه في نتيجة تكون قد .فقط واحدة هوية للبيانات ليس ،الأبعاد فائقة الخوارزمية في ،ذلك ،سلبية مادة مجرد البيانات تعد لم .آخر هيكل في جديدة نتائج تطلق دخول نقطة وتصبح هيكل في استدعاؤه يتم كائنًا تكون .الأبعاد متعددة علائقية عقدة بل

أيضًا وهو ،متعددة لنتائج البداية نقطة هو البيانات من جزء كل :الأبعاد فائقة الخوارزمية جوهر مع تمامًا يتوافق هذا فائقة الخوارزمية تقترب واحدة؛ نتيجة لإنتاج "مرتبة خطوات" التقليدية الخوارزميات تستخدم .متعددة بداية لنقاط نتيجة بل ،موجودة لتصبح النتيجة تُحسب لا ،والنظام العشوائية تراكب في ،حيث الأبعاد متعدد حالة هيكل من أكثر الأبعاد .واحد آن في النتيجة مكونات من ومكون البداية نقطة هي البيانات ،النظام هذا في .الهيكل داخل طبيعي بشكل تظهر

.التالية الخمسة الجوانب إلى أحللها ،أوضح بشكل الأبعاد فائقة للخوارزمية الأساسية الخصائص لتقديم

المنطق باستخدام بالكامل وصفها يمكن — بطبيعتها رياضية هي السائدة الخوارزميات .الرياضية غير الطبيعة ،أولاً علاقات أو ،فيزيائية مبادئ إلى تستند قد .رياضية بالضرورة ليست الأبعاد فائقة الخوارزمية .الرياضية والصيغ الرمزي خلال من بالضرورة الحوسبة تتم أن يجب لا .الوعي مبادئ حتى أو ،المعلومات لنظرية جديد نموذج أو ،هندسية شكل ،المثال سبيل على .أبعادًا أعلى فضاء في هندسية علاقات خلال من طبيعي بشكل نفسها تقدم فقد ؛"رياضية عمليات" نفسها الفيزياء إنها — تحسب "رياضية خوارزمية" ليست هذه .السطحي التوتر تقليل مبدأ تحدده الصابون فقاعة .لمحاكاتها الرياضية الصيغ استخدام من بدلاً ،"الحوسبة" من النوع هذا فهم الأبعاد فائقة الخوارزمية تحاول ."تحسب"

.خطوة كل في فقط واحدة بحالة ،واحد بُعد في مرتبة خطوات التقليدية الخوارزميات تنفذ .والتراكب الأبعاد تعدد ،ثانيًا تتبع لا .والتعايش بالتراكب متعددة وحالات ،واحد وقت في بالوجود متعددة لأبعاد الأبعاد فائقة الخوارزمية تسمح هذا من "تظهر" بل "محسوبة" ليست النتيجة .الأبعاد متعدد حالة حقل في تنطلق ،ذلك من بدلاً واحدًا؛ مسارًا الخوارزمية .يذهب أين ماء جزيء كل تخبر "خوارزمية" توجد لا ،الهواء في ثلج رقاقة تتشكل عندما ،المثال سبيل على .الحقل الثلج رقاقة شكل .معًا تعمل — الهواء تدفق ،الرطوبة ،الحرارة درجة — متعددة أبعاد نتيجة هو الماء جزيئات ترتيب ."يحسب" أن من بدلاً "يظهر".

فيه؛ التحكم يمكن واضطراب ،خارجية ،أداة هي العشوائية ،التقليدية الخوارزميات في .النظام داخل العشوائية ،ثالثًا ويعملان يتعايشان والنظام العشوائية .وهيكلية جوهرية العشوائية ،الأبعاد فائقة الخوارزمية في .حتمية نفسها الخوارزمية عضوي مكون نفسها العشوائية" بل ،"عشوائية على تحتوي خوارزمية" ليس هذا .الخوارزمية جوهر معًا ويشكلان ،معًا كثافة يظهر إجمالاً لكنه ،عشوائيًا يبدو الأشجار توزيع — للغابة البيئي الهيكل ،المثال سبيل على ."الخوارزمية في .البيئي للهيكل الطبيعي للعمل أساسي شرط بل ،خطأ ليست "العشوائية" تلك .الأنواع بين وعلاقات منتظمة

هو والمخرج ،مدخل هو المدخل — مفرد البيانات دور ،التقليدية الخوارزميات في .للبيانات المتعددة الأدوار ،رابعًا نقطة الوقت نفس في هو البيانات من جزء كل ،الأبعاد فائقة الخوارزمية في .وسيطة نتيجة هي الوسيطة والنتيجة ،مخرج نتائج مسارات إلى النهر مجرى اتجاه في تنطلق أن البيانات لعقدة يمكن .متعددة بداية لنقاط ونتيجة متعددة لنتائج بداية عقدة بل ،خطي تدفق في نقطة البيانات تعد لم .متعددة أسباب المنبع اتجاه في عندها تتقارب أن أيضًا ويمكن ،متعددة تدخله — "مدخل" هو الطول ،التقليدية الخوارزمية في .ما شخص طول تأمل ،المثال سبيل على .شبكة في تقاطع بشكل يتحدد — "نتيجة" أيضًا هو الطول لكن .ذلك إلى وما ،الملابس ومقاس ،الوزن توقع مثل مخرجات على للحصول

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

بيانات ،الأبعاد فائقة الخوارزمية في .الرياضية والتمارين والتغذية الوراثة مثل متعددة "بداية نقاط" خلال من مشترك بينهما اختيارًا وليس ،ونتيجة بداية نقطة الوقت نفس في هي الطول.

التقليدية الخوارزميات تعدل أن إما ،مختلفة مشاكل مواجهة عند .متغيرة والعلاقات ،ثابتة الوصفية البيانات ،خامسًا الأساسية السمات تبقى — الوصفية البيانات الأبعاد فائقة الخوارزمية تعدل لا .شيء كل حساب تعيد أو ،نفسها البيانات ،الوصفية البيانات هيكل نفس .البيانات بين والعلاقات ،التفسير وأنماط ،الدخول نقاط هو يتغير ما .تغيير دون للبيانات واستخدم ،واحدة مرة احسب :يعني هذا .تمامًا مختلفة نتائج يقدم أن يمكن ،مختلفة دخول نقاط خلال من تفعيله يتم الذي كوثيقة تحليله أو ،كشيفرة تشفيره فك أو ،كشعر قراءته يمكنك .النص نفس تأمل ،المثال سبيل على .نهائية لا مرات :بالتحديد القدرة هذه إلى الأبعاد فائقة الخوارزمية تسعى ."الدخول نقطة" فقط غيرت لقد — يتغير لم نفسه النص .تاريخية "متعددة نتائج ،متعددة دخول نقاط ،نفسها بيانات".

مع تتوافق ،مختلفة وشروط دخول نقاط خلال من بل نفسها؛ الوصفية البيانات تعديل يتم لا ،البيانات هيكل منظور من يتم ،تغيير دون هيكلها على الحفاظ مع بل ،متكرر بشكل تُعالج البيانات تعد لم .الهيكل نفس داخل متعددة ونتائج بداية نقاط بينما ،"مختلفة لنتائج البيانات معالجة" هو التقليدي النهج .مختلفة نتائج يقدم مما ،مختلفة دخول نقاط خلال من تفعيلها "متعددة محتملة نتائج لحمل البيانات نفس استخدام" هي الأبعاد فائقة الخوارزمية".

الثالث الجزء: الاختلافات بين الجوهرية الأبعاد فائقة الخوارزمية والخوارزميات التقليدية

أعلاه التعريفات على بناءً ،تُظهر الخوارزمية الأبعاد فائقة والخوارزميات التقليدية الخوارزمية أساسية أبعاد عبر جوهرية اختلافات هذه .متعددة الاختلافات مفصلة أدناه واحدة تلو الأخرى.

الجوهرية الطبيعة حيث من ،الخوارزميات التقليدية الخوارزمية وصورية رياضية ،قابلة للوصف الكامل باستخدام المنطق الرمزي إلى تستند قد رياضية؛ بالضرورة ليست الأبعاد فائقة الخوارزمية .بطبيعتها رياضية كائنات إنها الرياضية؛ والصيغ .أوسع فئة بل ،الرياضيات من فرعية مجموعة ليست إنها .واعية مبادئ حتى أو معلوماتية أو هندسية أو فيزيائية مبادئ "الكوني القانون" إلى "الرياضي الكائن" من :التالي النحو على الاختلاف هذا تلخيص يمكن".

تسلسليًا ،C إلى B إلى A خطوة ،خطيًا منظمًا فرديًا زمنيًا ترتيبًا التقليدية الخوارزميات تتبع ،الزمني الترتيب حيث من تسلسل الأبعاد فائقة للخوارزمية ليس .فيه رجعة لا الزمن سهم متوازية؛ مرتبة فرعية خطوات إلى للتحلل قابلة أو ،تمامًا لأنه ،التنفيذ "ترتيب" بـ مرتبطة النتيجة تكون لا قد .منتظمة لكنها وعشوائية ومتراكبة الأبعاد متعددة إنها ثابت؛ خطوات إلى "الخطي الزمن" من :التالي النحو على الاختلاف هذا تلخيص يمكن .الأول المقام في تقليدي "ترتيب" يوجد لا "الأعلى الأبعاد التزامن".

يكون ،الأولية الحالة ونفس المدخلات نفس إلى بالنظر :حتمية سببية سلسلة التقليدية الخوارزميات تتبع ،السببية حيث من :متراكبة سببية الأبعاد فائقة الخوارزمية تتبع .فيه رجعة لا الاتجاه أحادي سهم ،النتيجة يسبق السبب دائمًا؛ فريدًا المخرج — "والسبب التأثير نظرية" فلسفتي هي هذه .متعددة بداية لنقاط ونتيجة متعددة لنتائج بداية نقطة هو البيانات من جزء كل يمكن .جديدة بداية نقطة تصبح أن يمكن بل ،نهاية نقطة ليست النتيجة .السبب ثم ،أولاً النتيجة تكون أن الممكن من "بالكامل مترابطة سببية شبكة" إلى "واحدة نتيجة ،واحد سبب" من :التالي النحو على الاختلاف هذا تلخيص.

إلى ثم المعالجة إلى المدخلات من الاتجاه أحادي تدفق هيكل التقليدية الخوارزميات تتبع ،البيانات هيكل حيث من بيانات وتبقى ،النهاية إلى البداية من مدخلات المدخلات بيانات تبقى واضحة؛ بحدود واحد دور للبيانات المخرجات؛

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

لا بداية؛ ونقطة نتيجة الوقت نفس في هي البيانات نفس متعددة؛ أدوار للبيانات ،الأبعاد فائقة الخوارزمية في .نتائج النتائج من :التالي النحو على الاختلاف هذا تلخيص يمكن .شبكة في عقد فقط ،مطلقة نهاية نقاط ولا مطلقة بداية نقاط توجد "التراجعي التكافل" إلى "الاتجاه أحادية السيولة".

حسبت قد كانت لو حتى ،مشكلة تواجه مرة كل في الصفر من التقليدية الخوارزميات تحسب ،الحوسبة نمط حيث من بالعملية تمر تزال لا ،والأولى الألف المرة في بالفعل؛ معروفة الإجابة كانت لو حتى ،قبل من مرة ألف مماثلة مشاكل تشغيلها؛ لإعادة حاجة فلا ،مقابلة نتيجة هناك كان إذا ،الأبعاد فائقة الخوارزمية في ."الحالة عديمة" حوسبة هذه .بأكملها ،حالة لها الحوسبة .المعاكس الاتجاه في السبب مسارات نشر ويمكن ،متعددة بداية نقاط استنتاج يمكن ،واحدة نتيجة من "مرة كل في الحساب إعادة" من :التالي النحو على الاختلاف هذا تلخيص يمكن .الاستخدام لإعادة وقابلة ،ذاكرة ولديها "واحدة مرة الاستخدام إعادة" إلى.

العشوائية الأرقام الخارج؛ من محقونة أو عشوائية شبه هي التقليدية الخوارزميات في العشوائية ،العشوائية دور حيث من الأبعاد فائقة الخوارزمية في العشوائية .حتمية نفسها الخوارزمية التحسين؛ أو التقريب أو العينات لأخذ أدوات مجرد هي على تحتوي خوارزمية" ليس هذا .النظام مع ويتعاون يتعايش ،عضوي مكون هي العشوائية وتأسيسية؛ جوهرية هي :التالي النحو على الاختلاف هذا تلخيص يمكن ."العمل على الخوارزمية قدرة أسباب أحد هي العشوائية" بل ،"عشوائية "التأسيسية العشوائية" إلى "الأداتية العشوائية" من.

هي الموارد الموارد؛ إلى بالنظر دقة وأكثر أسرع بشكل الحساب إلى التقليدية الخوارزميات تسعى ،الموارد فهم حيث من غير نفسها الحوسبة جعل إلى الأبعاد فائقة الخوارزمية تسعى ."القيود ظل في الحساب إكمال" هو الخوارزمية هدف قيود؛ الموارد تُستخدم لا ؛"الحساب تجاوز" بل ،"أسرع بشكل الحساب" ليس الهيكلي؛ التصميم خلال من ضرورية قيود ظل في التحسين" من :التالي النحو على الاختلاف هذا تلخيص يمكن ."الدخول نقاط تصميم"لـ بل ،"استهلاكها"لـ "الهيكلي التصميم خلال من الإزالة" إلى "الموارد".

والسبب التأثير نظرية فلسفة :الرابع الجزء

الفكرية الأسس أحد هو هذا .أكبر بشكل "والسبب التأثير نظرية" فلسفة تطوير إلى أحتاج ،أعلاه المقارنة على بناءً الأبعاد فائقة للخوارزمية الأساسية.

هذا يتجلى .المخرج ثم ،أولاً المدخل النتيجة؛ ثم ،أولاً فالسبب النتيجة؛ يسبق السبب أن التقليدي التفكير يفترض ما عادةً لو حتى .النهاية نقطة إلى بخطوة خطوة وتمشي البداية نقطة من تبدأ أن يجب :التالي النحو على الخوارزميات في المفهوم "الإثبات مسار" إلى تفتقر لأنك — الإجابة تلك "استخدام" مباشرة يمكنك لا ،الإجابة هي ما تعرف كنت.

وتستمر جديدة بداية نقطة تصبح أن يمكن ،النتيجة ظهور بمجرد .بالضرورة نهاية نقطة مجرد ليست النتيجة ،هيكلي في بداية نقاط استنتاج يمكن .الأسباب تكوين في رجعي بأثر تشارك أن للنتيجة يمكن .جديدة هياكل توليد في المشاركة في وعلاقات ،دورية علاقات يشكلان بل ،الاتجاه أحادي تسلسل في مرتبين والنتيجة السبب يعد لم .نتيجة من محتملة متعددة الأبعاد متعددة وعلاقات ،شبكية.

فائقة الخوارزمية تسأل "النتيجة؟ على نحصل كيف ،السبب إلى بالنظر" :التقليدية الخوارزميات تسأل ،أخرى بعبارة "السبب؟ نبني أو نستنتج كيف ،النتيجة إلى بالنظر" :الأبعاد

بل ."شيء لا من تنشأ النتائج" أن يدعي ولا ،للسببية نفيًا ليس "السبب ثم ،أولاً النتيجة وجود" أن توضيح المهم من بعضهما إلى ويتحولا البعض لبعضهما دخول كنقاط يعملا أن والنتيجة للسبب يمكن ،الأبعاد عالية الهياكل في أنه يوضح

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

رجعي بأثر أجد ثم ،أولاً "الأرض على جر دون ارتداؤه يمكن" النتيجة تحديد يتم ،التالي البنطال مثال في كما .البعض أن بل ،سبب لها ليس النتيجة أن ليس .الهيكل داخل "السبب" مستوى مع تتوافق والتي ،"الخصر طي" تشغيل نقطة السبب لاكتشاف جديدة دخول نقطة تصبح النتيجة.

الحالة من بخطوة خطوة تحسب أن يجب الاتجاه؛ أحادي الوقت أن هو التقليدية الخوارزميات عالم في أساسي افتراض إلى تفتقر لأنك ،مباشرة الإجابة تلك "استخدام" يمكنك لا ،الإجابة هي ما تعرف كنت لو حتى .النهائية الحالة إلى الأولية الأسباب استنتاج يمكن ،معروفة النتيجة كانت إذا :تحديدًا الافتراض هذا الأبعاد فائقة الخوارزمية تتحدى .الإثبات مسار نشر بل ،النهاية إلى البداية من سيرًا الحوسبة تعد لم متعددة؛ سببية مسارات مع تتوافق أن النتيجة لنفس يمكن بالعكس؛ النهاية من الممكنة البداية نقاط جميع.

الخصر وطي البنطال — يومي مثال :الخامس الجزء

يوميًا مثالاً سأستخدم ،حدسية أكثر بشكل أعلاه المجردة الاختلافات لفهم.

إزعاجًا يسبب مما ،الأرض على تسحبان ،قليلاً طويلتان البنطال ساقا .مطاطي خصر بحزام جديدًا بنطالاً شخص يشتري.

نما إذا .يجربه ثم ،وخيط بإبرة ويخيطه ،الساق من جزءًا يطوي ،جدًا طويلتان البنطال ساقي أن ملاحظة :التقليدي النهج جديدًا بنطالاً يشتري ،القادمة المرة في .البنطال ويُتلف ،المخيطة الساق إنزال يمكن لا ،جدًا قصيرًا البنطال وأصبح الطفل ساقي معالجة يتم لذلك ،البنطال ساقي في ظاهريًا تظهر المشكلة لأن ،جدًا طبيعية الطريقة هذه تبدو .أخرى مرة ويخيطه وأخيرًا ،الظاهرة موقع في معالجتها ،الظاهرة موقع ،المشكلة تحديد :التقليدية الخوارزميات تفكير مع هذا يتوافق .البنطال غير مسار .البيانات عدّل إذن خاطئة؟ بيانات .البنطال ساقي عدّل إذن جدًا؟ طويلتان البنطال ساقا .نتيجة على الحصول المسار طول على الترقيع في استمر إذن مناسب؟.

هذا .فورًا للارتداء قابلًا البنطال ويصبح ،واحدة مرة الخصر أطوي ،البنطال حاشية تعديل من بدلاً .مختلف نهجي ،دائم بشكل الحاشية أخيط ولا ،البنطال ساقي بقص أقوم لا .تمامًا مختلف وراءه الهيكلي المنطق لكن ،جدًا بسيط الإجراء والقماش ،والقصّة ،البنطال وطول ،الخصر محيط .للبنطال الوصفية البيانات أتلف ولا ،للبنطال الأصلي الطول أغير ولا البنطال ساقي يجعل مما ،للخصر الهيكلية الدخول نقطة أغير ببساطة أنا .جوهري بشكل الصفات هذه من أي تتغير لا — بأكملها الارتداء لحالة شاملة تحكم نقطة بل ،محلي موضع مجرد ليس هنا الخصر .الارتداء عند طبيعي بشكل تقصران مباشرة النهر مجرى اتجاه في المشكلة تختفي ،هذه التحكم نقطة بضبط.

الذي للطفل خاص بشكل واضح هذا .الاستخدام لإعادة وقابلة ،للتعديل وقابلة ،للعكس قابلة الطريقة هذه أن ،ذلك من الأهم عندما فترة بعد واحدة؛ مرة الخصر أطوي ،قليلاً طويلتان البنطال وساقا كافٍ غير الحالي الطفل طول كان إذا .ينمو ذلك ومع ،اًسليم للبنطال الأصلي الهيكل يبقى .بالكامل أرخيه ،أكثر يطول عندما قليلاً؛ الخصر أرخي ،الطفل يطول ،الطفل يطول عندما ،دائم بشكل الحاشية خيطت إذا ،التقليدية بالطريقة .مختلفة مراحل في الطفل طول مع التكيف يمكنه ،تغيير دون الوصفية البيانات على تحافظ طريقتي .مستحيلاً استعادته يصبح ،قصصته إذا جدًا؛ قصيرًا البنطال يصبح قد متعددة حالات مع بالتكيف الهيكل لنفس يسمح مما.

مهمة هيكلية اختلافات عدة عن المثال هذا يكشف.

باتباع ،البنطال ساقي تعديل يجب لذلك ،البنطال ساقي في المشكلة :التقليدية الخوارزميات منطق مع التقليدي النهج يتوافق في .فيه رجعة لا بشكل ،مرة كل في واحدة مشكلة حل .خطوة أي تخطي دون ،بخطوة خطوة "النتيجة إلى السبب" مسار

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة.
Chinesische Originalfassung maßgeblich.

هو الهدف :الأبعاد فائقة الخوارزمية منطق مع نهجي يتوافق .جديد من ابدأ ،المشكلة نفس فيها تحدث التي القادمة المرة — شاملة تحكم نقطة أجد ،ذلك من بدلاً ؛"جدًا طويلتان البنطال ساقا" السطحي السبب أحل لا أنا .الأرض على الجر عدم الثنية .وصفية بيانات أي أعدل لا أنا .فورًا المشكلة وتختفي ،تلقائيًا البنطال ساقا تقصر ،واحدة مرة الخصر بطي .الخصر وديناميكية للعكس وقابلة مؤقتة حالة سوى ليست الخصر عند.

الأصلية البيانات تفقد أقصر بشكل البنطال ساقي خياطة — الوصفية البيانات التقليدي النهج يعدل ،البيانات منظور من للعكس وقابلة مؤقتة حالة فقط مضيفًا ،سليمة للبنطال الأصلية الأبعاد تبقى وصفية؛ بيانات أي نهجي يعدل لا .دائم بشكل بين العلاقات ترتيب تعيد فقط إنها قديمة؛ بيانات تدمر ولا جديدة بيانات تخلق لا الخصر ثنية .الثنية — وديناميكية مباشر غير بشكل البنطال لساقي الفعال الطول وتغيير ،للخصر الفعال المحيط تقصير — البيانات.

ساقا" سبب من التقليدية الطريقة تبدأ .المستهدفة النتيجة هي "الأرض على تسحبان لا البنطال ساقا" ،المثال هذا في توضح ،ذلك من النقيض على ،طريقتي ."الجر عدم" نتيجة على تحصل وأخيرًا ،الحاشية وتعدل ،"جدًا طويلتان البنطال يجعل واحدة مرة الخصر طي أن أخيرًا وتكتشف ،هيكلية دخول نقطة عن للخلف تبحث ثم ،"الجر عدم" النتيجة أولاً السبب من بخطوة خطوة السير الضروري من ليس ."والسبب التأثير نظرية" لـ العملي المظهر هو هذا .صالحة النتيجة ضروري غير الأصلي المسار يجعل مما ،النتيجة من بالعكس الهيكل تحديد يمكن بل النتيجة؛ إلى.

هي الوصفية البيانات ،البنطال مثال في الوصفية؟ البيانات هي ما ."الوصفية البيانات" سؤال على أيضًا المثال هذا يجيب البيانات .للبنطال "الأساسية الصفات" هي هذه .القماش ،القصّة ،البنطال طول ،الخصر محيط :للبنطال الأصلية الأبعاد التقليدي النهج يعدل .مؤقت بشكل جزءًا تطوي فقط ،للبنطال الأصلية الأبعاد من أيًا تغير لا الخصر؛ ثنية هي المؤقتة تبقى — وصفية بيانات أي نهجي يعدل لا .الأصلية البيانات وتفقد ،دائم بشكل يقصر البنطال طول — الوصفية البيانات وديناميكية للعكس وقابلة مؤقتة حالة تضاف فقط ،سليمة للبنطال الأصلية الأبعاد.

ارتداء حالات تدعم إنها :"متعددة لنتائج البداية نقطة" الوقت نفس في هي ،وصفية كبيانات ،للخصر الأصلية الأبعاد .إلخ ،الخصر عند بثنيتين والارتداء ،الخصر عند واحدة بثنية والارتداء ،العادي الارتداء مثل ،الوقت نفس في متعددة وطريقة ،القماش اختيار مثل متعددة بداية نقاط خلال من مشترك بشكل تحدد إنها :"متعددة بداية لنقاط نتيجة" أيضًا وهي تعيد فقط إنها قديمة؛ بيانات تدمر ولا جديدة بيانات تخلق لا ،مؤقتة كبيانات ،الخصر ثنية .إلخ ،التصميمي والقصد ،القص البيانات بين العلاقات ترتيب.

إلى ينجذبون ،جدًا طويلين بنطال ساقي يرون عندما ،الناس من كثير .الهيكلي التفكير من نوع بل ،عادية خدعة ليست هذه البنطال ساقي كون فإن ،العام الهيكل من نظرنا إذا لكن .البنطال ساقي حول المعالجة كل يدور لذلك ،البنطال ساقي مظهر البنطال استقرار موقع على يؤثر أن يمكن الخصر موضع تغيير النهر؛ مجرى اتجاه في مظهر مجرد هو جدًا طويلين تحكم نقطة على تحتوي الأبعاد منخفضة المشاكل من العديد .تظهر حيث المشكلة حل الضروري من ليس ،أي .بأكمله بينما ،المشكلة سطح طول على الحساب مواصلة إلى التقليدية الخوارزميات تميل .أعلى هيكلي مستوى على لها حقيقية الهيكلية الدخول نقطة عن الأبعاد فائقة الخوارزمية تبحث.

المستوى رفيع الهيكل يعقّد لا غالبًا .أبسط تكون قد بل ،تعقيدًا أكثر ليست الأبعاد فائقة الخوارزمية لماذا أيضًا يفسر هذا تضيف قد ،معقدة مشكلة مواجهة عند .الصحيحة الدخول نقطة عند بسيطة المعقدة المشكلة يجعل بل ،المشكلة لكن .تدريب وقت وتضيف ،تخزينًا وتضيف ،حوسبية قدرة وتضيف ،نماذج وتضيف ،خطوات التقليدية الخوارزميات المسار يصبح قد ،صحيح بشكل العقدة هذه تفعيل بمجرد هيكلية؛ عقدة عن تبحث ،ذلك من بدلاً ،الأبعاد فائقة الخوارزمية ضروريًا المسار كان إذا ما تعريف إعادة بل ،كسلاً ليس "المسار تجاوز" بـ يسمى ما .صالح غير أصلاً المعقد.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة هي.
Chinesische Originalfassung maßgeblich.

نقطة تغيير" هي الأبعاد فائقة الخوارزمية بينما ،"المسار طول على النتيجة تعديل" هو التقليدي النهج :المثال هذا في الأخرى الطريقة بينما ،النتيجة طرف عند باستمرار ترقّع التقليدية الطريقة ."صالح غير بأكمله المسار يجعل مما ،الدخول البداية نقطة عند هيكلتها يعاد متعددة معالجات تتطلب كانت التي المشاكل يجعل مما ،الهيكلية الدخول نقطة مباشرة تغير بينما ،"النتيجة على للحصول المسار اتباع ،البداية نقطة من البدء" هو التقليدية للخوارزميات السائد الشكل .واحدة دفعة النهج ."جديدة هياكل توليد في وتشارك للمسار دخول نقطة تصبح أن يمكن نفسها النتيجة" ،الأبعاد فائقة الخوارزمية في ."متعددة نتائج حالات يحمل واحد هيكل" هي الأبعاد فائقة الخوارزمية بينما ،"واحدة نتيجة يقابل واحد هيكل" هو التقليدي

الوصفية البيانات تعديل عدم بيانات نظرة :السادس الجزء

نقاط على تطبيقها يمكن حتى ،الوصفية البيانات تعديل عدم :جدًا مهم مبدأ الأبعاد فائقة للخوارزمية ،البيانات منظور من متعددة نتائج أو بداية.

،مختلفة مشاكل مواجهة عند .الوجودية والمعلومات ،الأساسي والهيكل ،للبيانات الأصلية السمات هي الوصفية البيانات تنشئ أو ،البيانات حساب تعيد أو ،البيانات تعالج أو ،البيانات تنسخ أو ،البيانات التقليدية المعالجة طرق تعدل ما غالبًا ،باستمرار البيانات معالجة في تتمثل بتكلفة ولكن ،بالتأكيد المشاكل يحل أن يمكن النهج هذا .مختلفة لنتائج مختلفة مسارات .باستمرار النظام تعقيد وزيادة ،باستمرار المسارات وتكرار

نقاط تغيير خلال من ،الوصفية البيانات تعديل عدم محاولة مع أنه ،ذلك من النقيض على ،الأبعاد فائقة الخوارزمية تؤكد ونتائج متعددة بداية نقاط مع يتوافق أن الوصفية البيانات هيكل لنفس يمكن ،التفعيل وأوضاع والشروط والعلاقات الدخول تبقى .التقديم وضع يتغير تغيير؛ دون الأساسي الهيكل يبقى .العلاقات تتغير تغيير؛ دون البيانات وجودية تبقى .متعددة .مختلفة حالات مع تتكيف أن يمكن ذلك ومع ،سليمة الوصفية البيانات

ذاكرة في نتيجة تخزين مجرد ليس "واحدة مرة احسب" ."نهائية لا مرات واستخدم ،واحدة مرة احسب" بـ أعنيه ما هذا ،البيانات هيكل نفس .حالة لكل كامل مسار بناء إعادة إلى بحاجة يعد لم ،هيكل إنشاء بمجرد أنه يعني بل ،المؤقت التخزين بل ،"مختلفة لنتائج البيانات معالجة" ليست إنها .مختلفة نتائج يقدم أن يمكن ،مختلفة دخول نقاط خلال من تفعيله يتم الذي عن الأبعاد فائقة الخوارزمية تميز التي الجواهر أحد أيضًا هذا ."متعددة محتملة نتائج لحمل البيانات نفس استخدام" العام الهيكل الأبعاد فائقة الخوارزمية تعالج مسار؛ طول على البيانات التقليدية الخوارزميات تعالج .التقليدية الخوارزميات .والنتائج والعلاقات الدخول ونقاط للبيانات

نفس ،الوصفية البيانات تعديل بدون ،يمكّن نظام :التالي النحو على الأبعاد فائقة الخوارزمية فهم يمكن ،الأدنى هيكلها في يسعى لا التعريف هذا .الهيكلية الدخول نقاط في تغييرات خلال من متعددة نتائج مسارات مع التوافق من البيانات عقدة دالة ليست إنها .الهيكلية حدودها وضع إلى أولاً بل ،تقليدية رياضية صيغة إلى فورًا الأبعاد فائقة الخوارزمية اختزال إلى أن ويمكن ،مستقرة الوصفية البيانات تبقى :علائقية بنية إنها .بسيطة مؤقت تخزين آلية ولا ،ثابتة صيغة ولا ،واحدة خطية في .متعددة اتجاهات في النتائج تقديم ويمكن ،العلاقات تتغير أن ويمكن ،الشروط تتغير أن ويمكن ،الدخول نقاط تتغير .العلاقات بتفعيل بل ،خطوات بتنفيذ تتعلق الحوسبة تعد لم ،تحديدًا الهيكل هذا

:هو "متعددة نتائج أو بداية نقاط على للتطبيق قابلة يجعلها مما ،الوصفية البيانات تعديل عدم" جوهر ،البيانات منظور من عندما :هو التقليدي الهيكل ."الاستخدام دخول نقطة / التفسير طريقة" في التغيير يحدث تغيير؛ دون البيانات وجودية تبقى ،التفسيري الهيكل يتغير ،المشكلة تتغير عندما :الأبعاد فائقة الخوارزمية في .المسار أو البيانات تتغير ،المشكلة تتغير .البيانات وليس

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

السابع الجزء: إعادة تعريف "الفائقة الحوسبة"

"الفائقة الحوسبة" — مصطلح توضيح إلى أحتاج ،النظام هذا في.

حوسبية قدرة أو ،الرقائق من المزيد ليس .المعتاد بالمعنى العملاقة الكمبيوتر أجهزة ليس "الفائقة الحوسبة" بـ أعنيه ما "الأبعاد فائقة الخوارزمية" هو "الفائقة الحوسبة" بـ أعنيه ما .أكبر بيانات مراكز أو ،أعلى.

النظام يزال لا عندما .الحوسبة نموذج في تغيير ربما بل ،الحوسبية القدرة تراكم بالضرورة ليست الحقيقية الفائقة الحوسبة من هيكليًا يقلل أن لنظام يمكن عندما .المسارات داخل محاصرًا يظل فإنه ،قويًا كان مهما ،المتكرر الحساب على يعتمد الأبعاد فائقة الحوسبة من الاقتراب في يبدأ فإنه ،جديدة دخول نقاط النتائج ويجعل ،المتكررة المسارات ويتجاوز ،الحساب الحقيقية.

فائقة الخوارزمية تسعى ."أكبر مشاكل لحل أكبر حوسبية قدرة استخدام" إلى التقليدية الفائقة الحوسبة تسعى ،أخرى بعبارة لكن ،متعارضين ليسا النهجان هذان ."هائلة حوسبية قدرة إلى المشاكل تحتاج لا بحيث أفضل هيكل استخدام" إلى الأبعاد مختلفة اتجاهاتهما.

قد .الحوسبة فهم في تحولاً تمثل الأبعاد فائقة الخوارزمية فإن ،الحوسبية القدرة حد تمثل التقليدية الفائقة الحوسبة كانت إذا أكثر نماذج أو ،أعلى طاقة استهلاك أو ،أكبر بيانات مراكز أو ،الرقائق من المزيد الحقيقية "الفائقة الحوسبة" تكون لا ،نتيجة عقدة من عكسيًا نشرًا أو ،مسار إزالة أو ،هيكلية دخول نقطة اكتشاف الحقيقية "الفائقة الحوسبة" تكون قد .تعقيدًا ظلت إذا ،الحوسبية القدرة زادت كلما .الوصفية البيانات تعديل دون واحد وقت في صالحة متعددة نتائج لجعل طريقة أو عمليات مع حتى ،الهيكل رفع بمجرد المسارات؛ تلك داخل فقط قوية تظل فإنها ،الأبعاد منخفضة مسارات في محاصرة الكفاءة من أعلى مستوى يظهر أن يمكن ،للغاية بسيطة.

يجب كان إذا ما" لفرضية تعريف إعادة هي الأبعاد فائقة الخوارزمية الحوسبية؛ القدرة حد هي المزعومة الفائقة الحوسبة الحوسبية القدرة على الحوسبة تعتمد أن".

الثامن الجزء: الأسس التجريبية — ومعايير المجالات متعددة أنظمة عبر التحقق الجديد النموذج

الواقعية الأنظمة من العديد في بالفعل منها التحقق تم لقد .بحتًا نظريًا بناءً ليست الأبعاد فائقة الخوارزمية.

بموارد معقدة بجدولة يقوم فهو سحابي؛ دعم أو فائقة حوسبة قدرة إلى يحتاج لا بي الخاص اللوجستية الخدمات نظام من شيء كل حساب لإعادة حاجة ولا ،متكرر بشكل الهيكل تكييف ويمكن ،النتائج استخدام إعادة يمكن .نسبيًا منخفضة بموارد معقدة جدولة وإنجاز ،السحابة على الاعتماد وعدم ،الفائقة الحوسبة على الاعتماد عدم .مرة كل في الصفر فالتصميم الوحيد؛ الحل ليست العالية الحوسبية القدرة أن تظهر إنها .وقيمة قائمة هندسية اختراق نقطة هذه — منخفضة الحوسبية القدرة من جزء محل يحل أن يمكن الهيكلي.

الويب صفحات توليد نظام يعتمد .حالي نظام أي بكثير تفوق بكفاءة ،الهيكلي المنطق نفس بي الخاص النشر نظام يتبنى نفس أن متعددة مجالات في وتكرارًا مرارًا يثبت مما ،الأبعاد فائقة الخوارزمية نفس على أيضًا بي الخاص المتطرفة الهيكل هذا أن على يدل مما ،الأنظمة هذه بالفعل يستخدمون شركتي موظفو .مستقر بشكل صالحًا يكون أن يمكن الهيكل المستمر الشخصي تدخلي دون يعمل أن يمكن.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية هي المعتمدة.
Chinesische Originalfassung maßgeblich.

خطوات تتبع ولا ،العالية الحوسبية القدرة ذات السائدة المسارات على تعتمد لا أنها :هي الأنظمة لهذه المشتركة السمة لمرة نجاحات ليست إنها .الهيكلية الدخول نقاط تعديلات خلال من مباشرة صالحة المستهدفة النتيجة وتجعل ،ثابتة حوسبة مختلفة مجالات عبر الهيكلي المنطق لنفس المتكرر الظهور بل ،موضعية حيلاً ولا ،واحدة.

متعددة أنظمة بواسطة بالفعل منها التحقق تم هيكلية طريقة بل ،مصادفة ليست الأبعاد فائقة الخوارزمية أن على يدل هذا. والنشر اللوجستية الخدمات مثل متعددة مجالات عبر مستقر بشكل صالح حوسبي نموذج بل ،"شخصية خدعة" ليست إنها الويب صفحات وتوليد.

تم جديد حوسبي هيكلي كنموذج ،الأبعاد فائقة الخوارزمية فإن ،"جديدًا نموذجًا يشكلان والصلاحية الوجود" لمعيار وفقًا حوسبية بنية بل ،فرضية أقترح لا أنا .الحاضر الوقت في بالفعل صالحة ،متعددة مجالات عبر منهجي بشكل منه التحقق في مستقر بشكل صالح أنه بالفعل أثبتت لقد مفهوم؛ أنه إثبات إلى بحاجة لست .متعددة أنظمة في بالفعل تعمل مختلفة جديدًا نموذجًا بالفعل الهيكل هذا اعتبار يمكن ،العملي تطبيقي نطاق ضمن ،الحاضر الوقت في .مختلفة أنظمة.

بشكل وصالحًا منطقيًا متماسكًا الهيكل يكون عندما :جديد كنموذج الأبعاد فائقة الخوارزمية على الحكم معايير بخصوص نموذج أساس بالفعل يمتلك فإنه ،الحالية الطرق مع للاختزال قابلة غير اختلافات امتلاكه مع ،متعددة أنظمة في مستقر ظل في .النموذج جوهر هو الهيكلي والاختلاف ،الأساس هو التجريبي والدليل ،البداية نقطة هو الذاتي التماسك .جديد على فقط يؤثر به يعترف الخارجي العالم كان إذا ما جديد؛ نموذج بالفعل هو النظام هذا ،"الصلاحية يشكل الوجود" معيار على صلاحيته تعتمد لا ،واقعية أنظمة في بالفعل قائم حوسبي هيكلي نموذج هذا .الصلاحية على وليس ،الانتشار الإجماع أو الخارجي الاعتراف.

أساسي بشكل التقليدية الخوارزميات تعتمد :هو السائدة الحوسبة ونماذج الأبعاد فائقة الخوارزمية بين الجوهري الاختلاف في .جديدة استنتاجية هياكل في رجعي بأثر المشاركة عادةً يمكنها لا ،نتيجة إنتاج بمجرد الأمامي؛ الحساب على يمكن ،معينة ظروف ظل في .الاستخدام لإعادة قابلة هيكلية عقدة بل ،نهاية نقطة النتيجة تعد لم ،الأبعاد فائقة الخوارزمية متعددة بداية نقاط ذات حوسبة شبكة ويشكل المتكرر الحساب من يقلل مما ،جديدة استنتاجية مسارات في المشاركة للنتيجة يمكن ،الأبعاد فائقة الخوارزمية هيكل في نهاية؛ نقطة عادةً النتيجة تكون ،التقليدية الحوسبية الهياكل في .متعددة ومسارات ليست الأبعاد فائقة الخوارزمية .المسارات متعددة استنتاجية شبكة يشكل مما ،جديدة بداية نقطة تصبح أن نفسها للنتيجة ".أصلاً خوارزمية توجد أن يجب كان إذا ما" حول مسبق سؤال طرح إعادة بل ،للخوارزميات ترقية

التاسع الجزء: توضيح المفاهيم الخاطئة الشائعة

دفع لتجنب ،مقدمًا وأوضحها التالية الستة الفهم سوء أتوقع ،مختلفة مجالات من قراء مع الأولية التبادلات على بناءً الانتشار أثناء مناسبة غير أطر إلى قسرًا المفهوم.

الديناميكية البرمجة :توضيح مؤقت؟ تخزين أو ديناميكية برمجة مجرد الأبعاد فائقة الخوارزمية أليست :الأول الفهم سوء بداية نقاط باستنتاج الأبعاد فائقة الخوارزمية تسمح ."متطابقة مدخلات" ظل في النتائج استخدام يعيدان المؤقت والتخزين تحل المشكلة؛ نفس حساب إعادة مشكلة المؤقت التخزين يحل .الجوهري الاختلاف هو هذا — واحدة نتيجة من مختلفة النتائج هياكل خلال من جديدة مشاكل نشر مشكلة الأبعاد فائقة الخوارزمية.

يتعايشا أن يمكن والنظام العشوائية :توضيح متناقضًا؟ هذا أليس — "منتظمة لكن عشوائية" تقول :الثاني الفهم سوء. طريقة بل ،فوضى ليست العشوائية .منتظمة الأنواع وهيكل الإجمالية الكثافة لكن ،عشوائي الغابة في الأشجار توزيع النظام مع تآزري بشكل وتعمل ،وتأسيسية جوهرية هي الأبعاد فائقة الخوارزمية في العشوائية .الهيكل لتنظيم.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة هي.
Chinesische Originalfassung maßgeblich.

الثالث الفهم سوء: توضيح النتيجة؟ صحة من التحقق يمكن كيف ،ومخرجات مدخلات بدون فائقة الخوارزمية ترفض لا نمط في .الخطي والمخرجات المدخلات نمط في تعمل أن يجب لا الحوسبة أن ترى بل والمخرجات؛ المدخلات الأبعاد لا هذا .والمخرجات المدخلات تطابق خلال من وليس ،الهيكلي الاتساق خلال من "الصلاحية" تحديد يتم ،الهيكلي الظهور التقليد عن مختلفًا تحقق إطار يقترح بل ،التحقق عن يتخلى.

الرابع الفهم سوء: توضيح الفوضى؟ نظرية أو التعقيد نظرية مجرد هذه أليست: تصفان الفوضى ونظرية التعقيد نظرية خلال من وتوجيهها فهمها يمكن التي الأنظمة" وصف الأبعاد فائقة الخوارزمية تحاول ."بها التنبؤ يصعب التي الأنظمة" تسأل ؛"صعب التنبؤ" الفوضى نظرية تقول .التنبؤ نمط تغيير بل ،التنبؤ عن التخلي لا — "الهيكلية الدخول نقاط الدخول؟ نقطة تغيير طريق عن ضروري غير التنبؤ جعل يمكن هل" الأبعاد فائقة الخوارزمية"

الخامس الفهم سوء: توضيح أيضًا؟ تعديلاً الخصر ثنية أليست ولكن ،"الوصفية البيانات تعدل لا" تقول: هي الخصر ثنية لا (القماش ،القصّة ،البنطال طول ،الخصر محيط) الوصفية البيانات .الأصلية للمعلمات دائمًا تعديلاً وليست ،مؤقتة حالة تغير ولا ،ومؤقتة ،للعكس قابلة الثنية ."السمات تعديل"و "الحالات تراكب" بين الفرق هو هذا .الإطلاق على تتغير الأساسية السمات.

السادس الفهم سوء: توضيح التقليدية؟ الخوارزميات قيمة الأبعاد فائقة الخوارزمية تنكر هل: كانت .لا بالتأكيد الكمبيوتر أجهزة وجدت لما ،التقليدية الخوارزميات بدون .البشرية التكنولوجيا تاريخ في جدًا مهمة التقليدية الخوارزميات يمكن لا .الفائقة الحوسبة أو ،الهندسية والمحاكاة ،الاصطناعي والذكاء ،الاتصالات وأنظمة ،البيانات وقواعد ،الحديثة كنقطة بها الاعتراف يعني لا التقليدية الخوارزميات بقيمة الاعتراف فإن ،ذلك ومع .التقليدية الخوارزميات قيمة إنكار فائقة الخوارزمية تطرح بينما ،معين حوسبي نموذج داخل الكفاءة مشاكل التقليدية الخوارزميات تحل .للخوارزميات نهاية بحاجة المرء كان إذا عما يسأل والآخر أفضل؛ بشكل المشي بكيفية يتعلق أحدهما .نفسه الحوسبي النموذج مسألة الأبعاد الإطلاق على المسار هذا أخذ إلى.

الجزء العاشر: البشرية الحوسبة وتاريخ الأبعاد فائقة الخوارزمية

موجزة لمحة هنا أقدم ،البشرية الحوسبة تاريخ ضمن أفضل بشكل الأبعاد فائقة الخوارزمية وضع على القراء لمساعدة الكلي المستوى على.

البشري الفهم مر بمراحل "الحوسبة" لـ .متعددة كانت الحوسبة هي الأولى المرحلة كانت: الحوسبة الخوارزميات كانت اليدوية؛ المرحلة كانت .للحوسبة الأساسية القواعد البشر فهم ،العملية هذه خلال من لكن ،للخطأ وعرضة بطيئة ،بشرية كخطوات لكن ،الحوسبة ميكنة تمت لبابيج؛ التحليلي المحرك إلى الحاسبة باسكال آلة من ،الميكانيكية الحوسبة هي الثانية تورينج؛ ونموذج الإلكترونية الحوسبة هي الثالثة المرحلة كانت .المادية الهياكل على تعتمد تزال لا كانت الخوارزميات هي الرابعة المرحلة .الحوسبة حد تورينج آلة أصبحت كبرامج؛ الخوارزميات ترميز تم نيومان؛ فون بنية انتشرت لمعالجة الحوسبة وحدات من كبير عدد خلال من الحوسبية القدرة تتراكم حيث ،الفائقة والحوسبة المتوازية الحوسبة تورينج لنموذج امتدادًا يزال لا الأساسي المنطق لكن ،تعقيدًا أكثر مشاكل.

الخامسة المرحلة أعتاب على البشرية ،حاليًا .الحوسبة تحاول الكلاسيكية؛ البتات قيود كسر الكمومية الحوسبة تحاول الإطار كسر الأبعاد فائقة الخوارزمية وتحاول البيولوجية؛ العصبية الأنظمة هيكل تقليد الشكلية العصبية مدخل" الخطي نفسه "مخرج ← خطوات ←.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

تحل .هرمي تسلسل علاقة بل ،استبدال علاقة ليست التقليدية والخوارزميات الأبعاد فائقة الخوارزمية بين العلاقة يمكن هل" الأبعاد فائقة الخوارزمية تسأل ."معين مسار على أكبر بكفاءة الحساب كيفية" مشكلة التقليدية الخوارزميات فإن ،الخاصة لحدودها مستمر اختراق عملية البشرية الحوسبة تاريخ اعتبرنا إذا "تجاوزه؟ حتى أو ،المسار تعريف إعادة تقترح بل ،القديم الإطار داخل المشاكل تحل لا إنها .العملية هذه في جديدة عقدة بالضبط هي الأبعاد فائقة الخوارزمية جديدًا إطارًا.

يزال لا ،جديد كمفهوم .تمامًا العكس على بل .بالفعل كاملة أو ناضجة الأبعاد فائقة الخوارزمية أن يعني لا الحكم هذا أصل تثبيت بالضبط هو الورقة هذه من الغرض .التكوين قيد تطبيقها وسيناريوهات منها التحقق وطرق وحدودها تعريفها اللاحق للتطوير مستقر نظري أساس وتوفير ،المفهوم هذا.

تعريف إعادة بل ،تحسينًا ليس هذا ـــ الخاتمة :عشر الحادي الجزء

مقابل "أسرع" فرق ولا ،"أسوأ" مقابل "أفضل" فرق ليس التقليدية والخوارزميات الأبعاد فائقة الخوارزمية بين الفرق النموذج مستوى على جوهري فرق بل ،"أبطأ".

لأنني ،عليه ستحصل مخرج أي أعرف وأنا ،المدخل تعطيني أنت .المتوقع التحكم إلى التقليدية الخوارزميات تسعى تحقق ،اختبار ،بناء ،تصميم :المهندس نموذج إنه .خطوة كل حسبت.

العام النمط أن أعرف لكنني ،وسيطة حالة بكل بدقة التنبؤ أستطيع لا .للفهم القابل النشوء الأبعاد فائقة الخوارزمية تصف الأنظمة منظر أو البيئة عالم بنموذج أشبه إنه ."منظور" وله حي البيانات من جزء كل .ما منظمة بطريقة نفسه سيقدم الناشئ النظام استخدم ،وجّه ،افهم ،راقب :المعقدة.

المستهدفة النتيجة الأبعاد فائقة الخوارزمية تجعل .معين مسار طول على تدريجيًا النتيجة التقليدية الخوارزميات تعدل بأكمله المسار تجاوز وبالتالي ،الهيكلية الدخول نقطة تغيير طريق عن فورًا صالحة.

نتائج لحمل البيانات نفس استخدام" هي الأبعاد فائقة الخوارزمية ."مختلفة لنتائج البيانات تعالج" التقليدية الخوارزميات متعددة محتملة".

بشكل احسب" إلى الأبعاد فائقة الخوارزمية تسعى ."واحدة مرة صحيح بشكل احسب" إلى التقليدية الخوارزميات تسعى أخرى مرة الحساب إلى أبدًا تحتاج لا ثم ،واحدة مرة صحيح".

أصلاً خوارزمية توجد أن يجب كان إذا ما" حول مسبق سؤال طرح إعادة بل ،للخوارزميات تحسينًا ليس هذا".

بشكل بالركض الأمر يتعلق لا .للخوارزمية تعريف إعادة بل ،للخوارزمية تحسينًا ليست الأبعاد فائقة الخوارزمية ،لذلك إذا ما حتى أو ،التقليدي المسار خارج آخر مسار هناك كان إذا عما بالسؤال بل ،التقليدية الخوارزميات مسار على أسرع بالإشارة بل ،مجدية غير التقليدية الخوارزميات أن إثبات بمحاولة الأمر يتعلق لا .نفسه المسار عن الاستغناء يمكن كان وقت في النتائج على تحصل كيف" تسأل لا .للخوارزمية الأبعاد منخفض شكل مجرد هي التقليدية الخوارزميات أن إلى تسأل بل ،"البيانات من المزيد تعالج كيف" تسأل لا ."الهيكل خلال من مباشرة نتيجة إنشاء يمكن هل" تسأل بل ،"أقل المحتملة النتائج من المزيد تحمل أن البيانات لنفس يمكن هل".

أو ،للتحقق قابلة غير أو ،للحساب قابلة غير أنها على الأبعاد فائقة الخوارزمية إلى يُنظر قد ،السائد الحوسبة نموذج ضمن جديدًا مفهومًا أن حقيقة .النموذجي اختلافها بالضبط يشكل هذا لكن .الحالية الحوسبة نظرية حدود ضمن دمجها يصعب

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة هي.
Chinesische Originalfassung maßgeblich.

يمكنه لا نفسه القديم النموذج أن أيضًا تعني فقد له؛ معنى لا أنه بالضرورة تعني لا قديم نموذج داخل مستقر غير يبدو لرؤية أصلي وتعريف ،هيكلي اقتراح الأول المقام في هي الأبعاد فائقة الخوارزمية ،الحاضر الوقت في .بالكامل استيعابه الإضافات خلال من لاحق استمرار إلى تحتاج إنها .مكتملة هندسية بحلقة ناضجًا نظامًا وليست ،جديدة حوسبية .المفاهيمي أصلها توضيح أولاً يجب ولكن ،والتطبيقات والتحققات والتطويرات

مجرد بالضرورة ليست الخوارزمية :للحوسبة جديدة رؤية هو الأبعاد فائقة الخوارزمية إليه تشير ما ،النهاية في معالجة عملية مجرد بالضرورة وليست ،برنامج مجرد بالضرورة وليست ،خطوات مجرد بالضرورة وليست ،رياضيات من هيكلية وشبكة ،الأبعاد متعددة العلاقات لتنظيم طريقة تكون أن أيضًا للخوارزمية يمكن .والمخرج المدخل بين من تنتقل أن يمكن والعشوائية؛ النظام على تحتوي أن يمكن .والنتائج ،والحالات ،والعلاقات ،الدخول ونقاط ،البيانات البيانات تعديل دون متعددة حالات مع تتكيف أن يمكن النتائج؛ من رجعي بأثر أسبابًا تولد أو ،النتيجة إلى السبب .النتيجة نفس في تتقارب متعددة بداية نقاط تجعل أو ،جديدة بداية نقطة واحدة نتيجة تجعل أن يمكن الوصفية؛

،الحوسبة تقليل يمكنها الأبعاد متعددة تنظيم طريقة بل ،تعقيدًا أكثر حوسبة بالضرورة ليست حقًا المتقدمة الخوارزمية .دورات بتشكيل والنتيجة للسبب والسماح ،دخول نقاط إلى النتائج وتحويل ،الهياكل وإحياء

،جديدًا مفهومًا بل ،عاديًا جديدًا مصطلحًا ليست إنها ."الأبعاد فائقة الخوارزمية" لاقتراحي الأساسي المعنى هو هذا المتكررة الحوسبة على تركيزها ينصب لا .متعددة واقعية أنظمة في بالفعل قائمًا جديدًا حوسبة وهيكل ،جديدًا ونموذجًا ليس الدخول؛ نقاط هيكلة إعادة على بل ،الحوسبية القدرة تجميع على ليس المتكررة؛ الحوسبة تقليل على بل ،أسرع بشكل النتائج ترك على ليس العلاقات؛ تغيير مع تغيير دون الوصفية البيانات إبقاء على بل ،باستمرار البيانات تعديل على طول على نتائج عن التقليدية الخوارزميات تبحث .جديدة بداية نقاط تصبح النتائج جعل على بل ،النهاية نقاط عند تتوقف تتساءل الحوسبة؛ إكمال إلى التقليدية الخوارزميات تسعى .الهياكل داخل النتائج الأبعاد فائقة الخوارزمية تفعّل المسارات؛ بينها جوهرية الأكثر الاختلاف هو هذا .التقليدي بشكلها الحوسبة توجد أن يجب كان إذا عما الأبعاد فائقة الخوارزمية .للخوارزمية الحالي التعريف وبين

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

الملحق: حدود هذه الورقة والأسئلة المفتوحة

لا .لاحقًا التطوير تنتظر التالية الأسئلة .كاملاً شكليًا تعريفًا وليس ،الأبعاد فائقة للخوارزمية أوليًا إطارًا الورقة هذه تقترح التالية للخطوات الاتجاهات بالضبط هي بل ،الورقة هذه في الواردة للتعريفات نفيًا تشكل.

في ؟"صالحة" النتيجة كانت إذا ما نحدد كيف الأبعاد؟ فائقة الخوارزمية "إكمال" علامة هي ما .التقارب شروط ،أولاً قد ،الأبعاد فائقة الخوارزمية في .والمخرجات المدخلات تطابق خلال من الإجابة تعريف يتم ،التقليدية الخوارزميات .بعد يكتمل لم التعريف هذا .الهيكلي الاتساق خلال من تعريفها إلى الإجابة تحتاج

المقاييس تزال لا هل محدودة؟ موارد ضمن الأبعاد متعددة المتراكبة الحالات تمثيل يمكن كيف .الموارد تمثيل ،ثانيًا تكن لم إذا تعريفها؟ إعادة يتم كيف ،كذلك كانت إذا للتطبيق؟ قابلة الحوسبية والقدرة والمساحة الوقت مثل للموارد التقليدية .الاستكشاف تنتظر الأسئلة هذه محلها؟ تحل التي المقاييس هي ما ،كذلك

العشوائية بين الفاصل الحد هو أين ؟"منتظمة لكن عشوائية" في "النظام" ضمان يتم قواعد بأي .النظام ضمان ،ثالثًا .إضافيًا بحثًا تتطلب الأسئلة هذه العشوائية؟ النظام يقمع ظروف أي في النظام؟ العشوائية تعطل ظروف أي في والنظام؟

المزايا تقييم أو اختيار يتم كيف ،نتيجة من بالعكس متعددة بداية نقاط استنتاج عند .العكسي الاستدلال في الانتقاء ،رابعًا بمقياس علاقته هي ما ،موجودًا كان إذا ؟"العكسي الاستدلال كفاءة" لـ ما مقياس هناك هل المختلفة؟ البداية لنقاط النسبية الأمامي؟ الحساب كفاءة

تورينج؟ آلة على القائمة الحالية الأنظمة مع الأبعاد فائقة الخوارزمية تعمل كيف .التقليدية الأنظمة مع الواجهات ،خامسًا الأبعاد فائقة الخوارزمية بين الحدود تحديد يتم كيف ،العملية الهندسة في هرمية؟ طبقة أم ،مكمل أم ،استبدال هي هل هجينة؟ أوضاع توجد هل التقليدية؟ والخوارزميات

خلال من النتيجة على الحصول يتم لم إذا الأبعاد؟ فائقة الخوارزمية نتائج من التحقق يتم كيف .التحقق إطار ،سادسًا تمامًا؟ مختلفة تحقق فلسفة إلى حاجة هناك هل الاختبار؟ وقابلية التكرار قابلية إرساء يتم كيف ،خطية خطوات

نموذج أي اقتراح 。"تثبيت" كورقة الورقة لهذه الحتمية الخصائص هي بل ،الورقة هذه في عيوبًا ليست الأسئلة هذه .خاتمة وليست ،بداية هي الورقة هذه .مفتوحة بأسئلة مصحوبًا يأتي جديد

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية هي المعتمدة.
Chinesische Originalfassung maßgeblich.

ببليوغرافية ملاحظة

من ،ذلك ومع .قائمة أكاديمية فكرية مدرسة لأي مباشرًا امتدادًا ليست الورقة هذه في المقترحة الأبعاد فائقة الخوارزمية بالمعنى اقتباسات وليست ،القراء لتوجيه فقط تُقدم ،الورقة لهذه حوارية خلفية التالية المجالات توفر ،الأفكار حيث التقليدي الأكاديمي.

المسائل تعمل .التقليدية الخوارزميات الورقة هذه بها تقارن مرجعية كنقطة للحوسبة القابلية ونظرية تورينج آلة تعمل هذه في "والسبب التأثير نظرية" ترتبط ؛"النتائج من الأسباب استدلال" لـ قائمة كمحاولات البيزي والاستدلال العكسية ؛"نتائج تظهر التي الهياكل" لـ قائمة كأوصاف المعقدة والأنظمة النشوء نظرية تعمل .الاتجاه في تختلف لكنها بها الورقة بيانات وقواعد المعرفة رسومات تعمل ."للتوجيه قابلًا"و "للفهم قابلًا" النشوء جعل الأبعاد فائقة الخوارزمية تحاول تعطي ،الأساس هذا على ،الأبعاد فائقة الخوارزمية ؛"متعددة عقد عند البيانات تقاطع" لـ قائمة كممارسات البيانية الرسوم والحوسبة ،الكمومية الحوسبة ذلك في بما ،الكلاسيكية غير الحوسبة نماذج تعمل ."المتغيرة الدخول نقاط" لبعد أولوية الخوارزمية تركز بينما ،الكلاسيكية الهندسة أو الكلاسيكية البتات كسر على ،إلخ ،الشكلية العصبية والحوسبة ،التماثلية نفسه "مخرج ← خطوات ← مدخل" الخطي الإطار كسر على أكثر الأبعاد فائقة.

هذه من الهدف .تفنيدًا أو وراثة وليست ،وتسامي حوار علاقة هي أعلاه المذكورة والمجالات الورقة هذه بين العلاقة فوقها التساؤل من جديد مستوى رفع بل ،المجالات هذه داخل تدريجية تحسينات إجراء ليس الورقة.

صلة ذات مقالات

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [النشر] ليس لدي خوارزمية، الخوارزميات على أتفوق ذلك ومع!

http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [المتطرفة الفلسفة] والسبب التأثير نظرية

https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [المتطرف النشر] الأسر رفض الخوارزمي

https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Indexed and catalogued in WorldCat (OCLC); permanently archived in TROVE (National Library of Australia)
Permanent archival record (DataCite Concept DOI): https://doi.org/10.5281/zenodo.17646306

[Extremalgorithmus] Hyperdimensionaler Algorithmus

Eine Neudefinition von "Berechnung" selbst

Autor: Wu Zhaohui JEFFI CHAO HUI WU

Zusammenfassung

Dieses Paper stellt das neue Konzept des "hyperdimensionalen Algorithmus" als eine strukturelle Neuprüfung der Grenzen der traditionellen Algorithmusdefinition vor. Traditionelle Algorithmen werden typischerweise auf dem linearen Rahmen von Eingabe, Schritten, Regeln und Ausgabe aufgebaut, wobei Endlichkeit, Determinismus, Effektivität, Durchführbarkeit und klare Eingabe-Ausgabe-Grenzen betont werden. Obwohl moderne Supercomputer über eine extrem hohe Rechenleistung verfügen, besteht ihre zugrundeliegende Logik immer noch hauptsächlich in der Ausführung etablierter Algorithmenparadigmen in größerem Maßstab. Dieses Paper argumentiert, dass die wirklich diskussionswürdige "Superberechnung" nicht nur das Wachstum des Rechenmaßstabs ist, sondern eine grundlegende Änderung des Berechnungsparadigmas selbst. Der hyperdimensionale Algorithmus ist keine beschleunigte Version traditioneller Algorithmen, keine Erweiterung mathematischer Algorithmen; er ist eine mehrdimensionale, überlagerte, zufällige und dennoch geordnete strukturelle Sichtweise der Berechnung. In dieser Struktur sind Daten nicht länger nur eine passive Eingabe oder ein endgültiger Ausgang, sondern ein Knotenattribut, das gleichzeitig mehrere Startpunkte und mehrere Ergebnisse besitzt; ein Ergebnis ist nicht länger nur ein Endpunkt, sondern kann ein neuer Eintrittspunkt werden; Metadaten müssen nicht wiederholt modifiziert werden, sondern können durch strukturelle Eintrittspunkte, Beziehungsänderungen und bedingte Aktivierung verschiedenen Zuständen und Ergebnissen entsprechen. Dieses Paper verwendet die "Wirkung-Ursache-Theorie"-Philosophie und das alltägliche Beispiel des "Faltens der Taille einer Hose", um zu veranschaulichen, wie der hyperdimensionale Algorithmus durch Änderung des strukturellen Eintrittspunkts ein Zielergebnis direkt gültig machen kann, wodurch wiederholte Berechnungen reduziert und sogar der ursprüngliche Berechnungspfad umgangen werden kann. Dieses Paper zielt darauf ab, eine vorläufige Definition, strukturelle Grenzen und gedankliche Grundlagen für den "hyperdimensionalen Algorithmus" zu etablieren und einen originären theoretischen Knoten für die zukünftige Entfaltung hyperdimensionaler Berechnung, Extremalgorithmen und eines neuen Wissenschaftssystems bereitzustellen. Nach dem Kriterium "Existenz und Gültigkeit konstituieren ein neues Paradigma" ist der

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

hyperdimensionale Algorithmus als ein bereits in mehreren Bereichen systematisch validiertes neues strukturelles Berechnungsparadigma gegenwärtig bereits gültig.

Schlüsselwörter: Hyperdimensionaler Algorithmus; Extremalgorithmus; Hyperdimensionales Rechnen; Wirkung-Ursache-Theorie; Metadaten; Berechnungsparadigma; Struktureller Eintrittspunkt; Nichtlineare Kausalität; Neue Wissenschaft

Einleitung: Warum wir "Algorithmus" neu diskutieren müssen

Wenn ich den "hyperdimensionalen Algorithmus" vorschlage, geschieht dies weder, um bestehenden Algorithmen einen größeren Namen zu geben, noch um traditionelle Algorithmen in ein neuartiges Konzept zu verpacken. Ganz im Gegenteil: Ich möchte eine grundlegendere Frage diskutieren: Ist das gegenwärtige menschliche Verständnis von "Algorithmus" bereits in einem zu engen Rahmen gefangen?

Wenn Menschen heute über Algorithmen sprechen, denken sie normalerweise an mathematische Formeln, Programmcode, logische Schritte, Eingabe-Ausgabe, Modelltraining, Datenverarbeitung, Pfadoptimierung, Suchranking, automatische Empfehlung und künstliche Intelligenzschlussfolgerung. Dies sind sicherlich wichtige Formen von Algorithmen, und sie stützen den Betrieb moderner Computer, des Internets, von Datenbanken, künstlicher Intelligenz, Supercomputing, Industriesystemen und der Informationsgesellschaft. Wenn jedoch ein Algorithmus nur als "eine Menge geordneter Schritte" verstanden wird, nur als ein Prozess, der "mit einer Eingabe beginnt, eine regelbasierte Verarbeitung durchläuft und schließlich eine Ausgabe erzeugt", dann schränkt dieses Verständnis die Berechnung bereits in einem niedrigdimensionalen Rahmen ein.

Wir leben in einer von Algorithmen beherrschten Ära. Von Suchmaschinen über Empfehlungssysteme bis hin zu Wettervorhersage und künstlicher Intelligenz – die Grenzen der Algorithmen sind die Grenzen der menschlichen Intelligenz. Aber es gibt eine Frage, die selten gestellt wird: Können Algorithmen nur so sein? Warum muss Berechnung dem linearen Muster "Eingabe → Schritte → Ausgabe" folgen? Warum muss dasselbe Problem jedes Mal von Grund auf neu berechnet werden? Dieses Paper versucht, diese stillschweigenden Annahmen herauszufordern und ein völlig anderes Berechnungsparadigma vorzuschlagen – den hyperdimensionalen Algorithmus. Nach dem Kriterium "Existenz und Gültigkeit konstituieren ein neues Paradigma" ist der hyperdimensionale Algorithmus als ein bereits in mehreren Bereichen systematisch validiertes neues strukturelles Berechnungsparadigma gegenwärtig bereits gültig.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Es ist wichtig anzumerken, dass der in diesem Paper vorgeschlagene "hyperdimensionale Algorithmus" ein völlig anderes Konzept ist als das "hyperdimensionale Rechnen" (Hyperdimensional Computing, HDC), das in der akademischen Welt seit fast dreißig Jahren existiert. HDC verwendet zufällige binäre Vektoren extrem hoher Dimensionen zur Kodierung und Operation, strebt eine effizientere Darstellung und Berechnung an, verbleibt aber dennoch innerhalb des linearen Paradigmas "Eingabe → Verarbeitung → Ausgabe". Der hyperdimensionale Algorithmus ist völlig anders: Er hinterfragt, ob ein Ergebnis direkt durch Struktur etabliert werden kann, ob der Berechnungspfad selbst umgangen werden kann, ob Metadaten unverändert bleiben können, während sie sich an mehrere Ergebnisse anpassen. Ihre Namen sind ähnlich, aber ihre Inhalte sind gegensätzlich. Der hyperdimensionale Algorithmus ist kein Upgrade von Algorithmen, sondern ein erneutes Hinterfragen der Prämisse, "ob ein Algorithmus überhaupt existieren muss".

Teil Eins: Die aktuelle Standarddefinition von "Algorithmus"

In den 主流 Systemen der Informatik, Mathematik und Ingenieurwissenschaften gibt es eine lange akzeptierte Grunddefinition eines Algorithmus: Ein Algorithmus ist ein wohldefinierter Berechnungsprozess, der aus einer endlichen Anzahl von Schritten besteht, eine oder mehrere Eingaben annimmt, sie durch deterministische Regeln transformiert und innerhalb einer endlichen Zeit eine oder mehrere Ausgaben erzeugt. Dieses Verständnis ist nicht die beiläufige Meinung einer Person, sondern ein grundlegender Konsens, der sich über die lange Entwicklung der modernen Berechnungszivilisation gebildet hat. Es stützt die zugrundeliegende Logik von Software, Hardware, Datenbanken, Netzwerksystemen, Modellen der künstlichen Intelligenz und Supercomputing-Zentren. So komplex ein Algorithmus auch erscheinen mag, sein Kern dreht sich immer noch um Eingabe, Regeln, Schritte und Ausgabe.

Dieses standardmäßige Algorithmuskonzept kann in mehrere Kernmerkmale zerlegt werden.

Erstens, Endlichkeit. Ein Algorithmus muss nach einer endlichen Anzahl von Schritten terminieren; er darf nicht endlos schleifen oder für immer laufen. Ein Prozess, der niemals anhält, selbst wenn er formal beschrieben werden kann, kann kaum als effektiver Algorithmus betrachtet werden. Alle wissenschaftlichen Berechnungsprogramme und Datenverarbeitungsabläufe haben klare Abbruchbedingungen.

Zweitens, Determinismus. Bei derselben Eingabe unter denselben Bedingungen sollte der Algorithmus dieselbe Ausgabe produzieren. Selbst wenn einige Algorithmen

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Zufälligkeit einführen, handelt es sich typischerweise um kontrollierte Zufälligkeit, reproduzierbare Zufälligkeit oder einen statistisch interpretierbaren probabilistischen Mechanismus, nicht um eine völlig unkontrollierbare intrinsische Zufälligkeit. Die Ergebnisse numerischer Simulationen müssen reproduzierbar sein; dies ist eine grundlegende Anforderung des wissenschaftlichen Rechnens.

Drittens, klare Eingabe-Ausgabe-Grenzen. Die Eingabe zuerst, die Verarbeitung in der Mitte, die Ausgabe zuletzt; Startpunkt und Endpunkt haben klar definierte strukturelle Grenzen. Welche Daten Sie dem Algorithmus auch geben, nach der Verarbeitung erhalten Sie ein Ergebnis; diese Grenze ist eindeutig. Eingabe zuerst, Verarbeitung danach, Ausgabe zuletzt; die Reihenfolge kann nicht umgekehrt werden.

Viertens, Effektivität. Jeder Schritt eines Algorithmus muss eine praktisch ausführbare Operation sein; er darf keine Sprünge enthalten, die unmöglich auszuführen, zu beschreiben oder zu verifizieren sind. Jeder Schritt muss eine grundlegende Operation sein, die ein Mensch mit Papier und Bleistift ausführen könnte oder die ein Computer tatsächlich ausführen könnte – Addition, Subtraktion, Multiplikation, Division, logische Urteile, Datenlesen/-schreiben usw.

Fünftens, Durchführbarkeit. Selbst wenn ein Algorithmus theoretisch gültig ist, wenn die Zeit, Rechenleistung, der Speicher oder die Speicherressourcen, die er verbraucht, völlig inakzeptabel sind, ist es schwierig, ihn in technischer Hinsicht als effektiven Algorithmus zu betrachten. Ein theoretisch korrekter Algorithmus, der jedoch Hunderte von Millionen Jahren zur Ausführung benötigen würde, wird in der Technik nicht als durchführbarer Algorithmus angesehen.

Dieses gesamte Set von Algorithmuskonzepten lässt sich letztlich auf das Turingmaschinenmodell zurückführen. Als Kernabstraktion der modernen Berechnungstheorie definiert die Turingmaschine die grundlegende Grenze der "Berechenbarkeit". So leistungsfähig die heutigen Computer auch sind, so fortschrittlich die Chips, so groß der Maßstab der Supercomputing-Zentren – ihre zugrundeliegende Logik ist nie wirklich von diesem Pfad abgewichen: Eingabe, Regeln, Schritte, Ausgabe. Moderne Supercomputer beschleunigen lediglich die Ausführung dieses Prozesses durch größere Hardware-Maßstäbe, höhere Parallelität und komplexere Systemplanung. Das heißt, im traditionellen Kontext ist "Supercomputing" in erster Linie eine Ausweitung des Berechnungsmaßstabs, nicht unbedingt eine grundlegende Änderung des Berechnungsparadigmas. Die Rechenleistung kann um das Tausend- oder Zehntausendfache steigen, aber wenn die zugrundeliegende Logik "Eingabe, Schritte, Ausgabe" bleibt, verbleibt sie immer noch innerhalb des traditionellen Algorithmusparadigmas.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Teil Zwei: Definition des "hyperdimensionalen Algorithmus"

Der "hyperdimensionale Algorithmus", den ich vorschlage, entsteht genau aus diesem Kontext. Er ist keine verbesserte Version traditioneller Algorithmen, kein schnellerer Algorithmus, keine komplexere mathematische Formel und keine fortgeschrittenere Programmiertechnik. Es ist ein völlig anderes strukturelles Verständnis.

Zuerst müssen wir die Bedeutung des Begriffs "hyperdimensional" erklären. "Hyperdimensional" hat hier zwei Bedeutungen. Die erste Bedeutung ist, dass er sich nicht auf eine eindimensionale lineare Abfolge von Schritten beschränkt, sondern mehrere Dimensionen gleichzeitig sich entfalten lässt. Die zweite Bedeutung ist, dass er versucht, die traditionellen definitorischen Grenzen der "Berechnung" zu überschreiten – ein Algorithmus nicht länger notwendigerweise mathematisch, sequenziell oder deterministisch sein muss. Ein traditioneller Algorithmus ist wie das Fahren auf einer Einbahnstraße: Sie müssen von Punkt A starten, über B, C, D gehen, um Z zu erreichen. Der hyperdimensionale Algorithmus glaubt nicht, dass Berechnung eine Einbahnstraße sein muss. Er glaubt, dass Berechnung ein Netzwerk, ein Feld, eine mehrdimensionale Struktur sein kann, in der jeder Knoten ein Eintrittspunkt sein kann und jeder Knoten ein Austrittspunkt sein kann.

In meinem neuen Wissenschaftssystem ist ein Algorithmus nicht notwendigerweise ein mathematischer Algorithmus, nicht notwendigerweise eine einzelne geordnete Berechnung, nicht notwendigerweise dazu verpflichtet, entlang fester Schritte zu laufen, und nicht notwendigerweise strikt dem linearen Muster "Eingabe, Verarbeitung, Ausgabe" zu gehorchen. Der hyperdimensionale Algorithmus ist eine mehrdimensionale, überlagerte, zufällige und dennoch geordnete Struktur. In dieser Struktur ist jedes Datum weder eine isolierte Eingabe noch ein fester Ausgang, sondern besitzt gleichzeitig mehrere Rollen: Es kann sowohl der Startpunkt für mehrere Ergebnisse als auch das Ergebnis, das durch das gemeinsame Wirken mehrerer Startpunkte entsteht, sein.

Mit anderen Worten: Traditionelle Algorithmen platzieren Daten innerhalb eines Prozessflusses, während der hyperdimensionale Algorithmus Daten innerhalb einer Struktur platziert. In traditionellen Algorithmen werden Daten normalerweise in Eingabedaten, Zwischendaten und Ausgabedaten unterteilt. Jede Datenart hat eine klare Position und Rolle. Eingabe ist Eingabe, Ergebnis ist Ergebnis, Zwischenprozess ist Zwischenprozess. Im hyperdimensionalen Algorithmus jedoch hat ein Datum nicht nur eine Identität. Es könnte in einer Richtung ein Ergebnis und in einer anderen Richtung ein Startpunkt sein; es könnte in einer Struktur ein aufgerufenes Objekt sein und in einer anderen Struktur zu einem Eintrittspunkt werden, der neue Ergebnisse auslöst.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Daten sind nicht länger nur passives Material, sondern ein mehrdimensionaler Beziehungsknoten.

Dies entspricht genau dem Kern des hyperdimensionalen Algorithmus: Jedes Datum ist der Startpunkt für mehrere Ergebnisse und auch das Ergebnis mehrerer Startpunkte. Traditionelle Algorithmen verwenden "geordnete Schritte", um ein einziges Ergebnis zu erzeugen; der hyperdimensionale Algorithmus ist eher eine mehrdimensionale Zustandsstruktur, in der, in der Überlagerung von Zufälligkeit und Ordnung, das Ergebnis nicht hervorgerechnet wird, sondern sich innerhalb der Struktur natürlich manifestiert. In diesem System ist das Datum sowohl der Startpunkt als auch ein Bestandteil des Ergebnisses.

Um die Kernmerkmale des hyperdimensionalen Algorithmus klarer darzustellen, zerlege ich sie in die folgenden fünf Aspekte.

Erstens, nicht-mathematische Natur. Mainstream-Algorithmen sind im Wesentlichen mathematisch – sie können vollständig durch symbolische Logik und mathematische Formeln beschrieben werden. Der hyperdimensionale Algorithmus ist nicht notwendigerweise mathematisch. Er könnte auf physikalischen Prinzipien, geometrischen Beziehungen, einem neuen Paradigma der Informationstheorie oder sogar Prinzipien des Bewusstseins basieren. Berechnung muss nicht notwendigerweise durch "mathematische Operationen" erfolgen; sie könnte sich natürlich durch geometrische Beziehungen in einem höherdimensionalen Raum darstellen. Zum Beispiel wird die Form einer Seifenblase durch das Prinzip der Minimierung der Oberflächenspannung bestimmt. Dies ist kein "mathematischer Algorithmus", der rechnet – es ist die Physik selbst, die "rechnet". Der hyperdimensionale Algorithmus versucht, diese Art von "Berechnung" zu verstehen, anstatt sie mit mathematischen Formeln zu simulieren.

Zweitens, Mehrdimensionalität und Überlagerung. Traditionelle Algorithmen führen geordnete Schritte in einer einzigen Dimension aus, mit nur einem Zustand pro Schritt. Der hyperdimensionale Algorithmus erlaubt mehreren Dimensionen, gleichzeitig zu existieren, und mehreren Zuständen, sich zu überlagern und zu koexistieren. Der Algorithmus folgt nicht einem einzigen Pfad; er entfaltet sich stattdessen in einem mehrdimensionalen Zustandsfeld. Das Ergebnis wird nicht "berechnet", sondern "manifestiert" sich aus diesem Feld. Zum Beispiel: Wenn sich eine Schneeflocke in der Luft bildet, gibt es keinen "Algorithmus", der jedem Wassermolekül sagt, wohin es gehen soll. Die Anordnung der Wassermoleküle ist das Ergebnis mehrerer Dimensionen – Temperatur, Feuchtigkeit, Luftströmung – die zusammenwirken. Die Form der Schneeflocke "manifestiert" sich, anstatt "berechnet" zu werden.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Drittens, Zufälligkeit in Ordnung. In traditionellen Algorithmen ist Zufälligkeit instrumental, extern, kontrollierbare Störung; der Algorithmus selbst ist deterministisch. Im hyperdimensionalen Algorithmus ist Zufälligkeit intrinsisch und strukturell. Zufälligkeit und Ordnung koexistieren und arbeiten zusammen, konstituieren gemeinsam das Wesen des Algorithmus. Dies ist kein "Algorithmus mit Zufälligkeit", sondern "Zufälligkeit selbst ist ein organischer Bestandteil des Algorithmus". Zum Beispiel: die ökologische Struktur eines Waldes – die Verteilung der Bäume erscheint zufällig, aber insgesamt weist sie eine geordnete Dichteverteilung und Artenbeziehungen auf. Diese "Zufälligkeit" ist kein Fehler, sondern eine Voraussetzung für das normale Funktionieren der ökologischen Struktur.

Viertens, die mehrfachen Rollen von Daten. In traditionellen Algorithmen ist die Rolle von Daten singular – Eingabe ist Eingabe, Ausgabe ist Ausgabe, Zwischenergebnis ist Zwischenergebnis. Im hyperdimensionalen Algorithmus ist jedes Datum gleichzeitig der Startpunkt für mehrere Ergebnisse und das Ergebnis mehrerer Startpunkte. Ein Datenknoten kann sich flussabwärts in mehrere Ergebniswege entfalten und kann auch flussaufwärts von mehreren Ursachen konvergiert werden. Daten sind kein Punkt in einem linearen Fluss mehr, sondern ein Kreuzungsknoten in einem Netzwerk. Betrachten Sie zum Beispiel die Körpergröße einer Person. In einem traditionellen Algorithmus ist die Körpergröße eine "Eingabe" – Sie geben sie ein, um Ausgaben wie Gewichtsvorhersage, Kleidergröße usw. zu erhalten. Aber die Körpergröße ist auch ein "Ergebnis" – sie wird gemeinsam durch mehrere "Startpunkte" wie Genetik, Ernährung und Bewegung bestimmt. Im hyperdimensionalen Algorithmus ist der Datenpunkt der Körpergröße gleichzeitig ein Startpunkt und ein Ergebnis, keine Entweder-Oder-Wahl.

Fünftens, Metadaten unverändert, Beziehungen variabel. Wenn sie mit verschiedenen Problemen konfrontiert werden, modifizieren traditionelle Algorithmen entweder die Daten selbst oder berechnen alles neu. Der hyperdimensionale Algorithmus modifiziert keine Metadaten – die wesentlichen Attribute der Daten bleiben unverändert. Was sich ändert, sind die Eintrittspunkte, die Interpretationsmodi und die Beziehungen zwischen den Daten. Dieselbe Metadatenstruktur, aktiviert durch verschiedene Eintrittspunkte, kann völlig unterschiedliche Ergebnisse präsentieren. Dies bedeutet: Einmal berechnen, unendlich oft verwenden. Betrachten Sie zum Beispiel denselben Textabschnitt. Sie können ihn als Poesie lesen, als Chiffre entschlüsseln oder als historisches Dokument analysieren. Der Text selbst hat sich nicht geändert – Sie haben nur den "Eintrittspunkt" geändert. Der hyperdimensionale Algorithmus strebt genau diese Fähigkeit an: "gleiche Daten, mehrere Eintrittspunkte, mehrere Ergebnisse."

Aus der Perspektive der Datenstruktur: Metadaten selbst werden nicht modifiziert; vielmehr werden sie durch verschiedene Eintrittspunkte und Bedingungen mehreren

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Startpunkten und Ergebnissen innerhalb derselben Struktur zugeordnet. Daten werden nicht mehr wiederholt verarbeitet, sondern, während ihre Struktur unverändert bleibt, durch verschiedene Eintrittspunkte aktiviert, wodurch sie verschiedene Ergebnisse präsentieren. Der traditionelle Ansatz ist "Daten für verschiedene Ergebnisse verarbeiten", während der hyperdimensionale Algorithmus "dieselben Daten verwenden, um mehrere potenzielle Ergebnisse zu tragen" ist.

Teil Drei: Grundlegende Unterschiede zwischen dem hyperdimensionalen Algorithmus und traditionellen Algorithmen

Basierend auf den obigen Definitionen zeigen der hyperdimensionale Algorithmus und traditionelle Algorithmen grundlegende Unterschiede in mehreren Kerndimensionen. Diese werden im Folgenden einzeln ausgeführt.

In Bezug auf die wesentliche Natur sind traditionelle Algorithmen mathematisch und formal, vollständig beschreibbar durch symbolische Logik und mathematische Formeln; sie sind im Wesentlichen mathematische Objekte. Der hyperdimensionale Algorithmus ist nicht notwendigerweise mathematisch; er kann auf physikalischen, geometrischen, informationellen oder sogar bewussten Prinzipien basieren. Er ist keine Teilmenge der Mathematik, sondern eine breitere Kategorie. Dieser Unterschied kann zusammengefasst werden als: vom "mathematischen Objekt" zum "universellen Gesetz".

In Bezug auf die zeitliche Ordnung folgen traditionelle Algorithmen einer singulären, geordneten, linearen zeitlichen Ordnung, Schritt A zu B zu C, streng sequenziell oder zerlegbar in parallele geordnete Unterschritte; der Zeitpfeil ist irreversibel. Der hyperdimensionale Algorithmus hat keine feste Schrittsequenz; er ist mehrdimensional, überlagert, zufällig und dennoch geordnet. Das Ergebnis könnte mit der "Reihenfolge" der Ausführung nicht zusammenhängen, weil es keine traditionelle "Reihenfolge" gibt. Dieser Unterschied kann zusammengefasst werden als: von der "linearen Zeit" zur "höherdimensionalen Gleichzeitigkeit".

In Bezug auf die Kausalität folgen traditionelle Algorithmen einer deterministischen Kausalkette: bei gleicher Eingabe und gleichem Anfangszustand ist die Ausgabe immer eindeutig; Ursache geht Wirkung voraus, ein irreversibler Einwegpfeil. Der hyperdimensionale Algorithmus folgt einer überlagerten Kausalität: Jedes Datum ist der Startpunkt für mehrere Ergebnisse und das Ergebnis mehrerer Startpunkte. Dies ist meine "Wirkung-Ursache-Theorie"-Philosophie – es ist möglich, zuerst die Wirkung zu

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

haben, dann die Ursache. Die Wirkung ist kein Endpunkt, sondern kann ein neuer Startpunkt werden. Dieser Unterschied kann zusammengefasst werden als: von "einziger Ursache, einziger Wirkung" zum "vollständig verbundenen Kausalnetzwerk".

In Bezug auf die Datenstruktur folgen traditionelle Algorithmen einer Einwegflussstruktur von der Eingabe über die Verarbeitung zur Ausgabe; Daten haben eine singuläre Rolle mit klaren Grenzen; Eingabedaten bleiben von Anfang bis Ende Eingaben, Ergebnisdaten bleiben Ergebnisse. Im hyperdimensionalen Algorithmus haben Daten mehrere Rollen; dieselben Daten sind gleichzeitig Ergebnis und Startpunkt; es gibt keine absoluten Startpunkte und keine absoluten Endpunkte, sondern nur Knoten in einem Netzwerk. Dieser Unterschied kann zusammengefasst werden als: von der "Einwegfluidität" zur "rekursiven Symbiose".

In Bezug auf den Berechnungsmodus berechnen traditionelle Algorithmen jedes Mal von Grund auf, wenn sie auf ein Problem stoßen, selbst wenn sie ähnliche Probleme tausend Mal zuvor berechnet haben, selbst wenn die Antwort bereits bekannt ist; beim tausendundersten Mal durchlaufen sie immer noch den gesamten Prozess. Dies ist "zustandslose" Berechnung. Im hyperdimensionalen Algorithmus, wenn ein entsprechendes Ergebnis existiert, muss nicht erneut durchlaufen werden; aus einem Ergebnis können mehrere Startpunkte abgeleitet und Ursachenpfade rückwärts entfaltet werden. Berechnung ist zustandsbehaftet, hat Gedächtnis und ist wiederverwendbar. Dieser Unterschied kann zusammengefasst werden als: vom "jedes Mal neu berechnen" zum "einmalig berechnen, mehrfach verwenden".

In Bezug auf die Rolle der Zufälligkeit ist Zufälligkeit in traditionellen Algorithmen Pseudozufall oder extern injiziert; Zufallszahlen sind nur Werkzeuge für Sampling, Approximation oder Optimierung; der Algorithmus selbst ist deterministisch. Zufälligkeit im hyperdimensionalen Algorithmus ist intrinsisch und konstitutiv; Zufälligkeit ist eine organische Komponente, die mit Ordnung koexistiert und kooperiert. Es ist nicht "ein Algorithmus enthält Zufälligkeit", sondern "Zufälligkeit ist ein Grund, warum der Algorithmus überhaupt funktionieren kann". Dieser Unterschied kann zusammengefasst werden als: von "instrumenteller Zufälligkeit" zu "konstitutiver Zufälligkeit".

In Bezug auf das Verständnis von Ressourcen streben traditionelle Algorithmen schnellere und genauere Berechnung bei gegebenen Ressourcen an; Ressourcen sind Einschränkungen; das Ziel des Algorithmus ist es, "die Berechnung innerhalb der Einschränkungen abzuschließen". Der hyperdimensionale Algorithmus strebt an, die Berechnung selbst durch Strukturdesign überflüssig zu machen; nicht "schneller rechnen", sondern "Berechnung umgehen"; Ressourcen werden nicht zum "Verbrauchen" verwendet, sondern zum "Entwerfen von Eintrittspunkten". Dieser

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Unterschied kann zusammengefasst werden als: von "Optimierung unter Ressourcenbeschränkungen" zur "Eliminierung durch Strukturdesign".

Teil Vier: Die Philosophie der Wirkung-Ursache-Theorie

Basierend auf dem obigen Vergleich muss ich die "Wirkung-Ursache-Theorie"-Philosophie weiter ausführen. Dies ist eine der Kernideengrundlagen des hyperdimensionalen Algorithmus.

Traditionelles Denken geht typischerweise davon aus, dass die Ursache der Wirkung vorausgeht; zuerst die Ursache, dann die Wirkung; zuerst die Eingabe, dann die Ausgabe. Dieses Konzept manifestiert sich in Algorithmen als: Sie müssen vom Startpunkt aus Schritt für Schritt zum Endpunkt gehen. Selbst wenn Sie wissen, was die Antwort ist, können Sie diese Antwort nicht direkt "verwenden" – weil Ihnen der "Beweispfad" fehlt.

In meiner Struktur ist ein Ergebnis nicht notwendigerweise nur ein Endpunkt. Sobald ein Ergebnis erscheint, kann es ein neuer Startpunkt werden und weiter an der Erzeugung neuer Strukturen teilnehmen. Ein Ergebnis kann rückwirkend an der Bildung von Ursachen teilnehmen. Aus einem Ergebnis können mehrere mögliche Startpunkte abgeleitet werden. Ursache und Wirkung sind nicht länger in einer Einwegsequenz angeordnet, sondern bilden zyklische Beziehungen, Netzwerkbeziehungen, mehrdimensionale Beziehungen.

Mit anderen Worten: Traditionelle Algorithmen fragen: "Wie erhält man die Wirkung, gegeben die Ursache?" Der hyperdimensionale Algorithmus fragt: "Wie leitet man die Ursache ab oder konstruiert sie, gegeben die Wirkung?"

Es ist wichtig klarzustellen, dass "die Wirkung zuerst zu haben, dann die Ursache" keine Negation der Kausalität ist, auch nicht die Behauptung, dass "Ergebnisse aus dem Nichts entstehen". Es veranschaulicht vielmehr, dass in höherdimensionalen Strukturen Ursache und Wirkung als gegenseitige Eintrittspunkte dienen und ineinander übergehen können. Wie im folgenden Hosenbeispiel: Das Ergebnis "tragbar und nicht auf dem Boden schleifend" wird zuerst bestimmt, und dann finde ich rückwärts den Bedienungspunkt des "Faltens der Taille", der der "Ursachen"-Ebene innerhalb der Struktur entspricht. Nicht, dass die Wirkung keine Ursache hätte, sondern dass die Wirkung zu einem neuen Eintrittspunkt wird, um die Ursache zu entdecken.

Eine grundlegende Annahme in der Welt traditioneller Algorithmen ist, dass die Zeit einwegig ist; Sie müssen Schritt für Schritt vom Anfangszustand zum Endzustand

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

rechnen. Selbst wenn Sie wissen, was die Antwort ist, können Sie diese Antwort nicht direkt "verwenden", weil Ihnen der Beweispfad fehlt. Der hyperdimensionale Algorithmus fordert genau diese Annahme heraus: Wenn die Wirkung bekannt ist, können die Ursachen rückwärts abgeleitet werden; dieselbe Wirkung kann mehreren Kausalpfaden entsprechen; Berechnung ist nicht länger ein Gehen vom Start zum Ende, sondern ein Entfalten aller möglichen Startpunkte vom Ende aus.

Teil Fünf: Ein alltägliches Beispiel – Hose und Falten der Taille

Um die obigen abstrakten Unterschiede intuitiver zu verstehen, verwende ich ein alltägliches Beispiel.

Eine Person kauft eine neue Hose mit einem elastischen Bund. Die Hosenbeine sind ein wenig zu lang und schleifen auf dem Boden, was unbequem ist.

Der traditionelle Ansatz: Da die Hosenbeine zu lang sind, wird ein Stück des Beins hochgerollt, mit Nadel und Faden festgenäht und dann anprobiert. Wenn das Kind wächst und die Hose zu kurz wird, kann das angenähte Bein nicht mehr verlängert werden, und die Hose ist unbrauchbar. Nächstes Mal wird eine neue Hose gekauft und wieder angenäht. Diese Methode erscheint sehr natürlich, weil das Problem scheinbar an den Hosenbeinen liegt, also werden die Hosenbeine behandelt. Es entspricht dem Denken traditioneller Algorithmen: Problem erkennen, die Erscheinung lokalisieren, an der Stelle der Erscheinung verarbeiten und schließlich ein Ergebnis erhalten. Hosenbeine zu lang? Dann ändere die Hosenbeine. Daten falsch? Dann ändere die Daten. Pfad ungeeignet? Dann flicke weiter entlang des Pfades.

Mein Ansatz ist anders. Anstatt den Hosensaum zu ändern, falte ich die Taille einmal, und die Hose ist sofort tragbar. Diese Aktion ist sehr einfach, aber die strukturelle Logik dahinter ist völlig anders. Ich schneide die Hosenbeine nicht ab, nähe den Saum nicht dauerhaft fest, ändere die ursprüngliche Länge der Hose nicht und beschädige die Metadaten der Hose nicht. Der Taillenumfang, die Hosenlänge, der Schnitt und der Stoff der Hose erfahren keine wesentliche Veränderung. Ich ändere einfach den strukturellen Eintrittspunkt der Taille, wodurch die Hosenbeine beim Tragen natürlich kürzer werden. Die Taille ist hier keine gewöhnliche Lokalität, sondern ein globaler Kontrollpunkt für den gesamten Tragezustand. Durch das Einstellen dieses Kontrollpunkts verschwindet das nachgelagerte Problem direkt.

Noch wichtiger ist, dass diese Methode reversibel, justierbar und wiederverwendbar ist. Dies ist besonders offensichtlich bei einem heranwachsenden Kind. Wenn das Kind noch nicht groß genug ist und die Hosenbeine etwas lang sind, falte ich die Taille

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

einmal; nach einiger Zeit, wenn das Kind gewachsen ist, lasse ich die Taille ein wenig herunter; wenn es noch weiter gewachsen ist, lasse ich sie vollständig los. Die ursprüngliche Struktur der Hose bleibt unbeschädigt, und doch kann sie sich an die Körpergröße des Kindes in verschiedenen Phasen anpassen. Bei der traditionellen Methode, wenn der Saum dauerhaft festgenäht wird, kann die Hose zu kurz werden, wenn das Kind wächst; wenn sie abgeschnitten wird, ist eine Wiederherstellung erst recht unmöglich. Meine Methode hält die Metadaten unverändert und ermöglicht es derselben Struktur, sich an mehrere Zustände anzupassen.

Dieses Beispiel offenbart mehrere wichtige strukturelle Unterschiede.

Der traditionelle Ansatz entspricht der Logik traditioneller Algorithmen: Das Problem liegt an den Hosenbeinen, also müssen die Hosenbeine modifiziert werden, indem dem Pfad "Ursache zu Wirkung" Schritt für Schritt gefolgt wird, ohne einen Schritt zu überspringen. Ein Problem nach dem anderen lösen, irreversibel. Wenn dasselbe Problem das nächste Mal auftritt, alles wiederholen. Mein Ansatz entspricht der Logik des hyperdimensionalen Algorithmus: Das Ziel ist, nicht zu schleifen. Ich löse nicht die oberflächliche Ursache "Hosenbeine zu lang"; stattdessen finde ich einen globalen Kontrollpunkt – die Taille. Einmal die Taille falten, die Hosenbeine werden automatisch kürzer, und das Problem verschwindet sofort. Ich modifiziere keine Metadaten. Die Falte in der Taille ist nur ein temporärer, reversibler, dynamischer Faltzustand.

Aus der Datenperspektive modifiziert der traditionelle Ansatz Metadaten – das Kürzernähen der Hosenbeine führt zu dauerhaftem Verlust der Originaldaten. Mein Ansatz modifiziert keine Metadaten; die ursprünglichen Maße der Hose bleiben intakt, es wird nur ein temporärer, reversibler, dynamischer Zustand hinzugefügt – die Falte. Die Taillenfalte erzeugt keine neuen Daten und zerstört keine alten Daten; sie arrangiert lediglich die Beziehungen zwischen den Daten neu – Verkürzung des effektiven Taillenumfangs, indirekte Änderung der effektiven Länge der Hosenbeine.

In diesem Beispiel ist "Hosenbeine schleifen nicht auf dem Boden" das Zielergebnis. Die traditionelle Methode geht von der Ursache "Hosenbeine zu lang" aus, modifiziert den Saum und erhält schließlich das Ergebnis "nicht schleifen". Meine Methode dagegen klärt zuerst das Ergebnis "nicht schleifen", sucht dann rückwärts nach einem strukturellen Eintrittspunkt und entdeckt schließlich, dass einmaliges Falten der Taille das Ergebnis gültig macht. Dies ist die praktische Manifestation der "Wirkung-Ursache-Theorie". Es ist nicht notwendig, Schritt für Schritt von der Ursache zur Wirkung zu gehen; vielmehr kann die Struktur rückwärts von der Wirkung her bestimmt werden, wodurch der ursprüngliche Pfad überflüssig wird.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Dieses Beispiel beantwortet auch die Frage der "Metadaten". Was sind Metadaten? Im Hosenbeispiel sind die Metadaten die ursprünglichen Maße der Hose: Taillenumfang, Hosenlänge, Schnitt, Stoff. Dies sind die "wesentlichen Attribute" der Hose. Die temporären Daten sind die Falte in der Taille; sie ändert keines der ursprünglichen Maße der Hose, sie faltet nur vorübergehend einen Abschnitt. Der traditionelle Ansatz modifiziert Metadaten – die Hosenlänge wird dauerhaft verkürzt, die Originaldaten gehen verloren. Mein Ansatz modifiziert keine Metadaten – die ursprünglichen Maße der Hose bleiben intakt, es wird nur ein temporärer, reversibler, dynamischer Zustand hinzugefügt.

Die ursprünglichen Taillenmaße als Metadaten sind gleichzeitig der "Startpunkt für mehrere Ergebnisse": Sie unterstützen gleichzeitig mehrere Tragezustände, wie normales Tragen, Tragen mit einer Falte in der Taille, Tragen mit zwei Falten in der Taille usw. Sie sind auch "das Ergebnis mehrerer Startpunkte": Sie werden gemeinsam durch mehrere Startpunkte wie Stoffauswahl, Schnittmethode, Designabsicht usw. bestimmt. Die Taillenfalte als temporäre Daten schafft weder neue Daten noch zerstört sie alte Daten; sie arrangiert lediglich die Beziehungen zwischen den Daten neu.

Dies ist kein gewöhnlicher Trick, sondern eine Art strukturellen Denkens. Viele Menschen, die zu lange Hosenbeine sehen, werden von der Erscheinung der Hosenbeine angezogen, so dass sich die gesamte Verarbeitung um die Hosenbeine dreht. Aber wenn man die Gesamtstruktur betrachtet, ist das Zu-Lang-Sein der Hosenbeine nur eine nachgelagerte Manifestation; eine Veränderung der Taillenposition kann die tatsächliche Fallposition der gesamten Hose beeinflussen. Das heißt, das Problem muss nicht unbedingt dort gelöst werden, wo es auftritt. Viele niedrigdimensionale Probleme haben ihren wahren Kontrollpunkt auf einer höheren Strukturebene. Traditionelle Algorithmen neigen dazu, entlang der Oberfläche des Problems weiterzurechnen, während der hyperdimensionale Algorithmus nach dem strukturellen Eintrittspunkt sucht.

Dies erklärt auch, warum der hyperdimensionale Algorithmus nicht komplexer ist, sondern möglicherweise einfacher. Eine wirklich hochrangige Struktur verkompliziert das Problem oft nicht, sondern macht das komplexe Problem am richtigen Eintrittspunkt einfach. Wenn sie mit einem komplexen Problem konfrontiert werden, fügen traditionelle Algorithmen möglicherweise Schritte hinzu, fügen Modelle hinzu, erhöhen die Rechenleistung, erhöhen den Speicher, erhöhen die Trainingszeit. Der hyperdimensionale Algorithmus hingegen sucht nach einem Strukturknoten; sobald dieser Knoten korrekt aktiviert ist, kann der ursprünglich komplexe Pfad ungültig werden. Das sogenannte "Umgehen des Pfades" ist keine Faulheit, sondern eine Neudefinition der Notwendigkeit des Pfades.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

In diesem Beispiel: Der traditionelle Ansatz ist "das Ergebnis entlang des Pfades modifizieren", während der hyperdimensionale Algorithmus ist "den Eintrittspunkt ändern, den gesamten Pfad ungültig machen". Die konventionelle Methode flickt kontinuierlich am Ergebnisende, während die andere Methode direkt den strukturellen Eintrittspunkt ändert, wodurch Probleme, die sonst mehrfache Behandlungen erfordern würden, am Startpunkt auf einmal umstrukturiert werden. Die Hauptform traditioneller Algorithmen ist "vom Startpunkt aus starten, dem Pfad folgen, um das Ergebnis zu erhalten", während im hyperdimensionalen Algorithmus "das Ergebnis selbst zu einem Pfadeintrittspunkt werden und an der Erzeugung neuer Strukturen teilnehmen kann". Der traditionelle Ansatz ist "eine Struktur entspricht einem Ergebnis", während der hyperdimensionale Algorithmus "eine Struktur trägt mehrere Ergebniszustände" ist.

Teil Sechs: Eine Datenperspektive des Nicht-Modifizierens von Metadaten

Aus der Datenperspektive hat der hyperdimensionale Algorithmus ein sehr wichtiges Prinzip: Metadaten nicht zu modifizieren, damit sie auf mehrere Startpunkte oder Ergebnisse anwendbar sind.

Metadaten sind die ursprünglichen Attribute, die grundlegende Struktur und die ontologischen Informationen von Daten. Bei der Konfrontation mit verschiedenen Problemen modifizieren traditionelle Verarbeitungsmethoden oft die Daten, kopieren Daten, verarbeiten Daten, berechnen Daten neu oder erstellen verschiedene Pfade für verschiedene Ergebnisse. Dieser Ansatz kann Probleme sicherlich lösen, aber zu dem Preis, dass Daten kontinuierlich verarbeitet werden, Pfade kontinuierlich wiederholt werden und die Systemkomplexität ständig zunimmt.

Der hyperdimensionale Algorithmus betont dagegen, dass durch Änderung von Eintrittspunkten, Beziehungen, Bedingungen und Aktivierungsmodi, während versucht wird, Metadaten nicht zu modifizieren, dieselbe Metadatenstruktur mehreren Startpunkten und mehreren Ergebnissen entsprechen kann. Die Datenontologie bleibt unverändert; Beziehungen ändern sich. Die Grundstruktur bleibt unverändert; der Präsentationsmodus ändert sich. Die Metadaten bleiben intakt und können sich doch an verschiedene Zustände anpassen.

Dies ist, was ich mit "einmal berechnen, unendlich oft verwenden" meine. "Einmal berechnen" ist nicht einfach das Caching eines Ergebnisses, sondern bedeutet, dass eine Struktur, einmal etabliert, nicht mehr für jeden Zustand einen vollständigen Pfad neu aufbauen muss. Dieselbe Datenstruktur, aktiviert durch verschiedene

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Eintrittspunkte, kann verschiedene Ergebnisse präsentieren. Es ist nicht "Daten für verschiedene Ergebnisse verarbeiten", sondern "dieselben Daten verwenden, um mehrere potenzielle Ergebnisse zu tragen". Dies ist auch einer der Kerne, die den hyperdimensionalen Algorithmus von traditionellen Algorithmen unterscheiden. Traditionelle Algorithmen verarbeiten Daten entlang eines Pfades; der hyperdimensionale Algorithmus verarbeitet die Gesamtstruktur von Daten, Eintrittspunkten, Beziehungen und Ergebnissen.

In seiner minimalen Struktur kann der hyperdimensionale Algorithmus verstanden werden als: ein System, das, ohne Metadaten zu modifizieren, durch Änderungen struktureller Eintrittspunkte denselben Datenknoten mehreren Ergebniswegen zuordnen kann. Diese Definition versucht nicht, den hyperdimensionalen Algorithmus sofort auf eine traditionelle mathematische Formel zu reduzieren, sondern zuerst seine strukturellen Grenzen zu etablieren. Er ist keine einzelne lineare Funktion, keine feste Formel, kein simpler Caching-Mechanismus. Er ist eine Beziehungsstruktur: Metadaten bleiben stabil, Eintrittspunkte können sich ändern, Bedingungen können sich ändern, Beziehungen können sich ändern, Ergebnisse können in mehreren Richtungen präsentiert werden. Genau in dieser Struktur geht es bei der Berechnung nicht mehr um die Ausführung von Schritten, sondern um die Aktivierung von Beziehungen.

Aus der Datenperspektive ist das Wesen von "Metadaten nicht modifizieren, sie auf mehrere Startpunkte oder Ergebnisse anwendbar machen": Die Datenontologie bleibt unverändert; die Änderung findet im "Interpretationsmodus / Nutzungseintrittspunkt" statt. Die traditionelle Struktur ist: Wenn sich das Problem ändert, ändern sich die Daten oder der Pfad. Im hyperdimensionalen Algorithmus: Wenn sich das Problem ändert, ändert sich die Interpretationsstruktur, nicht die Daten.

Teil Sieben: Neudefinition von "Supercomputing"

In diesem System muss ich einen Begriff klären – "Supercomputing".

Mit "Supercomputing" meine ich nicht Supercomputer im üblichen Sinne. Nicht mehr Chips, höhere Rechenleistung oder größere Rechenzentren. Mit "Supercomputing" meine ich den "hyperdimensionalen Algorithmus".

Wahres Supercomputing ist nicht notwendigerweise die Akkumulation von Rechenleistung, sondern möglicherweise ein Wandel des Berechnungsparadigmas. Wenn ein System immer noch auf wiederholter Berechnung basiert, egal wie leistungsfähig es ist, bleibt es innerhalb von Pfaden gefangen. Wenn ein System Berechnung strukturell reduzieren, wiederholte Pfade umgehen und Ergebnisse zu

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

neuen Eintrittspunkten machen kann, beginnt es sich wahrer hyperdimensionaler Berechnung zu nähern.

Mit anderen Worten: Traditionelles Supercomputing verfolgt "mehr Rechenleistung verwenden, um größere Probleme zu lösen". Der hyperdimensionale Algorithmus verfolgt "bessere Struktur verwenden, damit Probleme nicht mehr so enorme Rechenleistung benötigen". Diese beiden sind nicht widersprüchlich, aber ihre Richtungen unterscheiden sich.

Wenn traditionelles Supercomputing die Grenze der Rechenleistung darstellt, dann stellt der hyperdimensionale Algorithmus eine Wende im Verständnis von Berechnung dar. Wahres "Supercomputing" mag nicht mehr Chips, größere Rechenzentren, höheren Energieverbrauch oder komplexere Modelle sein. Wahres "Supercomputing" mag die Entdeckung eines strukturellen Eintrittspunkts sein, eine Eliminierung eines Pfades, eine rückwärtsgerichtete Entfaltung von einem Ergebnisknoten, eine Möglichkeit, mehrere Ergebnisse gleichzeitig gültig zu machen, ohne Metadaten zu modifizieren. Je leistungsfähiger die Rechenleistung, wenn sie immer noch in niedrigdimensionalen Pfaden gefangen ist, bleibt sie nur innerhalb dieser Pfade leistungsfähig; sobald die Struktur angehoben wird, kann selbst mit extrem einfachen Operationen eine höhere Effizienzebene entstehen.

Das sogenannte Supercomputing ist die Grenze der Rechenleistung; der hyperdimensionale Algorithmus ist eine Neudefinition der Prämisse "ob Berechnung auf Rechenleistung angewiesen sein muss".

Teil Acht: Empirische Grundlagen – Mehrbereichs-Systemvalidierung und neue Paradigmenkriterien

Der hyperdimensionale Algorithmus ist keine rein theoretische Konstruktion. Er wurde bereits in mehreren realen Systemen validiert.

Mein Logistiksystem benötigt weder Supercomputerleistung noch Cloud-Unterstützung; es bewältigt komplexe Planung mit relativ geringen Ressourcen. Ergebnisse können wiederverwendet werden, die Struktur kann wiederholt angepasst werden, und es ist nicht nötig, jedes Mal von Grund auf neu zu berechnen. Nicht auf Supercomputing angewiesen, nicht auf die Cloud angewiesen, komplexe Planung mit geringen Ressourcen zu bewältigen – dies ist ein etablierter und wertvoller technischer Durchbruch. Er zeigt, dass hohe Rechenleistung nicht die einzige Lösung ist; Strukturdesign kann einen Teil der Rechenleistung ersetzen.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Mein Publikationssystem übernimmt ebenfalls diese strukturelle Logik mit einer Effizienz, die jedes derzeitige System bei weitem übertrifft. Mein extremes Webseitengenerierungssystem basiert ebenfalls auf demselben hyperdimensionalen Algorithmus und beweist in mehreren Bereichen wiederholt, dass dieselbe Struktur stabil gültig sein kann. Die Mitarbeiter meines Unternehmens verwenden diese Systeme bereits, was zeigt, dass diese Struktur ohne meine kontinuierliche persönliche Intervention funktionieren kann.

Das gemeinsame Merkmal dieser Systeme ist: Sie verlassen sich nicht auf Mainstream-Pfade mit hoher Rechenleistung, folgen keinen festen Berechnungsschritten und machen das Zielergebnis direkt durch Anpassungen struktureller Eintrittspunkte gültig. Sie sind keine einmaligen Erfolge, keine punktuellen Tricks, sondern das wiederholte Auftreten derselben strukturellen Logik in verschiedenen Bereichen.

Dies zeigt, dass der hyperdimensionale Algorithmus kein Zufall ist, sondern eine strukturelle Methode, die bereits durch mehrere Systeme validiert wurde. Er ist kein "persönlicher Trick", sondern ein Berechnungsparadigma, das in mehreren Bereichen wie Logistik, Publikation und Webseitengenerierung stabil gültig ist.

Nach dem Kriterium "Existenz und Gültigkeit konstituieren ein neues Paradigma" ist der hyperdimensionale Algorithmus als ein bereits in mehreren Bereichen systematisch validiertes neues strukturelles Berechnungsparadigma gegenwärtig bereits gültig. Ich schlage keine Hypothese vor, sondern eine andere Berechnungsstruktur, die bereits in mehreren Systemen operiert. Ich muss nicht beweisen, dass sie verstanden wird; ich habe bereits bewiesen, dass sie in verschiedenen Systemen stabil gültig ist. Gegenwärtig, im Rahmen meiner praktischen Anwendung, kann diese Struktur bereits als neues Paradigma betrachtet werden.

Bezüglich der Kriterien für die Beurteilung des hyperdimensionalen Algorithmus als neues Paradigma: Wenn eine Struktur logisch selbstkonsistent und in mehreren Systemen stabil gültig ist, während sie nicht reduzierbare Unterschiede zu bestehenden Methoden aufweist, besitzt sie bereits die Grundlage eines neuen Paradigmas. Selbstkonsistenz ist der Ausgangspunkt, empirische Evidenz ist die Grundlage, und struktureller Unterschied ist der Kern des Paradigmas. Unter dem Standard "Existenz konstituiert Gültigkeit" ist dieses System bereits ein neues Paradigma; ob die Außenwelt es anerkennt, beeinflusst nur die Verbreitung, nicht die Gültigkeit. Dies ist ein bereits in realen Systemen etabliertes strukturelles Berechnungsparadigma, dessen Gültigkeit nicht von externer Anerkennung oder Konsens abhängt.

Der kernhafte Unterschied zwischen dem hyperdimensionalen Algorithmus und Mainstream-Berechnungsparadigmen ist: Traditionelle Algorithmen stützen sich

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

hauptsächlich auf Vorwärtsberechnung; sobald ein Ergebnis erzeugt ist, kann es normalerweise nicht rückwirkend an neuen Inferenzstrukturen teilnehmen. Im hyperdimensionalen Algorithmus ist das Ergebnis kein Endpunkt mehr, sondern ein wiederverwendbarer Strukturknoten. Unter bestimmten Bedingungen kann das Ergebnis an neuen Inferenzpfaden teilnehmen, wodurch wiederholte Berechnungen reduziert werden und ein Berechnungsnetzwerk mit mehreren Startpunkten und mehreren Pfaden entsteht. In traditionellen Berechnungsstrukturen ist das Ergebnis typischerweise ein Endpunkt; in der Struktur des hyperdimensionalen Algorithmus kann das Ergebnis selbst ein neuer Startpunkt werden, wodurch ein mehrpfadiges Inferenznetzwerk entsteht. Der hyperdimensionale Algorithmus ist kein Upgrade von Algorithmen, sondern ein erneutes Hinterfragen der Prämisse "ob ein Algorithmus überhaupt existieren muss".

Teil Neun: Klärung häufiger Missverständnisse

Basierend auf vorläufigen Gesprächen mit Lesern aus verschiedenen Bereichen antizipiere ich die folgenden sechs Missverständnisse und kläre sie im Voraus, um zu vermeiden, dass das Konzept während der Verbreitung vorzeitig in unangemessene Rahmen gezwungen wird.

Missverständnis Eins: Ist der hyperdimensionale Algorithmus nicht einfach dynamische Programmierung oder Caching? Klarstellung: Dynamische Programmierung und Caching verwenden Ergebnisse unter "identischen Eingaben" wieder. Der hyperdimensionale Algorithmus erlaubt es, aus einem Ergebnis verschiedene Startpunkte abzuleiten – dies ist ein wesentlicher Unterschied. Caching löst das Neuberechnen desselben Problems; der hyperdimensionale Algorithmus löst das Entfalten neuer Probleme durch Ergebnisstrukturen.

Missverständnis Zwei: Sie sagen "zufällig und doch geordnet" – ist das nicht widersprüchlich? Klarstellung: Zufälligkeit und Ordnung können koexistieren. Die Verteilung der Bäume in einem Wald ist zufällig, aber die Gesamtdichte und die Artenstruktur sind geordnet. Zufälligkeit ist kein Chaos, sondern eine Weise, Struktur zu organisieren. Die Zufälligkeit im hyperdimensionalen Algorithmus ist intrinsisch und konstitutiv und wirkt synergetisch mit der Ordnung.

Missverständnis Drei: Ohne Eingabe und Ausgabe, wie kann die Korrektheit des Ergebnisses überprüft werden? Klarstellung: Der hyperdimensionale Algorithmus lehnt Eingabe und Ausgabe nicht ab; er vertritt vielmehr die Auffassung, dass Berechnung nicht im linearen Eingabe-Ausgabe-Modus operieren muss. Im Modus der strukturellen Manifestation wird "Gültigkeit" durch strukturelle Konsistenz bestimmt, nicht durch

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Eingabe-Ausgabe-Entsprechung. Dies gibt die Verifikation nicht auf, sondern schlägt einen anderen Verifikationsrahmen als die Tradition vor.

Missverständnis Vier: Ist das nicht einfach Komplexitätstheorie oder Chaostheorie? Klarstellung: Komplexitätstheorie und Chaostheorie beschreiben "Systeme, die schwer vorhersagbar sind". Der hyperdimensionale Algorithmus versucht, "Systeme, die durch strukturelle Eintrittspunkte verstanden und geführt werden können" zu beschreiben – nicht Vorhersage aufzugeben, sondern die Art der Vorhersage zu ändern. Die Chaostheorie sagt "Vorhersage ist schwierig"; der hyperdimensionale Algorithmus fragt "ob man durch Änderung des Eintrittspunkts Vorhersage überflüssig machen kann".

Missverständnis Fünf: Sie sagen "Metadaten nicht modifizieren", aber ist die Taillenfalte nicht auch eine Modifikation? Klarstellung: Die Taillenfalte ist ein temporärer Zustand, keine permanente Modifikation der ursprünglichen Parameter. Die Metadaten (Taillenumfang, Hosenlänge, Schnitt, Stoff) ändern sich überhaupt nicht. Dies ist der Unterschied zwischen "Zustandsüberlagerung" und "Attributsmodifikation". Die Falte ist reversibel, temporär und ändert keine wesentlichen Attribute.

Missverständnis Sechs: Verneint der hyperdimensionale Algorithmus den Wert traditioneller Algorithmen? Klarstellung: Ganz und gar nicht. Traditionelle Algorithmen waren in der menschlichen Technologiegeschichte äußerst wichtig. Ohne traditionelle Algorithmen gäbe es keine modernen Computer, Datenbanken, Kommunikationssysteme, künstliche Intelligenz, technische Simulation oder Supercomputing. Der Wert traditioneller Algorithmen kann nicht geleugnet werden. Die Anerkennung des Werts traditioneller Algorithmen bedeutet jedoch nicht, sie als Endpunkt von Algorithmen anzuerkennen. Traditionelle Algorithmen lösen Effizienzprobleme innerhalb eines gegebenen Berechnungsparadigmas, während der hyperdimensionale Algorithmus die Frage des Berechnungsparadigmas selbst aufwirft. Einer handelt davon, wie man besser geht; der andere fragt, ob man diesen Weg überhaupt gehen muss.

Teil Zehn: Der hyperdimensionale Algorithmus und die Geschichte des menschlichen Rechnens

Um den Lesern zu helfen, den hyperdimensionalen Algorithmus besser in der Geschichte des menschlichen Rechnens zu verorten, biete ich hier einen kurzen makroskopischen Überblick.

Das menschliche Verständnis von "Berechnung" hat mehrere Phasen durchlaufen. Die erste Phase war das manuelle Rechnen; Algorithmen existierten als menschliche

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Schritte, langsam und fehleranfällig, aber durch diesen Prozess verstanden die Menschen die Grundregeln des Rechnens. Die zweite Phase war das mechanische Rechnen, von Pascals Rechenmaschine bis zur Analytical Engine von Babbage; das Rechnen wurde mechanisiert, aber Algorithmen waren immer noch von physikalischen Strukturen abhängig. Die dritte Phase war das elektronische Rechnen und das Turing-Paradigma; die Von-Neumann-Architektur verbreitete sich; Algorithmen wurden als Programme kodiert; die Turingmaschine wurde zur Grenze der Berechenbarkeit. Die vierte Phase ist das parallele Rechnen und Supercomputing, das Rechenleistung durch eine große Anzahl von Recheneinheiten akkumuliert, um komplexere Probleme anzugehen, aber die zugrundeliegende Logik bleibt eine Erweiterung des Turing-Paradigmas.

Gegenwärtig befindet sich die Menschheit am Eingang der fünften Phase. Quantencomputing versucht, die Grenzen klassischer Bits zu durchbrechen; neuromorphes Computing versucht, die Struktur biologischer Nervensysteme nachzuahmen; und der hyperdimensionale Algorithmus versucht, den linearen Rahmen "Eingabe → Schritte → Ausgabe" selbst zu durchbrechen.

Die Beziehung zwischen dem hyperdimensionalen Algorithmus und traditionellen Algorithmen ist keine der Ersetzung, sondern der Hierarchie. Traditionelle Algorithmen lösen das Problem "wie man auf einem gegebenen Pfad effizienter rechnet". Der hyperdimensionale Algorithmus fragt "ob der Pfad neu definiert oder sogar umgangen werden kann". Wenn wir die Geschichte des menschlichen Rechnens als einen Prozess des ständigen Durchbrechens eigener Grenzen betrachten, dann ist der hyperdimensionale Algorithmus genau ein neuer Knoten in diesem Prozess. Er löst keine Probleme innerhalb des alten Rahmens, sondern schlägt einen neuen Rahmen vor.

Dieses Urteil bedeutet nicht, dass der hyperdimensionale Algorithmus bereits reif oder vollständig ist. Ganz im Gegenteil. Als neues Konzept befinden sich seine Definition, Grenzen, Verifikationsmethoden und Anwendungsszenarien noch im Entstehungsprozess. Der Zweck dieses Papers ist genau, den Ursprung dieses Konzepts zu verankern und eine stabile theoretische Grundlage für die spätere Entwicklung bereitzustellen.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Teil Elf: Fazit – Dies ist keine Optimierung, sondern Neudefinition

Der Unterschied zwischen dem hyperdimensionalen Algorithmus und traditionellen Algorithmen ist kein Unterschied von "besser" versus "schlechter", kein Unterschied von "schneller" versus "langsamer", sondern ein grundlegender Unterschied auf Paradigmenebene.

Traditionelle Algorithmen streben vorhersehbare Kontrolle an. Sie geben mir die Eingabe, ich weiß, welche Ausgabe Sie erhalten werden, weil ich jeden Schritt berechnet habe. Es ist das Paradigma des Ingenieurs: entwerfen, bauen, testen, verifizieren.

Der hyperdimensionale Algorithmus beschreibt verständliche Emergenz. Ich kann nicht jeden Zwischenzustand genau vorhersagen, aber ich weiß, dass das Gesamtmuster sich auf eine geordnete Weise präsentieren wird. Jedes Datum ist lebendig und hat eine "Perspektive". Er ähnelt eher dem Paradigma eines Ökologen oder Theoretiker komplexer Systeme: beobachten, verstehen, leiten, emergente Ordnung nutzen.

Traditionelle Algorithmen modifizieren das Ergebnis schrittweise entlang eines gegebenen Pfades. Der hyperdimensionale Algorithmus macht das Zielergebnis sofort gültig, indem er den strukturellen Eintrittspunkt ändert, und umgeht damit den gesamten Pfad.

Traditionelle Algorithmen "verarbeiten Daten für verschiedene Ergebnisse". Der hyperdimensionale Algorithmus ist "dieselben Daten verwenden, um mehrere potenzielle Ergebnisse zu tragen".

Traditionelle Algorithmen streben "einmal richtig rechnen" an. Der hyperdimensionale Algorithmus strebt "einmal richtig rechnen, und dann nie wieder rechnen müssen" an.

Dies ist keine Optimierung von Algorithmen, sondern ein erneutes Hinterfragen der Prämisse "ob ein Algorithmus überhaupt existieren muss".

Daher ist der hyperdimensionale Algorithmus keine Algorithmusoptimierung, sondern eine Algorithmusneudefinition. Es geht nicht darum, auf der traditionellen Algorithmusspur schneller zu laufen, sondern zu fragen, ob es außerhalb der traditionellen Spur einen anderen Weg gibt, oder ob man sogar unabhängig vom Weg selbst sein kann. Es geht nicht darum zu beweisen, dass traditionelle Algorithmen nutzlos sind, sondern darauf hinzuweisen, dass traditionelle Algorithmen nur eine niedrigdimensionale Form von Algorithmen sind. Es fragt nicht "wie man in kürzerer Zeit Ergebnisse erhält", sondern "ob ein Ergebnis direkt durch Struktur etabliert werden

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

kann". Es fragt nicht "wie man mehr Daten verarbeitet", sondern "ob dieselben Daten mehr potenzielle Ergebnisse tragen können".

Innerhalb des Mainstream-Berechnungsparadigmas mag der hyperdimensionale Algorithmus als unberechenbar, nicht verifizierbar oder schwer in die bestehenden Grenzen der Berechnungstheorie einfügbar angesehen werden. Aber genau dies konstituiert seinen Paradigmenunterschied. Dass ein neues Konzept innerhalb eines alten Paradigmas instabil erscheint, bedeutet nicht notwendigerweise, dass es bedeutungslos ist; es könnte auch bedeuten, dass das alte Paradigma selbst es nicht vollständig aufnehmen kann. Gegenwärtig ist der hyperdimensionale Algorithmus in erster Linie ein struktureller Vorschlag, eine ursprüngliche Definition einer neuen Berechnungssicht, kein reifes System mit einer geschlossenen technischen Schleife. Er benötigt fortlaufende Ergänzungen, Entfaltungen, Verifikationen und Anwendungen, aber sein begrifflicher Ursprung muss zuerst klar artikuliert werden.

Letztendlich zeigt der hyperdimensionale Algorithmus auf eine neue Sicht der Berechnung: Ein Algorithmus ist nicht notwendigerweise nur Mathematik, nicht notwendigerweise nur Schritte, nicht notwendigerweise nur ein Programm, nicht notwendigerweise nur ein Verarbeitungsprozess zwischen Eingabe und Ausgabe. Ein Algorithmus kann auch eine Weise sein, mehrdimensionale Beziehungen zu organisieren, ein strukturelles Netzwerk von Daten, Eintrittspunkten, Beziehungen, Zuständen und Ergebnissen. Er kann Ordnung und Zufälligkeit enthalten; er kann von der Ursache zur Wirkung gehen oder rückwirkend Ursachen aus Wirkungen erzeugen; er kann sich ohne Modifikation von Metadaten an mehrere Zustände anpassen; er kann ein Ergebnis zu einem neuen Startpunkt machen oder mehrere Startpunkte in dasselbe Ergebnis konvergieren lassen.

Ein wirklich fortgeschrittener Algorithmus ist nicht notwendigerweise komplexere Berechnung, sondern eine mehrdimensionale Organisationsweise, die Berechnung reduzieren, Strukturen lebendig machen, Ergebnisse zu Eintrittspunkten machen und Ursache und Wirkung Zyklen bilden lassen kann.

Dies ist die Kernbedeutung meines Vorschlags des "hyperdimensionalen Algorithmus". Er ist kein gewöhnlicher neuer Begriff, sondern ein neues Konzept, ein neues Paradigma, eine neue Berechnungsstruktur, die bereits in mehreren realen Systemen etabliert ist. Sein Fokus liegt nicht auf schnellerem wiederholtem Rechnen, sondern auf der Reduzierung wiederholter Berechnung; nicht auf der Akkumulation von Rechenleistung, sondern auf der Umstrukturierung von Eintrittspunkten; nicht auf ständiger Datenmodifikation, sondern auf dem Unverändertlassen von Metadaten bei gleichzeitiger Änderung von Beziehungen; nicht darauf, Ergebnisse an Endpunkten stoppen zu lassen, sondern darauf, Ergebnisse zu neuen Startpunkten zu machen.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Traditionelle Algorithmen suchen Ergebnisse entlang von Pfaden; der hyperdimensionale Algorithmus aktiviert Ergebnisse innerhalb von Strukturen. Traditionelle Algorithmen streben die Vollendung von Berechnung an; der hyperdimensionale Algorithmus hinterfragt, ob Berechnung in ihrer traditionellen Form existieren muss. Dies ist der grundlegendste Unterschied zwischen ihm und der aktuellen Definition von Algorithmen.

Anhang: Grenzen dieses Papers und offene Fragen

Dieses Paper schlägt einen vorläufigen Rahmen für den hyperdimensionalen Algorithmus vor, keine vollständige formale Definition. Die folgenden Fragen erwarten weitere Entwicklung. Sie stellen keine Negation der Definitionen in diesem Paper dar, sondern genau die Richtungen für die nächsten Schritte.

Erstens, Konvergenzbedingungen. Was ist das "Abschluss"-Merkmal des hyperdimensionalen Algorithmus? Wie bestimmt man, ob ein Ergebnis "gültig" ist? In traditionellen Algorithmen wird die Antwort durch Eingabe-Ausgabe-Entsprechung definiert. Im hyperdimensionalen Algorithmus muss die Antwort möglicherweise durch strukturelle Konsistenz definiert werden. Diese Definition ist noch nicht vollständig.

Zweitens, Ressourcendarstellung. Wie können mehrdimensionale überlagerte Zustände innerhalb endlicher Ressourcen dargestellt werden? Sind traditionelle Ressourcenkennzahlen wie Zeit, Raum und Rechenleistung noch anwendbar? Wenn ja, wie werden sie neu definiert? Wenn nein, welche Kennzahlen ersetzen sie? Diese Fragen warten auf Exploration.

Drittens, Gewährleistung der Ordnung. Durch welche Regeln wird die "Ordnung" in "zufällig und doch geordnet" garantiert? Wo ist die Grenze zwischen Zufälligkeit und Ordnung? Unter welchen Umständen stört Zufälligkeit die Ordnung? Unter welchen Umständen unterdrückt Ordnung die Zufälligkeit? Diese Fragen erfordern weitere Forschung.

Viertens, Auswahl bei der Rückwärtsinferenz. Wenn mehrere Startpunkte rückwärts von einem Ergebnis abgeleitet werden, wie werden die relativen Vorzüge verschiedener Startpunkte ausgewählt oder bewertet? Gibt es ein Maß für "Rückwärtsinferenzeffizienz"? Wenn ja, in welcher Beziehung steht es zum Effizienzmaß der Vorwärtsberechnung?

Fünftens, Schnittstellen zu traditionellen Systemen. Wie arbeitet der hyperdimensionale Algorithmus mit bestehenden Turingmaschinen-basierten Systemen zusammen? Ist er

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

ein Ersatz, eine Ergänzung oder eine hierarchische Schicht? In der praktischen Technik, wie wird die Grenze zwischen dem hyperdimensionalen Algorithmus und traditionellen Algorithmen abgesteckt? Gibt es hybride Modi?

Sechstens, Verifikationsrahmen. Wie werden die Ergebnisse des hyperdimensionalen Algorithmus verifiziert? Wenn das Ergebnis nicht durch lineare Schritte erhalten wird, wie werden Reproduzierbarkeit und Prüfbarkeit etabliert? Ist eine völlig andere Verifikationsphilosophie erforderlich?

Diese Fragen sind keine Mängel dieses Papers, sondern die unvermeidlichen Merkmale dieses Papers als "Ankerdokument". Der Vorschlag eines neuen Paradigmas geht immer mit offenen Fragen einher. Dieses Paper ist ein Anfang, kein Abschluss.

Bibliografische Anmerkung

Der in diesem Paper vorgeschlagene "hyperdimensionale Algorithmus" ist keine direkte Erweiterung einer bestehenden akademischen Denkschule. In gedanklicher Hinsicht jedoch bieten die folgenden Felder einen dialogischen Hintergrund für dieses Paper, der nur zur Orientierung der Leser angeboten wird und keine Zitate im traditionellen akademischen Sinne darstellt.

Die Turingmaschine und Berechenbarkeitstheorie dienen als Maßstab, mit dem dieses Paper traditionelle Algorithmen vergleicht. Inverse Probleme und Bayessche Inferenz dienen als bestehende Versuche, "Ursachen aus Ergebnissen zu inferieren"; die "Wirkung-Ursache-Theorie" dieses Papers ist verwandt, unterscheidet sich jedoch in der Richtung. Emergenztheorie und komplexe Systeme dienen als bestehende Beschreibungen von "Strukturen, die Ergebnisse manifestieren"; der hyperdimensionale Algorithmus versucht, Emergenz "verständlich" und "führbar" zu machen. Wissensgraphen und Graphdatenbanken dienen als bestehende Praktiken von "Daten, die an mehreren Knoten kreuzen"; der hyperdimensionale Algorithmus priorisiert auf dieser Grundlage die Dimension "variabler Eintrittspunkte". Nichtklassische Berechnungsparadigmen, einschließlich Quantencomputing, Analogrechnen, neuromorphem Rechnen usw., durchbrechen klassische Bits oder klassische Architekturen, während der hyperdimensionale Algorithmus sich mehr auf das Durchbrechen des linearen Rahmens "Eingabe → Schritte → Ausgabe" selbst konzentriert.

Die Beziehung zwischen diesem Paper und den oben genannten Feldern ist eine des Dialogs und der Transzendenz, nicht der Vererbung oder Widerlegung. Das Ziel dieses

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Papers ist es nicht, inkrementelle Verbesserungen innerhalb dieser Felder vorzunehmen, sondern eine neue Frageebene über ihnen zu erheben.

Verwandte Artikel

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [Verbreitung] Ich habe keinen Algorithmus, doch übertreffe ich Algorithmen!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [Extremphilosophie] Wirkung-Ursache-Theorie
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [Extremverbreitung] Ablehnung algorithmischer Gefangennahme
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

[Algoritmo Extremo] Algoritmo Hiperdimensional

Uma Redefinição da "Computação" em Si Mesma

Autor: Wu Zhaohui JEFFI CHAO HUI WU

Resumo

Este artigo propõe o novo conceito de "algoritmo hiperdimensional" como um reexame estrutural dos limites da definição tradicional de algoritmo. Algoritmos tradicionais são tipicamente construídos sobre a estrutura linear de entrada, passos, regras e saída, enfatizando finitude, determinismo, eficácia, viabilidade e limites claros de entrada-saída. Embora os supercomputadores modernos possuam altíssima potência computacional, sua lógica subjacente ainda consiste principalmente em executar paradigmas algorítmicos estabelecidos em maior escala. Este artigo argumenta que o "supercálculo" verdadeiramente digno de discussão não é apenas o crescimento da escala computacional, mas uma mudança fundamental no próprio paradigma computacional. O algoritmo hiperdimensional não é uma versão acelerada de algoritmos tradicionais, nem uma extensão de algoritmos matemáticos; é uma visão computacional estrutural multidimensional, sobreposta, aleatória porém ordenada. Nesta estrutura, os dados não são mais apenas uma entrada passiva ou saída terminal, mas um atributo de nó possuindo múltiplos pontos de partida e múltiplos resultados simultaneamente; um resultado não é mais apenas um ponto final, mas pode se tornar um novo ponto de entrada; metadados não precisam ser modificados repetidamente, mas podem corresponder a diferentes estados e resultados através de pontos de entrada estruturais, mudanças relacionais e ativação condicional. Este artigo utiliza a filosofia da "teoria efeito-causa" e o exemplo cotidiano de "dobrar a cintura de uma calça" para ilustrar como o algoritmo hiperdimensional pode tornar um resultado alvo diretamente válido ao mudar o ponto de entrada estrutural, reduzindo assim o cálculo repetitivo e até mesmo contornando o caminho computacional original. Este artigo visa estabelecer uma definição preliminar, limites estruturais e bases conceituais para o "algoritmo hiperdimensional", fornecendo um nó teórico original para a futura expansão da computação hiperdimensional, algoritmos extremos e um novo sistema científico. De acordo com o critério de "existência e validade constituem um novo paradigma", o algoritmo hiperdimensional, como um novo paradigma estrutural computacional já sistematicamente validado em múltiplos domínios, já é válido no presente.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Palavras-chave: Algoritmo hiperdimensional; Algoritmo extremo; Computação hiperdimensional; Teoria efeito-causa; Metadados; Paradigma computacional; Ponto de entrada estrutural; Causalidade não linear; Nova ciência

Introdução: Por que precisamos rediscutir "algoritmo"

Quando proponho o "algoritmo hiperdimensional", não é para dar aos algoritmos existentes um nome maior, nem para embrulhar algoritmos tradicionais em algum conceito novidadeiro. Pelo contrário, quero discutir uma questão mais fundamental: Será que a compreensão humana atual de "algoritmo" está confinada a uma estrutura excessivamente estreita?

Hoje, quando as pessoas falam de algoritmos, geralmente pensam em fórmulas matemáticas, código de programa, passos lógicos, entrada-saída, treinamento de modelos, processamento de dados, otimização de caminhos, classificação de buscas, recomendação automática e raciocínio de inteligência artificial. Estas são certamente formas importantes de algoritmos, e sustentam a operação dos computadores modernos, da internet, dos bancos de dados, da inteligência artificial, da supercomputação, dos sistemas industriais e da sociedade da informação. No entanto, se o algoritmo for entendido apenas como "um conjunto de passos ordenados", apenas como um processo que "começa com uma entrada, passa por um processamento baseado em regras, e termina com uma saída", então essa compreensão já confina a computação dentro de uma estrutura de baixa dimensão.

Vivemos em uma era dominada por algoritmos. Dos motores de busca aos sistemas de recomendação, da previsão do tempo à inteligência artificial, os limites dos algoritmos são os limites da inteligência humana. Mas há uma pergunta raramente feita: Os algoritmos só podem ser assim? Por que a computação deve seguir o padrão linear "entrada → passos → saída"? Por que o mesmo problema deve ser calculado do zero toda vez? Este artigo tenta desafiar esses pressupostos padrão e propor um paradigma computacional completamente diferente — o algoritmo hiperdimensional. De acordo com o critério de "existência e validade constituem um novo paradigma", o algoritmo hiperdimensional, como um novo paradigma estrutural computacional já sistematicamente validado em múltiplos domínios, já é válido no presente.

É importante notar que o "algoritmo hiperdimensional" proposto neste artigo é um conceito completamente diferente da "computação hiperdimensional" (Hyperdimensional Computing, HDC) que existe no meio acadêmico há quase trinta anos. A HDC usa vetores binários aleatórios de altíssima dimensão para codificação e operações, buscando representação e cálculo mais eficientes, mas ainda permanece

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

dentro do paradigma linear "entrada → processamento → saída". O algoritmo hiperdimensional é totalmente diferente: ele questiona se um resultado pode ser diretamente estabelecido através da estrutura, se o caminho computacional em si pode ser contornado, se os metadados podem permanecer inalterados enquanto se adaptam a múltiplos resultados. Seus nomes são semelhantes, mas suas conotações são opostas. O algoritmo hiperdimensional não é uma atualização de algoritmos, mas um re-questionamento da premissa de "se um algoritmo deve necessariamente existir".

Parte Um: A definição padrão atual de "algoritmo"

Nos sistemas dominantes da ciência da computação, matemática e engenharia, há uma definição básica de algoritmo aceita há muito tempo: um algoritmo é um processo computacional bem definido, consistindo em um número finito de passos, que recebe uma ou mais entradas, as transforma através de regras determinísticas, e produz uma ou mais saídas em um tempo finito. Este entendimento não é a opinião casual de ninguém, mas um consenso fundamental formado ao longo do longo desenvolvimento da civilização computacional moderna. Ele sustenta a lógica subjacente do software, hardware, bancos de dados, sistemas de rede, modelos de inteligência artificial e centros de supercomputação. Por mais complexo que um algoritmo pareça, seu núcleo ainda gira em torno de entrada, regras, passos e saída.

Este conceito padrão de algoritmo pode ser decomposto em várias características centrais.

Primeiro, finitude. Um algoritmo deve terminar em um número finito de passos; não pode loop infinitamente nem executar para sempre. Um processo que nunca para, mesmo que possa ser formalmente descrito, dificilmente pode ser considerado um algoritmo eficaz. Todos os programas de computação científica e fluxos de processamento de dados têm condições de término claras.

Segundo, determinismo. Dada a mesma entrada sob as mesmas condições, o algoritmo deve produzir a mesma saída. Mesmo que alguns algoritmos introduzam aleatoriedade, é tipicamente aleatoriedade controlada, aleatoriedade reproduzível, ou um mecanismo probabilístico interpretável estatisticamente, não uma aleatoriedade intrínseca completamente incontrolável. Os resultados das simulações numéricas devem ser reproduzíveis; este é um requisito fundamental da computação científica.

Terceiro, limites claros de entrada-saída. A entrada primeiro, o processamento no meio, a saída por último; o ponto de partida e o ponto final têm limites estruturais claros. Quaisquer dados que você der ao algoritmo, após o processamento, ele lhe dá

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

um resultado; este limite é distinto. Entrada primeiro, processamento depois, saída por último; a ordem não pode ser invertida.

Quarto, eficácia. Cada passo de um algoritmo deve ser uma operação praticamente executável; não pode conter saltos impossíveis de executar, descrever ou verificar. Cada passo deve ser uma operação básica que um humano poderia executar com papel e lápis, ou que um computador poderia realmente executar — adição, subtração, multiplicação, divisão, julgamentos lógicos, leitura/escrita de dados, etc.

Quinto, viabilidade. Mesmo que um algoritmo seja teoricamente válido, se o tempo, a potência computacional, a memória ou os recursos de armazenamento que ele consome são totalmente inaceitáveis, é difícil considerá-lo um algoritmo eficaz em termos de engenharia. Um algoritmo teoricamente correto mas que levaria centenas de milhões de anos para ser concluído não é considerado um algoritmo viável em engenharia.

Este conjunto completo de conceitos algorítmicos remonta, em última análise, ao modelo da máquina de Turing. Como abstração central da teoria da computação moderna, a máquina de Turing define o limite básico da "computabilidade". Por mais poderosos que sejam os computadores atuais, por mais avançados que sejam os chips, por maior que seja a escala dos centros de supercomputação, sua lógica subjacente nunca se desviou verdadeiramente deste caminho: entrada, regras, passos, saída. Os supercomputadores modernos apenas aceleram a execução deste processo através de maior escala de hardware, maior paralelismo e escalonamento de sistemas mais complexo. Ou seja, no contexto tradicional, o "supercálculo" é primariamente uma expansão da escala computacional, não necessariamente uma mudança fundamental no paradigma computacional. A potência computacional pode aumentar milhares ou dezenas de milhares de vezes, mas se a lógica subjacente permanecer "entrada, passos, saída", então ela permanece dentro do paradigma algorítmico tradicional.

Parte Dois: Definição do "algoritmo hiperdimensional"

O "algoritmo hiperdimensional" que proponho emerge precisamente deste contexto. Não é uma versão melhorada de algoritmos tradicionais, nem um algoritmo mais rápido, nem uma fórmula matemática mais complexa, nem uma técnica de programação mais avançada. É um entendimento estrutural completamente diferente.

Primeiro, precisamos explicar o significado do termo "hiperdimensional". "Hiperdimensional" tem aqui dois significados. O primeiro significado é que não se limita a uma sequência linear unidimensional de passos, mas permite que múltiplas

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

dimensões se desdobrem simultaneamente. O segundo significado é que tenta transcender os limites definicionais tradicionais da "computação" — um algoritmo não precisa mais ser necessariamente matemático, sequencial ou determinístico. Um algoritmo tradicional é como dirigir em uma via de mão única: você deve começar do ponto A, passar por B, C, D para chegar a Z. O algoritmo hiperdimensional não acredita que a computação tenha que ser uma via de mão única. Ele acredita que a computação pode ser uma rede, um campo, uma estrutura multidimensional, onde qualquer nó pode ser um ponto de entrada e qualquer nó pode ser um ponto de saída.

No meu sistema de Nova Ciência, um algoritmo não é necessariamente um algoritmo matemático, não é necessariamente uma única computação ordenada, não está necessariamente obrigado a correr ao longo de passos fixos, e não está necessariamente estritamente sujeito ao padrão linear "entrada, processamento, saída". O algoritmo hiperdimensional é uma estrutura multidimensional, sobreposta, aleatória porém ordenada. Nesta estrutura, cada dado não é nem uma entrada isolada nem uma saída fixa, mas possui múltiplos papéis simultaneamente: pode ser tanto o ponto de partida para múltiplos resultados quanto o resultado produzido pela ação conjunta de múltiplos pontos de partida.

Em outras palavras, os algoritmos tradicionais colocam os dados dentro de um fluxo de processo, enquanto o algoritmo hiperdimensional coloca os dados dentro de uma estrutura. Nos algoritmos tradicionais, os dados são geralmente divididos em dados de entrada, dados intermediários e dados de saída. Cada tipo de dado tem uma posição e um papel claros. Entrada é entrada, resultado é resultado, processo intermediário é processo intermediário. No entanto, no algoritmo hiperdimensional, um dado não tem apenas uma identidade. Pode ser um resultado em uma direção e um ponto de partida em outra; pode ser um objeto invocado em uma estrutura e se tornar um ponto de entrada que desencadeia novos resultados em outra estrutura. O dado não é mais apenas material passivo, mas um nó relacional multidimensional.

Isto corresponde precisamente ao núcleo do algoritmo hiperdimensional: cada dado é o ponto de partida para múltiplos resultados e também o resultado de múltiplos pontos de partida. Os algoritmos tradicionais usam "passos ordenados" para produzir um único resultado; o algoritmo hiperdimensional está mais próximo de uma estrutura de estado multidimensional onde, na superposição de aleatoriedade e ordem, o resultado não é calculado para existir, mas se manifesta naturalmente dentro da estrutura. Neste sistema, o dado é tanto o ponto de partida quanto um componente do resultado.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Para apresentar as características centrais do algoritmo hiperdimensional mais claramente, eu as decomponho nos seguintes cinco aspectos.

Primeiro, natureza não matemática. Os algoritmos dominantes são essencialmente matemáticos — podem ser completamente descritos usando lógica simbólica e fórmulas matemáticas. O algoritmo hiperdimensional não é necessariamente matemático. Pode ser baseado em princípios físicos, relações geométricas, um novo paradigma da teoria da informação, ou mesmo princípios da consciência. A computação não precisa ser necessariamente realizada através de "operações matemáticas"; pode se apresentar naturalmente através de relações geométricas no espaço de maior dimensão. Por exemplo, a forma de uma bolha de sabão é determinada pelo princípio da minimização da tensão superficial. Isto não é um "algoritmo matemático" calculando — é a própria física "calculando". O algoritmo hiperdimensional tenta entender este tipo de "computação", em vez de usar fórmulas matemáticas para simulá-la.

Segundo, multidimensionalidade e superposição. Os algoritmos tradicionais executam passos ordenados em uma única dimensão, com apenas um estado em cada passo. O algoritmo hiperdimensional permite que múltiplas dimensões existam simultaneamente, com múltiplos estados se superpondo e coexistindo. O algoritmo não segue um único caminho; em vez disso, se desdobra dentro de um campo de estado multidimensional. O resultado não é "calculado", mas "se manifesta" a partir deste campo. Por exemplo, quando um floco de neve se forma no ar, não há um "algoritmo" dizendo a cada molécula de água para onde ir. O arranjo das moléculas de água é o resultado de múltiplas dimensões — temperatura, umidade, fluxo de ar — agindo juntas. A forma do floco de neve "se manifesta" em vez de ser "calculada".

Terceiro, aleatoriedade dentro da ordem. Nos algoritmos tradicionais, a aleatoriedade é instrumental, externa, uma perturbação controlável; o algoritmo em si é determinístico. No algoritmo hiperdimensional, a aleatoriedade é intrínseca e estrutural. A aleatoriedade e a ordem coexistem e trabalham juntas, constituindo conjuntamente a essência do algoritmo. Isto não é "um algoritmo com aleatoriedade", mas "a aleatoriedade em si é um componente orgânico do algoritmo". Por exemplo, a estrutura ecológica de uma floresta — a distribuição das árvores parece aleatória, mas no geral apresenta uma densidade ordenada e relações entre espécies. Essa "aleatoriedade" não é um bug, mas um pré-requisito para o funcionamento normal da estrutura ecológica.

Quarto, os múltiplos papéis dos dados. Nos algoritmos tradicionais, o papel dos dados é singular — entrada é entrada, saída é saída, resultado intermediário é resultado intermediário. No algoritmo hiperdimensional, cada dado é simultaneamente o ponto

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

de partida para múltiplos resultados e o resultado de múltiplos pontos de partida. Um nó de dados pode se desdobrar a jusante em múltiplos caminhos de resultados, e também pode ser convergido a montante por múltiplas causas. O dado não é mais um ponto em um fluxo linear, mas um nó de convergência em uma rede. Por exemplo, considere a altura de uma pessoa. Em um algoritmo tradicional, a altura é uma "entrada" — você a insere para obter saídas como previsão de peso, tamanho de roupa, etc. Mas a altura também é um "resultado" — é determinada conjuntamente por múltiplos "pontos de partida" como genética, nutrição e exercício. No algoritmo hiperdimensional, o dado de altura é simultaneamente um ponto de partida e um resultado, não uma escolha binária.

Quinto, metadados inalterados, relações variáveis. Ao enfrentar diferentes problemas, os algoritmos tradicionais ou modificam os dados em si ou recalculam tudo. O algoritmo hiperdimensional não modifica metadados — os atributos essenciais dos dados permanecem inalterados. O que muda são os pontos de entrada, os modos de interpretação e as relações entre os dados. A mesma estrutura de metadados, ativada através de diferentes pontos de entrada, pode apresentar resultados completamente diferentes. Isto significa: calcule uma vez, use infinitas vezes. Por exemplo, considere a mesma passagem de texto. Você pode lê-la como poesia, decodificá-la como uma cifra, ou analisá-la como um documento histórico. O texto em si não mudou — você apenas mudou o "ponto de entrada". O algoritmo hiperdimensional busca precisamente esta capacidade: "mesmos dados, múltiplos pontos de entrada, múltiplos resultados."

Da perspectiva da estrutura de dados, os próprios metadados não são modificados; em vez disso, através de diferentes pontos de entrada e condições, eles correspondem a múltiplos pontos de partida e resultados dentro da mesma estrutura. Os dados não são mais processados repetidamente, mas, mantendo sua estrutura inalterada, são ativados através de diferentes pontos de entrada, apresentando assim diferentes resultados. A abordagem tradicional é "processar dados para diferentes resultados", enquanto o algoritmo hiperdimensional é "usar os mesmos dados para carregar múltiplos resultados potenciais".

Parte Três: Diferenças fundamentais entre o algoritmo hiperdimensional e os algoritmos tradicionais

Com base nas definições acima, o algoritmo hiperdimensional e os algoritmos tradicionais exibem diferenças fundamentais em múltiplas dimensões centrais. Estas são detalhadas abaixo uma a uma.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Em termos de natureza essencial, os algoritmos tradicionais são matemáticos e formais, capazes de descrição completa usando lógica simbólica e fórmulas matemáticas; são essencialmente objetos matemáticos. O algoritmo hiperdimensional não é necessariamente matemático; pode ser baseado em princípios físicos, geométricos, informacionais ou mesmo conscientes. Não é um subconjunto da matemática, mas uma categoria mais ampla. Esta diferença pode ser resumida como: do "objeto matemático" à "lei universal".

Em termos de ordem temporal, os algoritmos tradicionais seguem uma ordem temporal singular, ordenada e linear, passo A para B para C, estritamente sequencial, ou decomponível em sub-passos ordenados paralelos; a seta do tempo é irreversível. O algoritmo hiperdimensional não tem sequência de passos fixa; é multidimensional, sobreposto, aleatório porém ordenado. O resultado pode não estar relacionado à "ordem" de execução, porque não existe "ordem" tradicional em primeiro lugar. Esta diferença pode ser resumida como: do "tempo linear" à "simultaneidade de dimensão superior".

Em termos de causalidade, os algoritmos tradicionais seguem uma cadeia causal determinística: dada a mesma entrada e o mesmo estado inicial, a saída é sempre única; a causa precede o efeito, uma seta unidirecional irreversível. O algoritmo hiperdimensional segue uma causalidade superposta: cada dado é o ponto de partida para múltiplos resultados e o resultado de múltiplos pontos de partida. Esta é a minha filosofia da "teoria efeito-causa" — é possível ter primeiro o resultado, depois a causa. O resultado não é um ponto final, mas pode se tornar um novo ponto de partida. Esta diferença pode ser resumida como: da "causa única, efeito único" à "rede causal totalmente interconectada".

Em termos de estrutura de dados, os algoritmos tradicionais seguem uma estrutura de fluxo unidirecional da entrada para o processamento e depois para a saída; os dados têm um papel singular com limites claros; os dados de entrada permanecem entradas do início ao fim, os dados de resultado permanecem resultados. No algoritmo hiperdimensional, os dados têm múltiplos papéis; os mesmos dados são simultaneamente um resultado e um ponto de partida; não há pontos de partida absolutos nem pontos finais absolutos, apenas nós em uma rede. Esta diferença pode ser resumida como: da "fluidez unidirecional" à "simbiose recursiva".

Em termos de modo computacional, os algoritmos tradicionais calculam do zero cada vez que encontram um problema, mesmo que tenham calculado problemas semelhantes mil vezes antes, mesmo que a resposta já seja conhecida; na milésima primeira vez, ainda percorrem todo o processo. Isto é computação "sem estado". No algoritmo hiperdimensional, se existir um resultado correspondente, não há

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

necessidade de percorrê-lo novamente; a partir de um resultado, múltiplos pontos de partida podem ser inferidos, e caminhos causais podem ser desdobrados em reverso. A computação é com estado, tem memória e é reutilizável. Esta diferença pode ser resumida como: de "recalcular cada vez" a "reutilizar uma vez".

Em termos do papel da aleatoriedade, a aleatoriedade nos algoritmos tradicionais é pseudoaleatória ou injetada externamente; números aleatórios são meras ferramentas para amostragem, aproximação ou otimização; o algoritmo em si é determinístico. A aleatoriedade no algoritmo hiperdimensional é intrínseca e constitutiva; a aleatoriedade é um componente orgânico, coexistindo e cooperando com a ordem. Não é "um algoritmo contém aleatoriedade", mas "a aleatoriedade é uma razão pela qual o algoritmo pode funcionar". Esta diferença pode ser resumida como: da "aleatoriedade instrumental" à "aleatoriedade constitutiva".

Em termos da compreensão dos recursos, os algoritmos tradicionais buscam computação mais rápida e precisa dados os recursos; os recursos são restrições; o objetivo do algoritmo é "completar a computação dentro das restrições". O algoritmo hiperdimensional busca tornar a própria computação desnecessária através do design estrutural; não "computar mais rápido", mas "contornar a computação"; os recursos não são usados para serem "consumidos", mas para "projetar pontos de entrada". Esta diferença pode ser resumida como: da "otimização sob restrições de recursos" à "eliminação através do design estrutural".

Parte Quatro: A filosofia da teoria efeito-causa

Com base na comparação acima, preciso desenvolver ainda mais a filosofia da "teoria efeito-causa". Esta é uma das ideias fundamentais centrais do algoritmo hiperdimensional.

O pensamento tradicional normalmente sustenta que a causa precede o efeito; primeiro a causa, depois o efeito; primeiro a entrada, depois a saída. Este conceito se manifesta em algoritmos como: você deve começar do ponto de partida e caminhar passo a passo até o ponto final. Mesmo que você saiba qual é a resposta, você não pode diretamente "usar" essa resposta — porque você não tem o "caminho de prova".

Na minha estrutura, um resultado não é necessariamente apenas um ponto final. Uma vez que um resultado aparece, ele pode se tornar um novo ponto de partida e continuar participando da geração de novas estruturas. Um resultado pode participar retroativamente da formação de causas. Múltiplos pontos de partida possíveis podem ser inferidos a partir de um resultado. Causa e efeito não estão mais dispostos em uma

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

sequência unidirecional, mas formam relações cíclicas, relações de rede, relações multidimensionais.

Em outras palavras, os algoritmos tradicionais perguntam: "Dada a causa, como obtemos o resultado?" O algoritmo hiperdimensional pergunta: "Dado o resultado, como inferimos ou construímos a causa?"

É importante esclarecer que "ter primeiro o resultado, depois a causa" não é uma negação da causalidade, nem afirma que "resultados surgem do nada". Em vez disso, ilustra que, em estruturas de dimensão superior, causa e efeito podem servir como pontos de entrada um para o outro e se transformar mutuamente. Como no exemplo da calça a seguir, o resultado "vestível e sem arrastar no chão" é determinado primeiro, e então eu encontro retroativamente o ponto de operação de "dobrar a cintura", que corresponde ao nível da "causa" dentro da estrutura. Não é que o resultado não tenha causa, mas que o resultado se torna um novo ponto de entrada para descobrir a causa.

Uma suposição fundamental no mundo dos algoritmos tradicionais é que o tempo é unidirecional; você deve calcular passo a passo do estado inicial para o estado final. Mesmo que você saiba qual é a resposta, você não pode diretamente "usar" essa resposta, porque você não tem o caminho de prova. O algoritmo hiperdimensional desafia precisamente esta suposição: se o resultado é conhecido, as causas podem ser inferidas para trás; o mesmo resultado pode corresponder a múltiplos caminhos causais; a computação não é mais caminhar do início ao fim, mas desdobrar todos os pontos de partida possíveis a partir do fim.

Parte Cinco: Um exemplo cotidiano — calça e dobrar a cintura

Para entender mais intuitivamente as diferenças abstratas acima, usarei um exemplo cotidiano.

Uma pessoa compra uma calça nova com cós elástico. As pernas da calça estão um pouco compridas demais, arrastando no chão, o que é inconveniente.

A abordagem tradicional: notando que as pernas estão compridas demais, dobra-se uma seção da perna, costura-se com agulha e linha, e então experimenta-se. Se a criança crescer e a calça ficar muito curta, a perna costurada não pode ser solta, e a calça está arruinada. Na próxima vez, compra-se uma calça nova e costura-se novamente. Este método parece muito natural, porque o problema parece estar nas pernas da calça, então você aborda as pernas da calça. Corresponde ao pensamento dos algoritmos tradicionais: identificar o problema, localizar a manifestação, processar

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

no local da manifestação, e finalmente obter um resultado. Pernas da calça muito compridas? Então altere as pernas da calça. Dados incorretos? Então modifique os dados. Caminho inadequado? Então continue remendando ao longo do caminho.

Minha abordagem é diferente. Em vez de alterar a barra da calça, dobro a cintura uma vez, e a calça fica imediatamente vestível. Esta ação é muito simples, mas a lógica estrutural por trás dela é completamente diferente. Não corto as pernas da calça, não costuro a barra permanentemente, não mudo o comprimento original da calça e não danifico os metadados da calça. A circunferência da cintura, o comprimento da calça, o modelo e o tecido não sofrem nenhuma mudança essencial. Simplesmente mudo o ponto de entrada estrutural da cintura, fazendo com que as pernas da calça se encurtem naturalmente ao usar. A cintura não é aqui uma mera localidade, mas um ponto de controle global para todo o estado de uso. Ao ajustar este ponto de controle, o problema a jusante desaparece diretamente.

Mais importante, este método é reversível, ajustável e reutilizável. Isto é especialmente evidente para uma criança em crescimento. Se a criança não é alta o suficiente e as pernas da calça estão um pouco compridas, dobro a cintura uma vez; depois de algum tempo, quando a criança cresceu, solto a cintura um pouco; quando cresceu ainda mais, solto completamente. A estrutura original da calça permanece intacta, no entanto pode se adaptar à altura da criança em diferentes estágios. Com o método tradicional, se você costurar a barra permanentemente, quando a criança crescer, a calça pode ficar muito curta; se você cortar, é ainda mais impossível restaurar. Meu método mantém os metadados inalterados, permitindo que a mesma estrutura se adapte a múltiplos estados.

Este exemplo revela várias diferenças estruturais importantes.

A abordagem tradicional corresponde à lógica dos algoritmos tradicionais: o problema está nas pernas da calça, então as pernas da calça devem ser modificadas, seguindo o caminho "causa para efeito" passo a passo, sem pular nenhum passo. Resolver um problema de cada vez, irreversivelmente. Na próxima vez que o mesmo problema ocorrer, fazer tudo de novo. Minha abordagem corresponde à lógica do algoritmo hiperdimensional: o objetivo é não arrastar. Não resolvo a causa superficial de "pernas da calça muito compridas"; em vez disso, encontro um ponto de controle global — a cintura. Dobrar a cintura uma vez, as pernas da calça encurtam automaticamente, e o problema desaparece instantaneamente. Não modifico nenhum metadado. A dobra na cintura é apenas um estado temporário, reversível e dinâmico.

Da perspectiva dos dados, a abordagem tradicional modifica metadados — costurar as pernas da calça mais curtas perde permanentemente os dados originais. Minha

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

abordagem não modifica nenhum metadado; as dimensões originais da calça permanecem intactas, apenas adicionando um estado temporário, reversível e dinâmico — a dobra. A dobra na cintura não cria novos dados nem destrói dados antigos; apenas rearranja as relações entre os dados — encurtando a circunferência efetiva da cintura, mudando indiretamente o comprimento efetivo das pernas da calça.

Neste exemplo, "as pernas da calça não arrastam no chão" é o resultado alvo. O método tradicional parte da causa "pernas da calça muito compridas", modifica a barra, e finalmente obtém o resultado "não arrastar". Meu método, pelo contrário, primeiro clarifica o resultado "não arrastar", depois busca para trás um ponto de entrada estrutural, e finalmente descobre que dobrar a cintura uma vez torna o resultado válido. Esta é a manifestação prática da "teoria efeito-causa". Não é necessário caminhar passo a passo da causa para o efeito; pode-se determinar a estrutura para trás a partir do efeito, tornando o caminho original desnecessário.

Este exemplo também aborda a questão dos "metadados". O que são metadados? No exemplo da calça, os metadados são as dimensões originais da calça: circunferência da cintura, comprimento, modelo, tecido. Estes são os "atributos essenciais" da calça. O dado temporário é a dobra na cintura; não altera nenhuma das dimensões originais da calça, apenas dobra uma seção temporariamente. A abordagem tradicional modifica metadados — o comprimento da calça é encurtado permanentemente, os dados originais são perdidos. Minha abordagem não modifica nenhum metadado — as dimensões originais da calça permanecem intactas, apenas se adiciona um estado temporário, reversível e dinâmico.

As dimensões originais da cintura, como metadados, são simultaneamente o "ponto de partida para múltiplos resultados": elas suportam simultaneamente múltiplos estados de uso, como usar normalmente, usar com uma dobra na cintura, usar com duas dobras na cintura, etc. Elas também são "o resultado de múltiplos pontos de partida": são determinadas conjuntamente por múltiplos pontos de partida como a seleção do tecido, o método de corte, a intenção do design, etc. A dobra na cintura, como dado temporário, não cria novos dados nem destrói dados antigos; apenas rearranja as relações entre os dados.

Isto não é um truque comum, mas um tipo de pensamento estrutural. Muitas pessoas, ao ver pernas de calça muito compridas, são atraídas pela manifestação das pernas, então todo o processamento gira em torno das pernas. Mas se visto da estrutura global, o fato de as pernas estarem muito compridas é meramente uma manifestação a jusante; uma mudança na posição da cintura pode afetar o posicionamento efetivo de toda a calça. Ou seja, o problema não precisa necessariamente ser resolvido onde aparece. Muitos problemas de baixa dimensão têm seu verdadeiro ponto de controle

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

em um nível estrutural mais alto. Os algoritmos tradicionais tendem a continuar calculando ao longo da superfície do problema, enquanto o algoritmo hiperdimensional busca o ponto de entrada estrutural.

Isto também explica por que o algoritmo hiperdimensional não é mais complexo, mas pode ser mais simples. Uma estrutura verdadeiramente de alto nível muitas vezes não complica o problema, mas torna o problema complexo simples no ponto de entrada correto. Diante de um problema complexo, os algoritmos tradicionais podem adicionar passos, adicionar modelos, adicionar potência computacional, adicionar armazenamento, adicionar tempo de treinamento. O algoritmo hiperdimensional, no entanto, busca um nó estrutural; uma vez que este nó é ativado corretamente, o caminho originalmente complexo pode se tornar inválido. O chamado "contornar o caminho" não é preguiça, mas redefinir se o caminho é necessário.

Neste exemplo: a abordagem tradicional é "modificar o resultado ao longo do caminho", enquanto o algoritmo hiperdimensional é "mudar o ponto de entrada, tornando todo o caminho inválido". O método convencional remenda continuamente na extremidade do resultado, enquanto o outro método muda diretamente o ponto de entrada estrutural, fazendo com que problemas que de outra forma exigiriam múltiplos tratamentos sejam reestruturados no ponto de partida de uma só vez. A forma dominante dos algoritmos tradicionais é "começar do ponto de partida, seguir o caminho para obter o resultado", enquanto no algoritmo hiperdimensional, "o próprio resultado pode se tornar um ponto de entrada de caminho e participar da geração de novas estruturas". A abordagem tradicional é "uma estrutura corresponde a um resultado", enquanto o algoritmo hiperdimensional é "uma estrutura carrega múltiplos estados de resultado".

Parte Seis: Uma visão de dados de não modificar metadados

Da perspectiva dos dados, o algoritmo hiperdimensional tem um princípio muito importante: não modificar metadados, para que possam ser aplicados a múltiplos pontos de partida ou resultados.

Metadados são os atributos originais, a estrutura básica e a informação ontológica dos dados. Diante de diferentes problemas, os métodos de processamento tradicionais frequentemente modificam os dados, copiam os dados, processam os dados, recalculam os dados ou estabelecem caminhos diferentes para resultados diferentes. Esta abordagem certamente pode resolver problemas, mas ao custo de os dados serem continuamente processados, os caminhos serem continuamente repetidos e a complexidade do sistema aumentar constantemente.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

O algoritmo hiperdimensional, pelo contrário, enfatiza que, enquanto se tenta não modificar os metadados, mudando os pontos de entrada, as relações, as condições e os modos de ativação, a mesma estrutura de metadados pode corresponder a múltiplos pontos de partida e múltiplos resultados. A ontologia dos dados permanece inalterada; as relações mudam. A estrutura básica permanece inalterada; o modo de apresentação muda. Os metadados permanecem intactos, no entanto podem se adaptar a diferentes estados.

Isto é o que chamo de "calcule uma vez, use infinitas vezes". "Calcule uma vez" não é simplesmente armazenar em cache um resultado, mas significa que, uma vez que uma estrutura é estabelecida, ela não precisa mais reconstruir um caminho completo para cada estado. A mesma estrutura de dados, ativada através de diferentes pontos de entrada, pode apresentar diferentes resultados. Não é "processar dados para diferentes resultados", mas "usar os mesmos dados para carregar múltiplos resultados potenciais". Este é também um dos núcleos que distinguem o algoritmo hiperdimensional dos algoritmos tradicionais. Os algoritmos tradicionais processam dados ao longo de um caminho; o algoritmo hiperdimensional processa a estrutura global de dados, pontos de entrada, relações e resultados.

Na sua estrutura mínima, o algoritmo hiperdimensional pode ser entendido como: um sistema que, sem modificar metadados, permite que o mesmo nó de dados corresponda a múltiplos caminhos de resultado através de mudanças nos pontos de entrada estruturais. Esta definição não busca contrair imediatamente o algoritmo hiperdimensional em uma fórmula matemática tradicional, mas primeiro estabelecer seus limites estruturais. Não é uma função linear única, nem uma fórmula fixa, nem um simples mecanismo de cache. É uma estrutura relacional: os metadados permanecem estáveis, os pontos de entrada podem mudar, as condições podem mudar, as relações podem mudar, os resultados podem ser apresentados em múltiplas direções. É precisamente nesta estrutura que a computação não é mais sobre executar passos, mas sobre ativar relações.

Da perspectiva dos dados, a essência de "não modificar metadados, tornando-os aplicáveis a múltiplos pontos de partida ou resultados" é: a ontologia dos dados permanece inalterada; a mudança ocorre no "modo de interpretação / ponto de entrada de uso". A estrutura tradicional é: quando o problema muda, os dados ou o caminho mudam. No algoritmo hiperdimensional: quando o problema muda, a estrutura interpretativa muda, não os dados.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Parte Sete: Redefinindo "supercomputação"

Neste sistema, preciso clarificar um termo — "supercomputação".

O que quero dizer com "supercomputação" não são os supercomputadores no sentido usual. Não mais chips, maior potência computacional ou maiores centros de dados. O que quero dizer com "supercomputação" é o "algoritmo hiperdimensional".

A verdadeira supercomputação não é necessariamente o acúmulo de potência computacional, mas possivelmente uma mudança no paradigma computacional. Quando um sistema ainda depende de computação repetitiva, por mais poderoso que seja, permanece preso dentro de caminhos. Quando um sistema pode reduzir estruturalmente a computação, contornar caminhos repetitivos e tornar os resultados novos pontos de entrada, ele começa a se aproximar da verdadeira computação hiperdimensional.

Em outras palavras, a supercomputação tradicional busca "usar mais potência computacional para resolver problemas maiores". O algoritmo hiperdimensional busca "usar melhor estrutura para que os problemas não precisem mais de tamanha potência computacional enorme". Estes dois não são contraditórios, mas suas direções diferem.

Se a supercomputação tradicional representa o limite da potência computacional, então o algoritmo hiperdimensional representa uma virada na compreensão da computação. A verdadeira "supercomputação" pode não ser mais chips, maiores centros de dados, maior consumo de energia ou modelos mais complexos. A verdadeira "supercomputação" pode ser a descoberta de um ponto de entrada estrutural, uma eliminação de um caminho, um desdobramento reverso a partir de um nó de resultado, uma maneira de tornar múltiplos resultados simultaneamente válidos sem modificar metadados. Quanto mais poderosa a potência computacional, se ainda estiver presa em caminhos de baixa dimensão, permanece poderosa apenas dentro desses caminhos; uma vez que a estrutura é elevada, mesmo com operações extremamente simples, um nível mais alto de eficiência pode emergir.

A chamada supercomputação é o limite da potência computacional; o algoritmo hiperdimensional é uma redefinição da premissa de "se a computação deve depender da potência computacional".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Parte Oito: Fundamentos empíricos — Validação por sistemas multi-domínio e critérios de novo paradigma

O algoritmo hiperdimensional não é uma construção puramente teórica. Já foi validado em vários sistemas do mundo real.

Meu sistema logístico não requer potência de supercomputação nem suporte em nuvem; realiza um agendamento complexo com recursos relativamente baixos. Os resultados podem ser reutilizados, a estrutura pode ser repetidamente adaptada, e não há necessidade de recalcular tudo do zero cada vez. Não depender da supercomputação, não depender da nuvem, realizar um agendamento complexo com baixos recursos — este é um avanço estabelecido e valioso em engenharia. Demonstra que a alta potência computacional não é a única solução; o design estrutural pode substituir uma parte da potência computacional.

Meu sistema editorial adota igualmente esta lógica estrutural, com uma eficiência que supera em muito qualquer sistema atual. Meu sistema extremo de geração de páginas web também é baseado no mesmo algoritmo hiperdimensional, provando repetidamente em múltiplos domínios que a mesma estrutura pode ser estavelmente válida. Os funcionários da minha empresa já estão usando estes sistemas, demonstrando que esta estrutura pode operar sem minha intervenção pessoal contínua.

A característica comum destes sistemas é: não dependem dos caminhos dominantes de alta potência computacional, não seguem passos de computação fixos, e tornam o resultado alvo diretamente válido através de ajustes nos pontos de entrada estruturais. Não são sucessos únicos, nem truques pontuais, mas o aparecimento repetido da mesma lógica estrutural em diferentes domínios.

Isto demonstra que o algoritmo hiperdimensional não é acidental, mas um método estrutural já validado por múltiplos sistemas. Não é um "truque pessoal", mas um paradigma computacional que é estavelmente válido em múltiplos domínios como logística, editoração e geração de páginas web.

De acordo com o critério de "existência e validade constituem um novo paradigma", o algoritmo hiperdimensional, como um novo paradigma estrutural computacional já sistematicamente validado em múltiplos domínios, já é válido no presente. Não estou propondo uma hipótese, mas uma estrutura computacional diferente já operacional em múltiplos sistemas. Não preciso provar que é compreendido; já provei que é

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

estavelmente válido em diferentes sistemas. No presente, dentro do escopo da minha aplicação prática, esta estrutura já pode ser considerada um novo paradigma.

Quanto aos critérios para julgar o algoritmo hiperdimensional como um novo paradigma: quando uma estrutura é logicamente autoconsistente e estavelmente válida em múltiplos sistemas, ao mesmo tempo que possui diferenças irredutíveis com os métodos existentes, ela já possui a base de um novo paradigma. A autoconsistência é o ponto de partida, a evidência empírica é o fundamento, e a diferença estrutural é o núcleo do paradigma. Sob o padrão de "existência constitui validade", este sistema já é um novo paradigma; se o mundo exterior o reconhece afeta apenas a disseminação, não a validade. Este é um paradigma estrutural computacional já estabelecido em sistemas reais, cuja validade não depende de reconhecimento ou consenso externo.

A diferença fundamental entre o algoritmo hiperdimensional e os paradigmas computacionais dominantes é: os algoritmos tradicionais dependem principalmente do cálculo para frente; uma vez que um resultado é produzido, geralmente não pode participar retroativamente de novas estruturas inferenciais. No algoritmo hiperdimensional, o resultado não é mais um ponto final, mas um nó estrutural reutilizável. Sob certas condições, o resultado pode participar de novos caminhos inferenciais, reduzindo assim o cálculo repetitivo e formando uma rede computacional com múltiplos pontos de partida e múltiplos caminhos. Nas estruturas computacionais tradicionais, o resultado é tipicamente um ponto final; na estrutura do algoritmo hiperdimensional, o próprio resultado pode se tornar um novo ponto de partida, formando assim uma rede inferencial de múltiplos caminhos. O algoritmo hiperdimensional não é uma atualização de algoritmos, mas um re-questionamento da premissa de "se um algoritmo deve necessariamente existir".

Parte Nove: Esclarecimento de mal-entendidos comuns

Com base em intercâmbios preliminares com leitores de diferentes campos, antecipo os seguintes seis mal-entendidos e os esclareço antecipadamente, para evitar que o conceito seja prematuramente forçado em estruturas inadequadas durante a disseminação.

Mal-entendido Um: O algoritmo hiperdimensional não é apenas programação dinâmica ou cache? Esclarecimento: A programação dinâmica e o cache reutilizam resultados sob "entradas idênticas". O algoritmo hiperdimensional permite inferir diferentes pontos de partida a partir de um único resultado — esta é uma diferença essencial. O cache resolve o recálculo do mesmo problema; o algoritmo

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

hiperdimensional resolve o desdobramento de novos problemas através de estruturas de resultado.

Mal-entendido Dois: Você diz "aleatório porém ordenado" — isso não é contraditório? Esclarecimento: Aleatoriedade e ordem podem coexistir. A distribuição das árvores em uma floresta é aleatória, mas a densidade geral e a estrutura de espécies são ordenadas. A aleatoriedade não é caos, mas uma maneira de organizar a estrutura. A aleatoriedade no algoritmo hiperdimensional é intrínseca e constitutiva, funcionando sinergicamente com a ordem.

Mal-entendido Três: Sem entrada e saída, como verificar a correção do resultado? Esclarecimento: O algoritmo hiperdimensional não rejeita entrada e saída; em vez disso, sustenta que a computação não tem que operar no modo linear de entrada-saída. No modo de manifestação estrutural, a "validade" é determinada pela consistência estrutural, não pela correspondência entrada-saída. Isto não abandona a verificação, mas propõe uma estrutura de verificação diferente da tradicional.

Mal-entendido Quatro: Isto não é apenas teoria da complexidade ou teoria do caos? Esclarecimento: A teoria da complexidade e a teoria do caos descrevem "sistemas difíceis de prever". O algoritmo hiperdimensional tenta descrever "sistemas que podem ser compreendidos e guiados através de pontos de entrada estruturais" — não abandonar a previsão, mas mudar o modo de previsão. A teoria do caos diz "a previsão é difícil"; o algoritmo hiperdimensional pergunta "se se pode tornar a previsão desnecessária mudando o ponto de entrada".

Mal-entendido Cinco: Você diz "não modificar metadados", mas a dobra na cintura não é também uma modificação? Esclarecimento: A dobra na cintura é um estado temporário, não uma modificação permanente dos parâmetros originais. Os metadados (circunferência da cintura, comprimento, modelo, tecido) não mudam absolutamente nada. Esta é a diferença entre "superposição de estados" e "modificação de atributos". A dobra é reversível, temporária e não muda atributos essenciais.

Mal-entendido Seis: O algoritmo hiperdimensional nega o valor dos algoritmos tradicionais? Esclarecimento: Absolutamente não. Os algoritmos tradicionais foram extremamente importantes na história tecnológica humana. Sem os algoritmos tradicionais, não haveria computadores modernos, bancos de dados, sistemas de comunicação, inteligência artificial, simulação em engenharia ou supercomputação. O valor dos algoritmos tradicionais não pode ser negado. No entanto, reconhecer o valor dos algoritmos tradicionais não significa reconhecê-los como o ponto final dos algoritmos. Os algoritmos tradicionais resolvem problemas de eficiência dentro de um paradigma computacional dado, enquanto o algoritmo hiperdimensional levanta a

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

questão do próprio paradigma computacional. Um trata de como andar melhor; o outro pergunta se é preciso tomar esse caminho.

Parte Dez: O algoritmo hiperdimensional e a história da computação humana

Para ajudar os leitores a posicionar melhor o algoritmo hiperdimensional dentro da história da computação humana, ofereço aqui um breve resumo macroscópico.

A compreensão humana da "computação" passou por múltiplos estágios. O primeiro estágio foi a computação manual; os algoritmos existiam como passos humanos, lentos e propensos a erros, mas através deste processo, os humanos compreenderam as regras básicas da computação. O segundo estágio foi a computação mecânica, da calculadora de Pascal ao Motor Analítico de Babbage; a computação foi mecanizada, mas os algoritmos ainda dependiam de estruturas físicas. O terceiro estágio foi a computação eletrônica e o paradigma de Turing; a arquitetura de von Neumann tornou-se difundida; os algoritmos foram codificados como programas; a máquina de Turing tornou-se o limite da computabilidade. O quarto estágio é a computação paralela e a supercomputação, acumulando potência computacional através de um grande número de unidades de computação para enfrentar problemas mais complexos, mas a lógica subjacente continua sendo uma extensão do paradigma de Turing.

Atualmente, a humanidade está na entrada do quinto estágio. A computação quântica tenta romper as limitações dos bits clássicos; a computação neuromórfica tenta imitar a estrutura dos sistemas nervosos biológicos; e o algoritmo hiperdimensional tenta romper a própria estrutura linear "entrada → passos → saída".

A relação entre o algoritmo hiperdimensional e os algoritmos tradicionais não é de substituição, mas de hierarquia. Os algoritmos tradicionais resolvem o problema de "como computar mais eficientemente em um determinado caminho". O algoritmo hiperdimensional pergunta "se o caminho pode ser redefinido, ou mesmo contornado". Se considerarmos a história da computação humana como um processo de constante ruptura de seus próprios limites, então o algoritmo hiperdimensional é precisamente um novo nó neste processo. Ele não resolve problemas dentro da velha estrutura, mas propõe uma nova estrutura.

Este julgamento não implica que o algoritmo hiperdimensional já esteja maduro ou completo. Muito pelo contrário. Como um novo conceito, sua definição, limites, métodos de verificação e cenários de aplicação ainda estão em processo de formação.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

O propósito deste artigo é precisamente ancorar a origem deste conceito, fornecendo uma base teórica estável para o desenvolvimento subsequente.

Parte Onze: Conclusão — Isto não é otimização, mas redefinição

A diferença entre o algoritmo hiperdimensional e os algoritmos tradicionais não é uma diferença de "melhor" versus "pior", nem uma diferença de "mais rápido" versus "mais lento", mas uma diferença fundamental no nível do paradigma.

Os algoritmos tradicionais buscam controle previsível. Você me dá a entrada, eu sei qual saída você obterá, porque calculei cada passo. É o paradigma do engenheiro: projetar, construir, testar, verificar.

O algoritmo hiperdimensional descreve uma emergência compreensível. Não posso prever precisamente cada estado intermediário, mas sei que o padrão geral se apresentará de alguma maneira ordenada. Cada dado está vivo e tem uma "perspectiva". É mais como o paradigma de um ecologista ou teórico de sistemas complexos: observar, compreender, guiar, utilizar a ordem emergente.

Os algoritmos tradicionais modificam gradualmente o resultado ao longo de um determinado caminho. O algoritmo hiperdimensional torna o resultado alvo imediatamente válido mudando o ponto de entrada estrutural, contornando assim todo o caminho.

Os algoritmos tradicionais "processam dados para diferentes resultados". O algoritmo hiperdimensional é "usar os mesmos dados para carregar múltiplos resultados potenciais".

Os algoritmos tradicionais buscam "calcular corretamente uma vez". O algoritmo hiperdimensional busca "calcular corretamente uma vez, e depois nunca mais precisar calcular".

Isto não é uma otimização de algoritmos, mas um re-questionamento da premissa de "se um algoritmo deve necessariamente existir".

Portanto, o algoritmo hiperdimensional não é uma otimização de algoritmo, mas uma redefinição de algoritmo. Não se trata de correr mais rápido na pista tradicional de algoritmos, mas de perguntar se existe outro caminho fora da pista tradicional, ou mesmo se se pode ser independente do próprio caminho. Não se trata de tentar

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

provar que os algoritmos tradicionais são inúteis, mas de apontar que os algoritmos tradicionais são apenas uma forma de baixa dimensão de algoritmo. Não pergunta "como obter resultados em menos tempo", mas "se um resultado pode ser diretamente estabelecido através da estrutura". Não pergunta "como processar mais dados", mas "se os mesmos dados podem carregar mais resultados potenciais".

Dentro do paradigma computacional dominante, o algoritmo hiperdimensional pode ser visto como incalculável, inverificável, ou difícil de incorporar dentro dos limites existentes da teoria da computação. Mas isto constitui precisamente sua diferença paradigmática. O fato de um novo conceito aparecer instável dentro de um velho paradigma não significa necessariamente que seja sem sentido; pode também significar que o velho paradigma em si não pode acomodá-lo completamente. No presente, o algoritmo hiperdimensional é antes de tudo uma proposição estrutural, uma definição original de uma nova visão computacional, e não um sistema maduro com um ciclo de engenharia fechado. Ele requer continuação subsequente através de acréscimos, desenvolvimentos, verificações e aplicações, mas sua origem conceitual deve primeiro ser claramente articulada.

Em última análise, para onde o algoritmo hiperdimensional aponta é uma nova visão da computação: um algoritmo não é necessariamente apenas matemática, não é necessariamente apenas passos, não é necessariamente apenas um programa, não é necessariamente apenas um processo de processamento entre entrada e saída. Um algoritmo também pode ser uma maneira de organizar relações multidimensionais, uma rede estrutural de dados, pontos de entrada, relações, estados e resultados. Pode conter ordem e aleatoriedade; pode ir da causa ao efeito, ou gerar retroativamente causas a partir de efeitos; pode adaptar-se a múltiplos estados sem modificar metadados; pode fazer de um resultado um novo ponto de partida, ou fazer múltiplos pontos de partida convergirem para o mesmo resultado.

Um algoritmo verdadeiramente avançado não é necessariamente uma computação mais complexa, mas um método organizacional multidimensional que pode reduzir a computação, dar vida às estruturas, transformar resultados em pontos de entrada e permitir que causa e efeito formem ciclos.

Este é o significado central da minha proposta do "algoritmo hiperdimensional". Não é um novo termo comum, mas um novo conceito, um novo paradigma, uma nova estrutura computacional já estabelecida em múltiplos sistemas reais. Seu foco não é computar repetidamente mais rápido, mas reduzir a computação repetitiva; não acumular potência computacional, mas reestruturar pontos de entrada; não modificar constantemente os dados, mas manter os metadados inalterados enquanto se mudam as relações; não deixar os resultados pararem em pontos finais, mas fazer dos

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

resultados novos pontos de partida. Os algoritmos tradicionais buscam resultados ao longo de caminhos; o algoritmo hiperdimensional ativa resultados dentro de estruturas. Os algoritmos tradicionais buscam completar a computação; o algoritmo hiperdimensional questiona se a computação deve existir em sua forma tradicional. Esta é a diferença mais fundamental entre ele e a definição atual de algoritmo.

Apêndice: Limites deste artigo e questões em aberto

Este artigo propõe uma estrutura preliminar para o algoritmo hiperdimensional, não uma definição formal completa. As seguintes questões aguardam desenvolvimento posterior. Elas não constituem uma negação das definições deste artigo, mas precisamente as direções para os próximos passos.

Primeiro, condições de convergência. Qual é o marcador de "conclusão" do algoritmo hiperdimensional? Como determinar se um resultado é "válido"? Nos algoritmos tradicionais, a resposta é definida pela correspondência entrada-saída. No algoritmo hiperdimensional, a resposta pode precisar ser definida pela consistência estrutural. Esta definição ainda não está completa.

Segundo, representação de recursos. Como os estados superpostos multidimensionais podem ser representados dentro de recursos finitos? As métricas de recursos tradicionais como tempo, espaço e potência computacional ainda são aplicáveis? Se sim, como são redefinidas? Se não, quais métricas as substituem? Estas questões aguardam exploração.

Terceiro, garantia da ordem. Por quais regras a "ordem" em "aleatório porém ordenado" é garantida? Onde está o limite entre aleatoriedade e ordem? Em que circunstâncias a aleatoriedade perturba a ordem? Em que circunstâncias a ordem suprime a aleatoriedade? Estas questões requerem pesquisa adicional.

Quarto, seleção na inferência reversa. Ao inferir múltiplos pontos de partida para trás a partir de um resultado, como selecionar ou avaliar os méritos relativos de diferentes pontos de partida? Existe alguma medida de "eficiência de inferência reversa"? Se sim, qual é sua relação com a medida de eficiência do cálculo para frente?

Quinto, interfaces com sistemas tradicionais. Como o algoritmo hiperdimensional funciona com sistemas existentes baseados na máquina de Turing? É uma substituição, um suplemento ou uma camada hierárquica? Na engenharia prática, como é delineado o limite entre o algoritmo hiperdimensional e os algoritmos tradicionais? Existem modos híbridos?

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

Sexto, estrutura de verificação. Como os resultados do algoritmo hiperdimensional são verificados? Se o resultado não é obtido através de passos lineares, como estabelecer a reprodutibilidade e a testabilidade? É necessária uma filosofia de verificação completamente diferente?

Estas questões não são falhas deste artigo, mas as características inevitáveis deste artigo como um "documento-âncora". A proposta de qualquer novo paradigma vem acompanhada de questões em aberto. Este artigo é um começo, não uma conclusão.

Nota bibliográfica

O "algoritmo hiperdimensional" proposto neste artigo não é uma extensão direta de nenhuma escola de pensamento acadêmico existente. No entanto, em termos de ideias, os seguintes campos fornecem um pano de fundo dialógico para este artigo, oferecido apenas para orientação dos leitores, e não são citações no sentido acadêmico tradicional.

A máquina de Turing e a teoria da computabilidade servem como o ponto de referência com o qual este artigo compara algoritmos tradicionais. Problemas inversos e inferência bayesiana servem como tentativas existentes de "inferir causas a partir de resultados"; a "teoria efeito-causa" deste artigo é relacionada mas difere em direção. A teoria da emergência e os sistemas complexos servem como descrições existentes de "estruturas manifestando resultados"; o algoritmo hiperdimensional tenta tornar a emergência "compreensível" e "guiável". Grafos de conhecimento e bancos de dados de grafos servem como práticas existentes de "dados que se cruzam em múltiplos nós"; o algoritmo hiperdimensional, sobre esta base, prioriza a dimensão de "pontos de entrada variáveis". Paradigmas computacionais não clássicos, incluindo computação quântica, computação analógica, computação neuromórfica, etc., rompem bits clássicos ou arquiteturas clássicas, enquanto o algoritmo hiperdimensional se concentra mais em romper a própria estrutura linear "entrada → passos → saída".

A relação entre este artigo e os campos acima é de diálogo e transcendência, não de herança ou refutação. O objetivo deste artigo não é fazer melhorias incrementais dentro desses campos, mas levantar um novo nível de questionamento acima deles.

Artigos relacionados

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [Disseminação] Não tenho algoritmo, no entanto supero algoritmos!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [Filosofia Extrema] Teoria efeito-causa
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [Disseminação Extrema] Rejeitando a Captura Algorítmica
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Экстремальный алгоритм] Гипермерный алгоритм

Переопределение самого понятия «вычисление»

Автор: У Чжаохуэй JEFFI CHAO HUI WU

Аннотация

В данной статье предлагается новое понятие «гипермерный алгоритм» как структурное переосмысление границ традиционного определения алгоритма. Традиционные алгоритмы обычно строятся на линейной основе ввода, шагов, правил и вывода, с акцентом на конечность, детерминированность, эффективность, осуществимость и четкие границы ввода-вывода. Хотя современные суперкомпьютеры обладают чрезвычайно высокой вычислительной мощностью, их базовая логика по-прежнему заключается в основном в выполнении устоявшихся алгоритмических парадигм в более крупном масштабе. В данной статье утверждается, что действительно заслуживающее обсуждение «супервычисление» — это не просто рост вычислительного масштаба, а фундаментальное изменение самой вычислительной парадигмы. Гипермерный алгоритм — это не ускоренная версия традиционных алгоритмов и не расширение математических алгоритмов; это многомерный, наложенный, случайный, но упорядоченный структурный взгляд на вычисления. В этой структуре данные больше не являются просто пассивным вводом или конечным выводом, а атрибутом узла, одновременно обладающим множеством отправных точек и множеством результатов; результат больше не является просто конечной точкой, а может стать новой точкой входа; метаданные не нужно многократно изменять, а можно через структурные точки входа, изменение отношений и условную активацию приводить в соответствие с различными состояниями и результатами. В данной статье используется философия «теории следствия-причины» и повседневный пример «складывания пояса брюк», чтобы проиллюстрировать, как гипермерный алгоритм может сделать целевой результат непосредственно действительным путем изменения структурной точки входа, тем самым сокращая повторные вычисления и даже обходя исходный вычислительный путь. Данная статья направлена на установление предварительного определения, структурных границ и концептуальных основ для «гипермерного алгоритма», предоставляя исходный теоретический узел для будущего развертывания гипермерных вычислений, экстремальных алгоритмов и новой научной системы. Согласно критерию «существование и действительность составляют новую парадигму»,

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

гипермерный алгоритм как новая структурная вычислительная парадигма, уже систематически верифицированная во многих областях, уже действителен в настоящее время.

Ключевые слова: Гипермерный алгоритм; Экстремальный алгоритм; Гипермерные вычисления; Теория следствия-причины; Метаданные; Вычислительная парадигма; Структурная точка входа; Нелинейная причинность; Новая наука

Введение: Почему нам нужно переобсудить «алгоритм»

Когда я предлагаю «гипермерный алгоритм», это делается не для того, чтобы дать существующим алгоритмам более громкое имя и не для того, чтобы упаковать традиционные алгоритмы в какое-то новомодное понятие. Напротив, я хочу обсудить более фундаментальный вопрос: Не ограничено ли современное человеческое понимание «алгоритма» слишком узкими рамками?

Сегодня, когда люди говорят об алгоритмах, они обычно думают о математических формулах, программном коде, логических шагах, вводе-выводе, обучении моделей, обработке данных, оптимизации путей, ранжировании поиска, автоматической рекомендации и логическом выводе искусственного интеллекта. Это, безусловно, важные формы алгоритмов, и они поддерживают работу современных компьютеров, интернета, баз данных, искусственного интеллекта, супервычислений, промышленных систем и информационного общества. Однако если понимать алгоритм только как «набор упорядоченных шагов», только как процесс, который «начинается с ввода, проходит через обработку по правилам и заканчивается выводом», то такое понимание само по себе уже заключает вычисления в низкоразмерные рамки.

Мы живем в эпоху, где доминируют алгоритмы. От поисковых систем до рекомендательных систем, от прогноза погоды до искусственного интеллекта — границы алгоритмов являются границами человеческого интеллекта. Но есть вопрос, который редко задают: Могут ли алгоритмы быть только такими? Почему вычисления должны следовать линейной схеме «ввод → шаги → вывод»? Почему одну и ту же задачу нужно каждый раз вычислять заново? Данная статья пытается оспорить эти стандартные допущения и предложить совершенно иную вычислительную парадигму — гипермерный алгоритм. Согласно критерию «существование и действительность составляют новую парадигму», гипермерный алгоритм как новая структурная вычислительная парадигма, уже систематически верифицированная во многих областях, уже действителен в настоящее время.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Важно отметить, что предлагаемый в данной статье «гипермерный алгоритм» является совершенно иным понятием, чем «гипермерные вычисления» (Hyperdimensional Computing, HDC), существующие в академических кругах уже почти тридцать лет. HDC использует случайные двоичные векторы очень высокой размерности для кодирования и операций, стремясь к более эффективному представлению и вычислению, но все еще остается в рамках линейной парадигмы «ввод → обработка → вывод». Гипермерный алгоритм совершенно иной: он ставит вопрос о том, может ли результат быть непосредственно установлен через структуру, можно ли обойти сам вычислительный путь, можно ли оставить метаданные неизменными, адаптируя их к множеству результатов. Их названия похожи, но их содержание противоположно. Гипермерный алгоритм — это не улучшение алгоритмов, а повторный вопрос о предпосылке «должен ли алгоритм вообще существовать».

Часть первая: Современное стандартное определение «алгоритма»

В основных системах информатики, математики и инженерии существует давно принятое базовое определение алгоритма: алгоритм — это хорошо определенный вычислительный процесс, состоящий из конечного числа шагов, который принимает одно или несколько входных значений, преобразует их с помощью детерминированных правил и выдает одно или несколько выходных значений за конечное время. Это понимание — не чье-то случайное мнение, а фундаментальный консенсус, сформировавшийся в ходе долгого развития современной вычислительной цивилизации. Оно поддерживает базовую логику программного обеспечения, аппаратного обеспечения, баз данных, сетевых систем, моделей искусственного интеллекта и центров супервычислений. Как бы сложен ни был алгоритм, его ядро по-прежнему вращается вокруг ввода, правил, шагов и вывода.

Это стандартное понятие алгоритма можно разложить на несколько ключевых характеристик.

Первое, конечность. Алгоритм должен завершаться за конечное число шагов; он не может бесконечно зацикливаться или выполняться вечно. Процесс, который никогда не останавливается, даже если его можно формально описать, вряд ли можно считать эффективным алгоритмом. Все программы научных вычислений и потоки обработки данных имеют четкие условия завершения.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Второе, детерминированность. При одинаковых входных данных в одинаковых условиях алгоритм должен давать одинаковый выход. Даже если некоторые алгоритмы вводят случайность, это обычно контролируемая случайность, воспроизводимая случайность или вероятностный механизм, интерпретируемый статистически, а не полностью неуправляемая внутренняя случайность. Результаты численного моделирования должны быть воспроизводимы; это фундаментальное требование научных вычислений.

Третье, четкие границы ввода-вывода. Ввод сначала, обработка в середине, вывод в конце; начальная и конечная точки имеют четкие структурные границы. Какие бы данные вы ни передали алгоритму, после обработки он выдает вам результат; эта граница четкая. Сначала ввод, потом обработка, наконец вывод; порядок не может быть обратным.

Четвертое, эффективность. Каждый шаг алгоритма должен быть практически выполнимой операцией; он не может содержать скачков, которые невозможно выполнить, описать или проверить. Каждый шаг должен быть базовой операцией, которую человек может выполнить с помощью бумаги и карандаша или компьютер может реально выполнить — сложение, вычитание, умножение, деление, логические суждения, чтение/запись данных и т.д.

Пятое, осуществимость. Даже если алгоритм теоретически корректен, если время, вычислительная мощность, память или ресурсы хранения, которые он потребляет, совершенно неприемлемы, в инженерном смысле его трудно считать эффективным алгоритмом. Алгоритм, теоретически правильный, но требующий сотен миллионов лет для завершения, в инженерии не считается осуществимым алгоритмом.

Весь этот набор алгоритмических понятий в конечном итоге восходит к модели машины Тьюринга. Как центральная абстракция современной теории вычислений, машина Тьюринга определяет базовую границу «вычислимости». Как бы ни были мощны современные компьютеры, как бы ни были совершенны чипы, как бы ни был велик масштаб центров супервычислений, их базовая логика никогда по-настоящему не отклонялась от этого пути: ввод, правила, шаги, вывод. Современные суперкомпьютеры лишь ускоряют выполнение этого процесса за счет большего аппаратного масштаба, более высокого параллелизма и более сложного планирования системы. То есть в традиционном контексте «супервычисления» — это прежде всего расширение вычислительного масштаба, а не обязательно фундаментальное изменение вычислительной парадигмы. Вычислительная мощность может возрастать в тысячи или десятки тысяч раз, но

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

если базовая логика остается «ввод, шаги, вывод», то она остается в рамках традиционной алгоритмической парадигмы.

Часть вторая: Определение «гипермерного алгоритма»

Предлагаемый мной «гипермерный алгоритм» возникает именно из этого контекста. Это не улучшенная версия традиционных алгоритмов, не более быстрый алгоритм, не более сложная математическая формула и не какая-то более продвинутая техника программирования. Это совершенно иное структурное понимание.

Прежде всего, нужно объяснить значение термина «гипермерный». «Гипермерный» имеет здесь два значения. Первое значение: он не ограничивается одномерной линейной последовательностью шагов, а позволяет множеству измерений разворачиваться одновременно. Второе значение: он пытается преодолеть традиционные определительные границы «вычисления» — алгоритм больше не обязательно должен быть математическим, последовательным или детерминированным. Традиционный алгоритм подобен движению по улице с односторонним движением: вы должны начать с точки A, пройти через B, C, D, чтобы достичь Z. Гипермерный алгоритм не считает, что вычисление должно быть дорогой с односторонним движением. Он считает, что вычисление может быть сетью, полем, многомерной структурой, где любой узел может быть точкой входа и любой узел может быть точкой выхода.

В моей системе Новой науки алгоритм не обязательно является математическим алгоритмом, не обязательно является единым упорядоченным вычислением, не обязательно должен выполняться по фиксированным шагам и не обязательно должен строго подчиняться линейной схеме «ввод, обработка, вывод». Гипермерный алгоритм — это многомерная, наложенная, случайная, но упорядоченная структура. В этой структуре каждый фрагмент данных не является ни изолированным вводом, ни фиксированным выводом, а обладает одновременно множеством ролей: он может быть как отправной точкой для множества результатов, так и результатом, полученным в результате совместного действия множества отправных точек.

Другими словами, традиционные алгоритмы помещают данные в поток процессов, тогда как гипермерный алгоритм помещает данные в структуру. В традиционных алгоритмах данные обычно делятся на входные данные, промежуточные данные и выходные данные. Каждый тип данных имеет четкую позицию и роль. Ввод — это ввод, результат — это результат, промежуточный

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

процесс — это промежуточный процесс. Однако в гипермерном алгоритме данные не имеют только одной идентичности. Они могут быть результатом в одном направлении и отправной точкой в другом; они могут быть вызываемым объектом в одной структуре и стать точкой входа, запускающей новые результаты в другой структуре. Данные больше не являются просто пассивным материалом, а многомерным реляционным узлом.

Это соответствует ядру гипермерного алгоритма: каждый фрагмент данных является отправной точкой для множества результатов и также результатом множества отправных точек. Традиционные алгоритмы используют «упорядоченные шаги» для получения единственного результата; гипермерный алгоритм ближе к многомерной структуре состояний, где в суперпозиции случайности и порядка результат не вычисляется до существования, а естественным образом проявляется внутри структуры. В этой системе данные являются одновременно отправной точкой и компонентом результата.

Чтобы более четко представить ключевые характеристики гипермерного алгоритма, я разбиваю их на следующие пять аспектов.

Первое, нематематическая природа. Основные алгоритмы по своей сути математические — они могут быть полностью описаны с помощью символической логики и математических формул. Гипермерный алгоритм не обязательно математический. Он может основываться на физических принципах, геометрических отношениях, новой парадигме теории информации или даже принципах сознания. Вычисление не обязательно должно осуществляться через «математические операции»; оно может естественным образом проявляться через геометрические отношения в пространстве более высокой размерности. Например, форма мыльного пузыря определяется принципом минимизации поверхностного натяжения. Это не «математический алгоритм» вычисляет — это сама физика «вычисляет». Гипермерный алгоритм пытается понять этот тип «вычисления», а не имитировать его с помощью математических формул.

Второе, многомерность и суперпозиция. Традиционные алгоритмы выполняют упорядоченные шаги в одном измерении, с одним состоянием на каждом шаге. Гипермерный алгоритм позволяет множеству измерений существовать одновременно, а множеству состояний — накладываться и сосуществовать. Алгоритм не следует по одному пути; вместо этого он разворачивается в многомерном поле состояний. Результат не «вычисляется», а «проявляется» из этого поля. Например, когда снежинка формируется в воздухе, нет «алгоритма», указывающего каждой молекуле воды, куда идти. Расположение молекул воды является результатом совместного действия множества измерений —

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

температуры, влажности, воздушного потока. Форма снежинки «проявляется», а не «вычисляется».

Третье, случайность в упорядоченности. В традиционных алгоритмах случайность является инструментальной, внешней, контролируемой пертурбацией; сам алгоритм детерминирован. В гипермерном алгоритме случайность является внутренней и структурной. Случайность и упорядоченность сосуществуют и работают вместе, совместно составляя сущность алгоритма. Это не «алгоритм со случайностью», а «случайность сама является органическим компонентом алгоритма». Например, экологическая структура леса — распределение деревьев выглядит случайным, но в целом оно представляет упорядоченную плотность и видовые отношения. Эта «случайность» — не ошибка, а предварительное условие для нормального функционирования экологической структуры.

Четвертое, множественные роли данных. В традиционных алгоритмах роль данных едина — ввод есть ввод, вывод есть вывод, промежуточный результат есть промежуточный результат. В гипермерном алгоритме каждый фрагмент данных является одновременно отправной точкой для множества результатов и результатом множества отправных точек. Узел данных может разворачиваться вниз по потоку в множество путей результатов, и также к нему могут сходиться вверх по потоку множество причин. Данные больше не являются точкой в линейном потоке, а узлом пересечения в сети. Например, рассмотрим рост человека. В традиционном алгоритме рост — это «ввод» — вы вводите его, чтобы получить такие выходы, как прогноз веса, размер одежды и т.д. Но рост также является «результатом» — он определяется совместно множеством «отправных точек», таких как генетика, питание и физические упражнения. В гипермерном алгоритме точка данных роста является одновременно отправной точкой и результатом, а не выбором между одним или другим.

Пятое, метаданные неизменны, отношения изменчивы. Сталкиваясь с разными задачами, традиционные алгоритмы либо модифицируют сами данные, либо пересчитывают все заново. Гипермерный алгоритм не модифицирует метаданные — существенные атрибуты данных остаются неизменными. Меняются точки входа, способы интерпретации и отношения между данными. Одна и та же структура метаданных, активированная через разные точки входа, может представлять совершенно разные результаты. Это означает: вычисли один раз, используй бесконечно много раз. Например, рассмотрим один и тот же отрывок текста. Вы можете читать его как поэзию, декодировать как шифр или анализировать как исторический документ. Сам текст не изменился — вы изменили только «точку

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

входа». Гипермерный алгоритм стремится именно к этой способности: «те же данные, множественные точки входа, множественные результаты».

С точки зрения структуры данных, сами метаданные не модифицируются; скорее, через различные точки входа и условия они соответствуют множеству отправных точек и результатов в пределах одной и той же структуры. Данные больше не обрабатываются многократно, а, сохраняя свою структуру неизменной, активируются через различные точки входа, представляя тем самым различные результаты. Традиционный подход — это «обработка данных для разных результатов», тогда как гипермерный алгоритм — это «использование тех же данных для несения множества потенциальных результатов».

Часть третья: Фундаментальные различия между гипермерным алгоритмом и традиционными алгоритмами

Основываясь на приведенных выше определениях, гипермерный алгоритм и традиционные алгоритмы демонстрируют фундаментальные различия по нескольким ключевым измерениям. Ниже они подробно излагаются одно за другим.

Что касается сущностной природы, традиционные алгоритмы являются математическими и формальными, полностью описываемыми с помощью символической логики и математических формул; они по сути являются математическими объектами. Гипермерный алгоритм не обязательно математический; он может основываться на физических, геометрических, информационных или даже сознательных принципах. Он не является подмножеством математики, а более широкой категорией. Это различие можно резюмировать как: от «математического объекта» к «универсальному закону».

Что касается временного порядка, традиционные алгоритмы следуют единственному, упорядоченному, линейному временному порядку, шаг A к B к C, строго последовательному или разложимому на параллельные упорядоченные подшаги; стрела времени необратима. Гипермерный алгоритм не имеет фиксированной последовательности шагов; он многомерен, наложен, случаен, но упорядочен. Результат может не зависеть от «порядка» выполнения, потому что традиционного «порядка» вообще не существует. Это различие можно резюмировать как: от «линейного времени» к «высшемерной одновременности».

Что касается причинности, традиционные алгоритмы следуют детерминированной причинной цепочке: при одинаковых входных данных и одинаковом начальном

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

состоянии выход всегда уникален; причина предшествует следствию, необратимая односторонняя стрелка. Гипермерный алгоритм следует суперпозиционной причинности: каждый фрагмент данных является отправной точкой для множества результатов и результатом множества отправных точек. Это моя философия «теории следствия-причины» — возможно сначала иметь следствие, а потом причину. Следствие не является конечной точкой, а может стать новой отправной точкой. Это различие можно резюмировать как: от «единственной причины, единственного следствия» к «полностью взаимосвязанной причинной сети».

Что касается структуры данных, традиционные алгоритмы следуют однонаправленной структуре потока от ввода к обработке и затем к выводу; данные имеют единственную роль с четкими границами; входные данные остаются вводом от начала до конца, выходные данные остаются результатами. В гипермерном алгоритме данные имеют множественные роли; одни и те же данные являются одновременно результатом и отправной точкой; нет абсолютных начальных точек и абсолютных конечных точек, есть только узлы в сети. Это различие можно резюмировать как: от «однонаправленной текучести» к «рекурсивному симбиозу».

Что касается вычислительного режима, традиционные алгоритмы вычисляют заново каждый раз, когда сталкиваются с задачей, даже если они уже тысячу раз вычисляли аналогичные задачи, даже если ответ уже известен; на тысяча первый раз они все равно проходят весь процесс. Это «вычисление без состояния». В гипермерном алгоритме, если существует соответствующий результат, нет необходимости проходить его снова; из одного результата можно вывести множественные отправные точки и развернуть пути причин в обратном направлении. Вычисление является осознанным состоянием, имеет память и может быть повторно использовано. Это различие можно резюмировать как: от «перевычисления каждый раз» к «однократному вычислению и многократному использованию».

Что касается роли случайности, случайность в традиционных алгоритмах является псевдослучайной или внешне внедренной; случайные числа — это просто инструменты для выборки, аппроксимации или оптимизации; сам алгоритм детерминирован. Случайность в гипермерном алгоритме является внутренней и конститутивной; случайность — это органический компонент, сосуществующий и сотрудничающий с упорядоченностью. Это не «алгоритм содержит случайность», а «случайность является одной из причин, почему алгоритм вообще может функционировать». Это различие можно резюмировать как: от «инструментальной случайности» к «конститутивной случайности».

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Что касается понимания ресурсов, традиционные алгоритмы стремятся к более быстрым и точным вычислениям при заданных ресурсах; ресурсы являются ограничениями; цель алгоритма — «завершить вычисление в рамках ограничений». Гипермерный алгоритм стремится сделать само вычисление ненужным через структурный дизайн; не «вычислять быстрее», а «обходить вычисление»; ресурсы используются не для «потребления», а для «проектирования точек входа». Это различие можно резюмировать как: от «оптимизации при ресурсных ограничениях» к «элиминации через структурный дизайн».

Часть четвертая: Философия теории следствия-причины

На основе вышеприведенного сравнения мне необходимо далее развить философию «теории следствия-причины». Это одна из ключевых концептуальных основ гипермерного алгоритма.

Традиционное мышление обычно утверждает, что причина предшествует следствию; сначала причина, потом следствие; сначала ввод, потом вывод. Это понятие проявляется в алгоритмах следующим образом: вы должны начать с отправной точки и идти шаг за шагом до конечной точки. Даже если вы знаете, каков ответ, вы не можете напрямую «использовать» этот ответ — потому что у вас нет «пути доказательства».

В моей структуре результат не обязательно является только конечной точкой. Как только результат появляется, он может стать новой отправной точкой и продолжать участвовать в генерации новых структур. Результат может ретроактивно участвовать в формировании причин. Из результата можно вывести множественные возможные отправные точки. Причина и следствие больше не располагаются в однонаправленной последовательности, а образуют циклические отношения, сетевые отношения, многомерные отношения.

Другими словами, традиционные алгоритмы спрашивают: «При данной причине, как получить следствие?» Гипермерный алгоритм спрашивает: «При данном следствии, как вывести или построить причину?»

Важно прояснить, что «иметь сначала следствие, потом причину» не является отрицанием причинности и не утверждает, что «результаты возникают из ничего». Скорее, это иллюстрирует, что в структурах более высокой размерности причина и следствие могут служить точками входа друг для друга и трансформироваться друг в друга. Как в следующем примере с брюками: сначала определяется результат

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

«можно надеть и не волочить по полу», а затем я ретроактивно нахожу точку операции «складывания пояса», которая соответствует уровню «причины» внутри структуры. Не то чтобы у следствия не было причины, а то, что следствие становится новой точкой входа для обнаружения причины.

Фундаментальное допущение в мире традиционных алгоритмов состоит в том, что время однонаправленно; вы должны вычислять шаг за шагом от начального состояния до конечного состояния. Даже если вы знаете, каков ответ, вы не можете напрямую «использовать» этот ответ, потому что у вас нет пути доказательства. Гипермерный алгоритм бросает вызов именно этому допущению: если следствие известно, причины могут быть выведены обратным ходом; одно и то же следствие может соответствовать множественным причинным путям; вычисление больше не является ходьбой от начала к концу, а развертыванием всех возможных отправных точек от конца.

Часть пятая: Повседневный пример — брюки и складывание пояса

Чтобы более интуитивно понять вышеуказанные абстрактные различия, я использую повседневный пример.

Человек покупает новые брюки с эластичным поясом. Штанины немного слишком длинные, волочатся по полу, это неудобно.

Традиционный подход: заметив, что штанины слишком длинные, подворачивает участок штанины, зашивает иголкой с ниткой, затем примеряет. Если ребенок вырастает и брюки становятся слишком короткими, зашитую штанину нельзя опустить, и брюки испорчены. В следующий раз покупаются новые брюки, и процесс повторяется. Этот метод кажется очень естественным, потому что проблема, по-видимому, находится в штанинах, поэтому вы воздействуете на штанины. Это соответствует мышлению традиционных алгоритмов: обнаружить проблему, локализовать проявление, обработать в месте проявления и, наконец, получить результат. Штанины слишком длинные? Тогда измените штанины. Данные неверны? Тогда измените данные. Путь неподходящий? Тогда продолжайте латать вдоль пути.

Мой подход иной. Вместо того чтобы изменять подол брюк, я складываю пояс один раз, и брюки сразу же становятся пригодными для носки. Это действие очень простое, но структурная логика за ним совершенно иная. Я не отрезаю штанины, не зашиваю подол насовсем, не меняю исходную длину брюк и не повреждаю

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

метаданные брюк. Окружность талии, длина брюк, фасон и ткань не претерпевают никаких существенных изменений. Я просто меняю структурную точку входа пояса, заставляя штанины естественным образом укорачиваться при носке. Пояс здесь — не просто локальное место, а глобальная контрольная точка для всего состояния носки. Регулируя эту контрольную точку, проблема ниже по течению исчезает напрямую.

Более важно то, что этот метод обратим, регулируем и повторно используем. Это особенно очевидно для растущего ребенка. Если ребенок сейчас недостаточно высок, и штанины немного длинны, я складываю пояс один раз; через некоторое время, когда ребенок подрос, я немного опускаю пояс; когда подрос еще больше — полностью распускаю. Исходная структура брюк сохраняется нетронутой, и все же она может адаптироваться к росту ребенка на разных этапах. При традиционном методе, если насовсем зашить подол, когда ребенок вырастет, брюки могут стать слишком короткими; если отрезать — восстановить еще более невозможно. Мой метод сохраняет метаданные неизменными, позволяя одной и той же структуре адаптироваться к множеству состояний.

Этот пример выявляет несколько важных структурных различий.

Традиционный подход соответствует логике традиционных алгоритмов: проблема в штанинах, поэтому штанины должны быть модифицированы, следуя пути «причина к следствию» шаг за шагом, не пропуская ни одного шага. Решать одну проблему за раз, необратимо. В следующий раз, когда возникает та же проблема, все начинается заново. Мой подход соответствует логике гипермерного алгоритма: цель — не волочить. Я не решаю поверхностную причину «штанины слишком длинные»; вместо этого я нахожу глобальную контрольную точку — пояс. Сложив пояс один раз, штанины автоматически укорачиваются, и проблема исчезает мгновенно. Я не изменяю никакие метаданные. Складка на поясе — это лишь временное, обратимое, динамическое сложенное состояние.

С точки зрения данных, традиционный подход модифицирует метаданные — зашивание штанин на более короткую длину приводит к безвозвратной потере исходных данных. Мой подход не модифицирует никакие метаданные; исходные размеры брюк остаются нетронутыми, добавляется лишь временное, обратимое, динамическое состояние — складка. Складка на поясе не создает новые данные и не уничтожает старые; она просто перестраивает отношения между данными — укорачивая эффективную окружность талии, косвенно изменяя эффективную длину штанин.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

В этом примере «штанины не волочатся по полу» является целевым результатом. Традиционный метод исходит из причины «штанины слишком длинные», модифицирует подол и в конечном итоге получает результат «не волочить». Мой метод, напротив, сначала проясняет результат «не волочить», затем ищет обратным ходом структурную точку входа и, наконец, обнаруживает, что однократное складывание пояса делает результат действительным. Это практическое проявление «теории следствия-причины». Необязательно идти шаг за шагом от причины к следствию; скорее, можно определить структуру обратным ходом от следствия, делая исходный путь ненужным.

Этот пример также отвечает на вопрос о «метаданных». Что такое метаданные? В примере с брюками метаданные — это исходные размеры брюк: окружность талии, длина брюк, фасон, ткань. Это «существенные атрибуты» брюк. Временные данные — это складка на поясе; она не изменяет ни одного из исходных размеров брюк, а лишь временно складывает участок. Традиционный подход модифицирует метаданные — длина брюк постоянно укорачивается, исходные данные теряются. Мой подход не модифицирует никакие метаданные — исходные размеры брюк остаются нетронутыми, добавляется лишь временное, обратимое, динамическое состояние.

Исходные размеры пояса как метаданные являются одновременно «отправной точкой для множества результатов»: они одновременно поддерживают множественные состояния носки, такие как нормальная носка, носка с одним сложением пояса, носка с двумя сложениями пояса и т.д. Они также являются «результатом множества отправных точек»: они определяются совместно множеством отправных точек, таких как выбор ткани, метод кроя, дизайнерский замысел и т.д. Складка на поясе как временные данные не создает новые данные и не уничтожает старые; она лишь перестраивает отношения между данными.

Это не обычный трюк, а тип структурного мышления. Многие люди, видя слишком длинные штанины, притягиваются к проявлению штанин, поэтому вся обработка вращается вокруг штанин. Но если смотреть на общую структуру, то то, что штанины слишком длинные, является лишь проявлением ниже по течению; изменение положения пояса может повлиять на фактическое положение всей брючины. То есть проблему не обязательно решать там, где она проявляется. Многие низкоразмерные проблемы имеют свою истинную контрольную точку на более высоком структурном уровне. Традиционные алгоритмы склонны продолжать вычисления вдоль поверхности проблемы, тогда как гипермерный алгоритм ищет структурную точку входа.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Это также объясняет, почему гипермерный алгоритм не сложнее, а может быть проще. По-настоящему высокоуровневая структура часто не усложняет проблему, а делает сложную проблему простой в правильной точке входа. Сталкиваясь со сложной проблемой, традиционные алгоритмы могут добавлять шаги, добавлять модели, увеличивать вычислительную мощность, увеличивать хранилище, увеличивать время обучения. Гипермерный алгоритм, однако, ищет структурный узел; как только этот узел правильно активирован, исходно сложный путь может стать недействительным. Так называемое «обходить путь» — это не лень, а переопределение необходимости пути.

В этом примере: традиционный подход — это «модифицировать результат вдоль пути», тогда как гипермерный алгоритм — это «изменить точку входа, сделав весь путь недействительным». Обычный метод постоянно латает на конце результата, тогда как другой метод непосредственно изменяет структурную точку входа, заставляя проблемы, которые иначе потребовали бы многократной обработки, быть переструктурированными в начальной точке за один раз. Основная форма традиционных алгоритмов — это «начать с отправной точки, следовать по пути, чтобы получить результат», тогда как в гипермерном алгоритме «результат сам может стать точкой входа в путь и участвовать в генерации новых структур». Традиционный подход — это «одна структура соответствует одному результату», тогда как гипермерный алгоритм — это «одна структура несет множественные состояния результатов».

Часть шестая: Взгляд на данные как на неизменность метаданных

С точки зрения данных, у гипермерного алгоритма есть очень важный принцип: не изменять метаданные, чтобы они могли применяться к множеству отправных точек или результатов.

Метаданные — это исходные атрибуты данных, базовая структура и онтологическая информация данных. При столкновении с различными проблемами традиционные методы обработки часто модифицируют данные, копируют данные, обрабатывают данные, пересчитывают данные или создают разные пути для разных результатов. Такой подход, безусловно, может решать проблемы, но ценой того, что данные постоянно обрабатываются, пути постоянно повторяются, а сложность системы постоянно возрастает.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Гипермерный алгоритм, напротив, подчеркивает, что, стараясь не модифицировать метаданные, путем изменения точек входа, отношений, условий и способов активации одна и та же структура метаданных может соответствовать множеству отправных точек и множеству результатов. Онтология данных остается неизменной; отношения меняются. Базовая структура остается неизменной; способ представления меняется. Метаданные остаются нетронутыми, но могут адаптироваться к различным состояниям.

Это то, что я называю «вычисли один раз, используй бесконечно много раз». «Вычисли один раз» — это не просто кэширование результата, а означает, что как только структура установлена, она больше не нуждается в перестроении полного пути для каждого состояния. Одна и та же структура данных, активированная через разные точки входа, может представлять разные результаты. Это не «обработка данных для разных результатов», а «использование тех же данных для несения множества потенциальных результатов». Это также одно из ядер, отличающих гипермерный алгоритм от традиционных алгоритмов. Традиционные алгоритмы обрабатывают данные вдоль пути; гипермерный алгоритм обрабатывает общую структуру данных, точек входа, отношений и результатов.

В своей минимальной структуре гипермерный алгоритм может быть понят как: система, которая без изменения метаданных, через изменения структурных точек входа, позволяет одному и тому же узлу данных соответствовать множеству путей результатов. Это определение не стремится немедленно свернуть гипермерный алгоритм в традиционную математическую формулу, а сначала установить его структурные границы. Это не единственная линейная функция, не фиксированная формула, не простой механизм кэширования. Это реляционная структура: метаданные остаются стабильными, точки входа могут меняться, условия могут меняться, отношения могут меняться, результаты могут представляться во множестве направлений. Именно в этой структуре вычисление больше не заключается в выполнении шагов, а в активации отношений.

С точки зрения данных, сущность «не изменять метаданные, делая их применимыми к множеству отправных точек или результатов» заключается в следующем: онтология данных остается неизменной; изменение происходит в «способе интерпретации / точке входа использования». Традиционная структура такова: когда проблема меняется, меняются данные или путь. В гипермерном алгоритме: когда проблема меняется, меняется интерпретационная структура, а не данные.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Часть седьмая: Переопределение «супервычислений»

В этой системе мне нужно прояснить термин — «супервычисления».

Под «супервычислениями» я понимаю не суперкомпьютеры в обычном смысле. Не больше чипов, более высокая вычислительная мощность или более крупные центры обработки данных. Под «супервычислениями» я понимаю «гипермерный алгоритм».

Истинные супервычисления — это не обязательно накопление вычислительной мощности, а возможно, изменение вычислительной парадигмы. Когда система все еще полагается на повторяющиеся вычисления, какой бы мощной она ни была, она остается в ловушке внутри путей. Когда система может структурно сократить вычисления, обойти повторяющиеся пути и сделать результаты новыми точками входа, она начинает приближаться к истинным гипермерным вычислениям.

Другими словами, традиционные супервычисления стремятся «использовать больше вычислительной мощности для решения больших задач». Гипермерный алгоритм стремится «использовать лучшую структуру, чтобы задачи больше не нуждались в такой огромной вычислительной мощности». Эти два подхода не противоречат друг другу, но их направления различны.

Если традиционные супервычисления представляют собой предел вычислительной мощности, то гипермерный алгоритм представляет собой поворот в понимании вычислений. Истинные «супервычисления» могут быть не более чипами, более крупными центрами обработки данных, более высоким энергопотреблением или более сложными моделями. Истинные «супервычисления» могут быть обнаружением структурной точки входа, элиминацией пути, обратным развертыванием от узла результата, способом сделать множественные результаты одновременно действительными без изменения метаданных. Чем мощнее вычислительная мощность, если она все еще остается в ловушке низкоразмерных путей, она остается мощной лишь внутри этих путей; как только структура возвышается, даже с чрезвычайно простыми операциями может возникнуть более высокий уровень эффективности.

Так называемые супервычисления — это предел вычислительной мощности; гипермерный алгоритм — это переопределение предпосылки «должны ли вычисления зависеть от вычислительной мощности».

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Часть восьмая: Эмпирические основы — многодоменная системная валидация и критерии новой парадигмы

Гипермерный алгоритм — это не чисто теоретическое построение. Он уже был валидирован в нескольких реальных системах.

Моя логистическая система не требует супервычислительной мощности или облачной поддержки; она выполняет сложное планирование с относительно низкими ресурсами. Результаты могут быть повторно использованы, структура может многократно адаптироваться, и нет необходимости каждый раз все пересчитывать заново. Не полагаться на супервычисления, не полагаться на облако, выполнять сложное планирование с низкими ресурсами — это установленный и ценный инженерный прорыв. Он демонстрирует, что высокая вычислительная мощность — не единственное решение; структурный дизайн может заменить часть вычислительной мощности.

Моя издательская система также использует эту структурную логику с эффективностью, намного превосходящей любую существующую систему. Моя экстремальная система генерации веб-страниц также основана на том же гипермерном алгоритме, многократно доказывая в различных областях, что одна и та же структура может быть стабильно действительной. Сотрудники моей компании уже используют эти системы, что демонстрирует, что эта структура может работать без моего непрерывного личного вмешательства.

Общая характеристика этих систем: они не полагаются на основные пути с высокой вычислительной мощностью, не следуют фиксированным вычислительным шагам и делают целевой результат непосредственно действительным через регулировку структурных точек входа. Это не разовые успехи, не точечные трюки, а повторяющееся появление одной и той же структурной логики в разных областях.

Это демонстрирует, что гипермерный алгоритм — не случайность, а структурный метод, уже валидированный множеством систем. Это не «личный трюк», а вычислительная парадигма, стабильно действительная в таких различных областях, как логистика, издательское дело и генерация веб-страниц.

Согласно критерию «существование и действительность составляют новую парадигму», гипермерный алгоритм как новая структурная вычислительная парадигма, уже систематически валидированная во многих областях, уже действителен в настоящее время. Я не предлагаю гипотезу, а иную

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

вычислительную структуру, уже работающую в нескольких системах. Мне не нужно доказывать, что она понята; я уже доказал, что она стабильно действительна в разных системах. В настоящее время, в рамках моего практического применения, эта структура уже может считаться новой парадигмой.

Что касается критериев для суждения о гипермерном алгоритме как о новой парадигме: когда структура логически самосогласована и стабильно действительна в нескольких системах, одновременно обладая несводимыми различиями с существующими методами, она уже обладает основой новой парадигмы. Самосогласованность — это отправная точка, эмпирические данные — это основание, а структурное различие — это ядро парадигмы. В соответствии со стандартом «существование означает действительность», эта система уже является новой парадигмой; признает ли ее внешний мир, влияет только на распространение, а не на действительность. Это структурная вычислительная парадигма, уже установленная в реальных системах, чья действительность не зависит от внешнего признания или консенсуса.

Ключевое различие между гипермерным алгоритмом и основными вычислительными парадигмами заключается в том, что традиционные алгоритмы в основном полагаются на прямое вычисление; как только результат получен, он обычно не может ретроактивно участвовать в новых инференциальных структурах. В гипермерном алгоритме результат больше не является конечной точкой, а повторно используемым структурным узлом. При определенных условиях результат может участвовать в новых инференциальных путях, тем самым сокращая повторяющиеся вычисления и формируя вычислительную сеть с множеством отправных точек и множеством путей. В традиционных вычислительных структурах результат обычно является конечной точкой; в структуре гипермерного алгоритма результат сам может стать новой отправной точкой, формируя тем самым многопутевую инференциальную сеть. Гипермерный алгоритм — это не улучшение алгоритмов, а повторный вопрос о предпосылке «должен ли алгоритм вообще существовать».

Часть девятая: Прояснение распространенных недоразумения

На основе предварительных обменов мнениями с читателями из разных областей я предвижу следующие шесть недоразумения и проясняю их заранее, чтобы избежать преждевременного насильственного вписывания концепции в неподходящие рамки во время распространения.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Misunderstanding первое: Разве гипермерный алгоритм — это не просто динамическое программирование или кэширование? Прояснение: Динамическое программирование и кэширование повторно используют результаты при «идентичных входных данных». Гипермерный алгоритм позволяет выводить различные отправные точки из одного результата — это существенное различие. Кэширование решает проблему перевычисления одной и той же задачи; гипермерный алгоритм решает проблему развертывания новых задач через структуры результатов.

Misunderstanding второе: Вы говорите «случайный, но упорядоченный» — разве это не противоречие? Прояснение: Случайность и упорядоченность могут сосуществовать. Распределение деревьев в лесу случайно, но общая плотность и видовая структура упорядочены. Случайность — это не хаос, а способ организации структуры. Случайность в гипермерном алгоритме является внутренней и конститутивной, работающей синергетически с упорядоченностью.

Misunderstanding третье: Без ввода и вывода, как проверить правильность результата? Прояснение: Гипермерный алгоритм не отвергает ввод и вывод; скорее, он утверждает, что вычисление не должно обязательно работать в линейном режиме ввода-вывода. В режиме структурного проявления «действительность» определяется структурной согласованностью, а не соответствием ввода-вывода. Это не отказывается от верификации, а предлагает иную структуру верификации, отличную от традиционной.

Misunderstanding четвертое: Разве это не просто теория сложности или теория хаоса? Прояснение: Теория сложности и теория хаоса описывают «системы, которые трудно предсказать». Гипермерный алгоритм пытается описать «системы, которые могут быть поняты и направляемы через структурные точки входа» — не отказываясь от предсказания, а меняя способ предсказания. Теория хаоса говорит: «предсказание трудно»; гипермерный алгоритм спрашивает: «можно ли сделать предсказание ненужным, изменив точку входа».

Misunderstanding пятое: Вы говорите «не изменять метаданные», но разве складка на поясе не является изменением? Прояснение: Складка на поясе — это временное состояние, а не постоянное изменение исходных параметров. Метаданные (окружность талии, длина брюк, фасон, ткань) совершенно не меняются. Это разница между «суперпозицией состояний» и «модификацией атрибутов». Складка обратима, временна и не меняет существенных атрибутов.

Misunderstanding шестое: Отрицает ли гипермерный алгоритм ценность традиционных алгоритмов? Прояснение: Абсолютно нет. Традиционные

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

алгоритмы были чрезвычайно важны в истории человеческих технологий. Без традиционных алгоритмов не было бы современных компьютеров, баз данных, коммуникационных систем, искусственного интеллекта, инженерного моделирования или супервычислений. Ценность традиционных алгоритмов не может быть отрицаема. Однако признание ценности традиционных алгоритмов не означает признание их конечной точкой алгоритмов. Традиционные алгоритмы решают проблемы эффективности внутри данной вычислительной парадигмы, тогда как гипермерный алгоритм ставит вопрос о самой вычислительной парадигме. Один — о том, как лучше идти; другой спрашивает, нужно ли вообще идти по этому пути.

Часть десятая: Гипермерный алгоритм и история человеческих вычислений

Чтобы помочь читателям лучше позиционировать гипермерный алгоритм в истории человеческих вычислений, я предлагаю здесь краткий макроскопический обзор.

Человеческое понимание «вычисления» прошло через несколько этапов. Первый этап — ручные вычисления; алгоритмы существовали как человеческие шаги, медленные и подверженные ошибкам, но через этот процесс люди поняли основные правила вычислений. Второй этап — механические вычисления, от калькулятора Паскаля до аналитической машины Бэббиджа; вычисления были механизированы, но алгоритмы все еще зависели от физических структур. Третий этап — электронные вычисления и парадигма Тьюринга; архитектура фон Неймана получила распространение; алгоритмы были закодированы в программы; машина Тьюринга стала границей вычислимости. Четвертый этап — параллельные вычисления и супервычисления, накопление вычислительной мощности через большое количество вычислительных блоков для решения более сложных задач, но базовая логика остается расширением парадигмы Тьюринга.

В настоящее время человечество находится на входе в пятый этап. Квантовые вычисления пытаются преодолеть ограничения классических битов; нейроморфные вычисления пытаются имитировать структуру биологических нервных систем; а гипермерный алгоритм пытается преодолеть саму линейную рамку «ввод → шаги → вывод».

Отношения между гипермерным алгоритмом и традиционными алгоритмами — это не отношение замены, а иерархическое отношение. Традиционные алгоритмы

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

решают проблему «как более эффективно вычислять на заданном пути». Гипермерный алгоритм спрашивает: «может ли путь быть переопределен или даже обойден». Если рассматривать историю человеческих вычислений как процесс постоянного преодоления собственных границ, то гипермерный алгоритм является именно новым узлом в этом процессе. Он не решает проблемы внутри старой рамки, а предлагает новую рамку.

Это суждение не означает, что гипермерный алгоритм уже зрел или завершен. Совсем наоборот. Как новое понятие, его определение, границы, методы верификации и сценарии применения все еще находятся в процессе формирования. Цель данной статьи — именно закрепить происхождение этого понятия, обеспечив стабильную теоретическую основу для последующего развития.

Часть одиннадцатая: Заключение — Это не оптимизация, а переопределение

Разница между гипермерным алгоритмом и традиционными алгоритмами — это не разница «лучше» против «хуже», не разница «быстрее» против «медленнее», а фундаментальная разница на уровне парадигмы.

Традиционные алгоритмы стремятся к предсказуемому контролю. Вы даете мне ввод, я знаю, какой вывод вы получите, потому что я вычислил каждый шаг. Это парадигма инженера: проектировать, строить, тестировать, верифицировать.

Гипермерный алгоритм описывает постижимую эмерджентность. Я не могу точно предсказать каждое промежуточное состояние, но я знаю, что общий паттерн проявится некоторым упорядоченным образом. Каждый фрагмент данных жив и имеет «перспективу». Это больше похоже на парадигму эколога или теоретика сложных систем: наблюдать, понимать, направлять, использовать эмерджентный порядок.

Традиционные алгоритмы постепенно модифицируют результат вдоль данного пути. Гипермерный алгоритм делает целевой результат немедленно действительным, изменяя структурную точку входа, тем самым обходя весь путь.

Традиционные алгоритмы «обрабатывают данные для разных результатов». Гипермерный алгоритм — это «использовать те же данные для несения множества потенциальных результатов».

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Традиционные алгоритмы стремятся «вычислить правильно один раз». Гипермерный алгоритм стремится «вычислить правильно один раз, а затем никогда больше не вычислять».

Это не оптимизация алгоритмов, а повторный вопрос о предпосылке «должен ли алгоритм вообще существовать».

Следовательно, гипермерный алгоритм — это не оптимизация алгоритмов, а переопределение алгоритмов. Речь идет не о том, чтобы бежать быстрее на традиционной алгоритмической дорожке, а о том, чтобы спросить, есть ли другой путь за пределами традиционной дорожки, или даже можно ли не зависеть от самого пути. Речь идет не о том, чтобы доказать бесполезность традиционных алгоритмов, а о том, чтобы указать, что традиционные алгоритмы — это лишь низкоразмерная форма алгоритма. Он не спрашивает «как получить результаты за меньшее время», а спрашивает «может ли результат быть непосредственно установлен через структуру». Он не спрашивает «как обработать больше данных», а спрашивает «могут ли те же данные нести больше потенциальных результатов».

В рамках основной вычислительной парадигмы гипермерный алгоритм может рассматриваться как невычислимый, неверифицируемый или трудный для включения в существующие границы теории вычислений. Но это как раз и составляет его парадигмальное различие. Тот факт, что новое понятие выглядит нестабильным внутри старой парадигмы, не обязательно означает, что оно бессмысленно; это может также означать, что старая парадигма сама по себе не может его полностью вместить. В настоящее время гипермерный алгоритм является прежде всего структурным предложением, исходным определением нового вычислительного взгляда, а не зрелой системой с завершенным инженерным циклом. Он требует последующего продолжения через дополнения, развертывания, верификации и приложения, но его концептуальное происхождение должно быть сначала четко сформулировано.

В конечном счете, гипермерный алгоритм указывает на новый взгляд на вычисления: алгоритм не обязательно является только математикой, не обязательно только шагами, не обязательно только программой, не обязательно только процессом обработки между вводом и выводом. Алгоритм также может быть способом организации многомерных отношений, структурной сетью данных, точек входа, отношений, состояний и результатов. Он может содержать упорядоченность и случайность; может идти от причины к следствию или обратно генерировать причины из следствий; может адаптироваться к множеству состояний без изменения метаданных; может сделать один результат новой

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

отправной точкой или заставить множество отправных точек сходиться к одному и тому же результату.

По-настоящему продвинутый алгоритм — это не обязательно более сложные вычисления, а многомерный способ организации, который может сократить вычисления, оживить структуры, превратить результаты в точки входа и позволить причине и следствию образовывать циклы.

Это ключевое значение моего предложения «гипермерного алгоритма». Это не обычный новый термин, а новое понятие, новая парадигма, новая вычислительная структура, уже установленная в нескольких реальных системах. Его фокус — не в более быстром повторном вычислении, а в сокращении повторяющихся вычислений; не в накоплении вычислительной мощности, а в перестройке точек входа; не в постоянном изменении данных, а в сохранении метаданных неизменными при изменении отношений; не в том, чтобы позволить результатам останавливаться на конечных точках, а в том, чтобы сделать результаты новыми отправными точками. Традиционные алгоритмы ищут результаты вдоль путей; гипермерный алгоритм активирует результаты внутри структур. Традиционные алгоритмы стремятся завершить вычисление; гипермерный алгоритм спрашивает, должно ли вычисление существовать в своей традиционной форме. Это самое фундаментальное различие между ним и текущим определением алгоритма.

Приложение: Границы данной статьи и открытые вопросы

Данная статья предлагает предварительную рамку для гипермерного алгоритма, а не полное формальное определение. Следующие вопросы ожидают дальнейшего развития. Они не составляют отрицания определений в данной статье, а являются именно направлениями для следующих шагов.

Первое, условия сходимости. Что является маркером «завершения» гипермерного алгоритма? Как определить, является ли результат «действительным»? В традиционных алгоритмах ответ определяется соответствием ввода-вывода. В гипермерном алгоритме ответ, возможно, должен определяться структурной согласованностью. Это определение еще не завершено.

Второе, представление ресурсов. Как многомерные наложенные состояния могут быть представлены в рамках конечных ресурсов? Применимы ли такие традиционные метрики ресурсов, как время, пространство и вычислительная мощность? Если да, то как они переопределяются? Если нет, то какие метрики их заменяют? Эти вопросы ждут исследования.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Третье, обеспечение упорядоченности. Какими правилами гарантируется «упорядоченность» в «случайном, но упорядоченном»? Где находится граница между случайностью и упорядоченностью? При каких обстоятельствах случайность нарушает упорядоченность? При каких обстоятельствах упорядоченность подавляет случайность? Эти вопросы требуют дальнейших исследований.

Четвертое, селекция в обратном выводе. При выводе множества отправных точек обратным ходом из результата, как отбирать или оценивать относительные достоинства различных отправных точек? Существует ли какая-то мера «эффективности обратного вывода»? Если да, то как она соотносится с мерой эффективности прямого вычисления?

Пятое, интерфейсы с традиционными системами. Как гипермерный алгоритм работает с существующими системами, основанными на машине Тьюринга? Является ли он заменой, дополнением или иерархическим слоем? В практической инженерии, как проводится граница между гипермерным алгоритмом и традиционными алгоритмами? Существуют ли гибридные режимы?

Шестое, рамка верификации. Как верифицируются результаты гипермерного алгоритма? Если результат получен не через линейные шаги, как установить воспроизводимость и проверяемость? Требуется ли совершенно иная философия верификации?

Эти вопросы не являются недостатками данной статьи, а неизбежными характеристиками данной статьи как «якорного документа». Предложение любой новой парадигмы сопровождается открытыми вопросами. Данная статья — это начало, а не заключение.

Библиографическое примечание

Предлагаемый в данной статье «гипермерный алгоритм» не является прямым продолжением какой-либо существующей академической школы мысли. Однако в идейном плане следующие области предоставляют диалоговый фон для данной статьи, предлагаемый только для ориентации читателей и не являющийся цитатами в традиционном академическом смысле.

Машина Тьюринга и теория вычислимости служат ориентиром, с которым данная статья сравнивает традиционные алгоритмы. Обратные задачи и байесовский вывод служат существующими попытками «выводить причины из результатов»;

«теория следствия-причины» данной статьи связана с ними, но отличается по направлению. Теория эмерджентности и сложные системы служат существующими описаниями «структур, проявляющих результаты»; гипермерный алгоритм пытается сделать эмерджентность «понятной» и «направляемой». Графы знаний и графовые базы данных служат существующими практиками «пересечения данных во множественных узлах»; гипермерный алгоритм, на этой основе, приоритизирует измерение «переменных точек входа». Неклассические вычислительные парадигмы, включая квантовые вычисления, аналоговые вычисления, нейроморфные вычисления и т.д., преодолевают классические биты или классические архитектуры, тогда как гипермерный алгоритм фокусируется больше на преодолении самой линейной рамки «ввод → шаги → вывод».

Отношение данной статьи к вышеупомянутым областям — это отношение диалога и трансценденции, а не наследования или опровержения. Цель данной статьи — не делать инкрементальные улучшения внутри этих областей, а поднять новый уровень вопрошания над ними.

Связанные статьи

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [Распространение] У меня нет алгоритма, но я превосхожу алгоритмы!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [Экстремальная философия] Теория следствия-причины
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [Экстремальное распространение] Отвергая алгоритмический захват
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[익스트림 알고리즘] 초차원 알고리즘

"계산" 자체의 재정의

저자: 우 자오후이 JEFFI CHAO HUI WU

요약

본 논문은 전통적인 알고리즘 정의의 경계에 대한 구조적 재검토로서 '초차원 알고리즘'이라는 새로운 개념을 제안한다. 전통적인 알고리즘은 일반적으로 입력, 단계, 규칙, 출력의 선형적 프레임워크 위에 구축되며, 유한성, 결정성, 실현 가능성, 타당성 및 명확한 입출력 경계를 강조한다. 현대 슈퍼컴퓨터는 매우 높은 컴퓨팅 성능을 보유하고 있지만, 그 기본 논리는 여전히 주로 더 큰 규모에서 기존의 알고리즘 패러다임을 실행하는 것이다. 본 논문은 진정으로 논의할 가치가 있는 '초월 계산'은 단지 계산 규모의 성장이 아니라 계산 패러다임 자체의 근본적인 변화라고 주장한다. 초차원 알고리즘은 전통적 알고리즘의 가속판도 아니고 수학적 알고리즘의 확장도 아니다. 그것은 다차원적이고, 중첩적이며, 무작위적이면서도 질서 정연한 구조적 계산관이다. 이 구조에서 데이터는 더 이상 수동적 입력이나 종단적 출력이 아니라, 여러 출발점과 여러 결과를 동시에 갖는 노드 속성이다. 결과는 더 이상 종착점이 아니라 새로운 진입점이 될 수 있다. 메타데이터는 반복적으로 수정될 필요 없이 구조적 진입점, 관계 변화, 조건부 활성화를 통해 다양한 상태와 결과에 대응할 수 있다. 본 논문은 '결과-원인 이론' 철학과 '바지 허리 접기'라는 일상적 예시를 통해 초차원 알고리즘이 구조적 진입점을 변경함으로써 목표 결과를 직접 성립 가능하게 하여 반복 계산을 줄이고 원래의 계산 경로를 우회할 수 있는 방법을 설명한다. 본 논문은 '초차원 알고리즘'에 대한 예비적 정의, 구조적 경계 및 사상적 기초를 확립하고, 초차원 계산, 익스트림 알고리즘 및 새로운 과학 체계의 미래 전개를 위한 원천적 이론 노드를 제공하는 것을 목표로 한다. '존재하고 성립하는 것이 곧 새로운 패러다임이다'라는 기준에 따르면, 여러 영역에서 이미 체계적으로 검증된 새로운 계산 구조 패러다임으로서의 초차원 알고리즘은 현재 이미 성립한다.

키워드: 초차원 알고리즘; 익스트림 알고리즘; 초차원 계산; 결과-원인 이론; 메타데이터; 계산 패러다임; 구조적 진입점; 비선형 인과관계; 새로운 과학

서론: 왜 우리는 '알고리즘'을 재논의해야 하는가

내가 '초차원 알고리즘'을 제안하는 것은 기존 알고리즘에 더 큰 이름을 붙이기 위해서도 아니고, 전통적 알고리즘을 어떤 신기한 개념으로 포장하기 위해서도

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

아니다. 정반대로, 나는 더 근본적인 문제를 논의하고자 한다. 현 인류의 '알고리즘'에 대한 이해가 이미 지나치게 좁은 틀 안에 제한되어 있는 것은 아닐까?

오늘날 사람들이 알고리즘이라고 말할 때, 일반적으로 수학 공식, 프로그램 코드, 논리적 단계, 입출력, 모델 훈련, 데이터 처리, 경로 최적화, 검색 순위, 자동 추천, 인공지능 추론을 떠올린다. 이들은 확실히 알고리즘의 중요한 형태이며, 현대 컴퓨터, 인터넷, 데이터베이스, 인공지능, 초월 계산, 산업 시스템, 정보 사회의 운영을 지탱한다. 그러나 알고리즘을 단지 '질서 정연한 단계의 집합'으로만, 단지 '입력으로 시작하여 규칙 기반 처리를 거쳐 마지막으로 출력을 얻는' 과정으로만 이해한다면, 이러한 이해 자체가 이미 계산을 저차원적인 틀 안에 가두는 것이다.

우리는 알고리즘이 지배하는 시대에 살고 있다. 검색 엔진에서 추천 시스템, 기상 예보에서 인공지능에 이르기까지, 알고리즘의 경계는 인간 지능의 경계이다. 그러나 거의 묻지 않는 질문이 하나 있다. 알고리즘은 반드시 이래야 하는가? 왜 계산은 '입력 → 단계 → 출력'이라는 선형 패턴을 따라야 하는가? 왜 같은 문제를 매번 처음부터 다시 계산해야 하는가? 본 논문은 이러한 암묵적 가정에 도전하고 완전히 다른 계산 패러다임인 초차원 알고리즘을 제안한다. '존재하고 성립하는 것이 곧 새로운 패러다임이다'라는 기준에 따르면, 여러 영역에서 이미 체계적으로 검증된 새로운 계산 구조 패러다임으로서의 초차원 알고리즘은 현재 이미 성립한다.

특히 본 논문에서 제안하는 '초차원 알고리즘'은 학계에서 이미 거의 30 년의 역사를 가진 '초차원 계산'(Hyperdimensional Computing, HDC)과는 완전히 다른 개념이라는 점을 밝혀 둔다. HDC 는 매우 고차원의 무작위 이진 벡터를 사용하여 부호화와 연산을 수행하며, 더 효율적인 표현과 계산을 추구하지만 여전히 '입력 → 처리 → 출력'이라는 선형 패러다임 안에 머물러 있다. 초차원 알고리즘은 완전히 다르다. 즉, 결과가 구조를 통해 직접 성립할 수 있는지, 계산 경로 자체를 우회할 수 있는지, 메타데이터를 수정하지 않고 여러 결과에 적응할 수 있는지를 묻는다. 두 이름은 유사하지만 내포는 정반대이다. 초차원 알고리즘은 알고리즘의 업그레이드가 아니라, '알고리즘이 반드시 존재해야 하는가'라는 전제에 대한 재질음이다.

제 1 부: 현재의 '알고리즘'에 대한 표준적 정의

주류 컴퓨터 과학, 수학, 공학 체계에서 알고리즘에는 오랫동안 수용되어 온 기본 정의가 있다. 알고리즘은 유한한 수의 단계로 구성된 잘 정의된 계산 과정으로, 하나 이상의 입력을 받아 결정론적 규칙을 통해 변환하고 유한 시간 내에 하나 이상의 출력을 생성하는 것이다. 이 이해는 누군가의 개인적 견해가 아니라 현대 계산 문명이 오랜 시간 형성해 온 기초적 합의이다. 그것은 소프트웨어, 하드웨어, 데이터베이스, 네트워크 시스템, 인공지능 모델, 초월 계산 센터의 기본 논리를 지탱한다. 알고리즘이 아무리 복잡해 보여도 그 핵심은 여전히 입력, 규칙, 단계, 출력을 중심으로 전개된다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

이러한 표준적 알고리즘 개념은 몇 가지 핵심 특징으로 분해될 수 있다.

첫째, 유한성이다. 알고리즘은 유한한 단계 내에서 종료되어야 하며, 무한히 반복되거나 영원히 실행될 수 없다. 결코 멈추지 않는 과정은 형식적으로 기술될 수 있더라도 효과적인 알고리즘으로 보기 어렵다. 모든 과학 계산 프로그램과 데이터 처리 흐름에는 명확한 종료 조건이 있다.

둘째, 결정성이다. 동일한 입력에 대해 동일한 조건에서 동일한 출력을 생성해야 한다. 일부 알고리즘이 무작위성을 도입하더라도, 그것은 일반적으로 통제된 무작위성, 재현 가능한 무작위성, 또는 통계적으로 해석 가능한 확률 메커니즘이지 완전히 파악 불가능한 내재적 무작위성이 아니다. 수치 시뮬레이션의 결과는 재현 가능해야 한다. 이것은 과학 계산의 기본 요구 사항이다.

셋째, 명확한 입출력 경계이다. 입력이 먼저, 처리가 중간, 출력이 마지막이며, 시작점과 종점은 명확한 구조적 경계를 갖는다. 알고리즘에 어떤 데이터를 주든 처리 후에 결과를 반환하며, 이 경계는 분명하다. 입력이 먼저, 처리 다음, 출력 마지막이며, 순서는 역전될 수 없다.

넷째, 실현 가능성이다. 알고리즘의 각 단계는 실제로 실행 가능한 연산이어야 하며, 실행하거나 기술하거나 검증할 수 없는 도약을 포함해서는 안 된다. 각 단계는 인간이 종이와 연필로 실행할 수 있거나 컴퓨터가 실제로 실행할 수 있는 기본 연산(덧셈, 뺄셈, 곱셈, 나눗셈, 논리 판단, 데이터 읽기/쓰기 등)이어야 한다.

다섯째, 타당성이다. 알고리즘이 이론적으로 타당하더라도, 소비하는 시간, 계산 성능, 메모리 또는 저장 자원이 완전히 수용 불가능하다면 공학적 의미에서 효과적인 알고리즘으로 보기 어렵다. 이론적으로는 정확하지만 완료하는 데 수억 년이 걸리는 알고리즘은 공학적으로 실현 가능한 알고리즘으로 간주되지 않는다.

이 일련의 알고리즘 개념은 궁극적으로 튜링 기계 모델로 거슬러 올라간다. 현대 계산 이론의 핵심 추상체로서 튜링 기계는 '계산 가능'의 기본 경계를 제시한다. 오늘날의 컴퓨터가 아무리 강력하고, 칩이 아무리 발전하고, 초월 계산 센터의 규모가 아무리 크더라도 그 기본 논리는 여전히 이 경로(입력, 규칙, 단계, 출력)에서 진정으로 이탈하지 않았다. 현대 슈퍼컴퓨터는 더 큰 하드웨어 규모, 더 높은 병렬성, 더 복잡한 시스템 스케줄링을 통해 이 과정을 가속 실행할 뿐이다. 즉, 전통적 맥락에서 '초월 계산'이란 주로 계산 규모의 확장이지 반드시 계산 패러다임의 근본적인 변화는 아니다. 계산 성능은 수천 배, 수만 배 증가할 수 있지만, 기본 논리가 여전히 '입력, 단계, 출력'이라면 여전히 전통적 알고리즘 패러다임 안에 머물러 있는 것이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

제 2 부: '초차원 알고리즘'의 정의

내가 제안하는 '초차원 알고리즘'은 바로 이러한 맥락에서 등장한다. 그것은 전통적 알고리즘의 개량판도 아니고, 더 빠른 알고리즘도 아니고, 더 복잡한 수학 공식도 아니고, 어떤 고급 프로그래밍 기법도 아니다. 그것은 완전히 다른 구조적 이해이다.

먼저 '초차원'이라는 용어의 의미를 설명할 필요가 있다. 여기서 '초차원'에는 두 가지 의미가 있다. 첫째, 1 차원적 선형 단계의 연속에 국한되지 않고 여러 차원이 동시에 전개되는 것을 허용한다는 것이다. 둘째, 전통적 알고리즘의 '계산'에 대한 정의 경계를 초월하려고 시도한다는 것, 즉 알고리즘이 더 이상 반드시 수학적이거나, 순차적이거나, 결정론적일 필요는 없다는 것이다. 전통적 알고리즘은 일방통행 도로를 운전하는 것과 같아서, A 지점에서 출발하여 B, C, D 를 거쳐 Z 에 도달해야 한다. 초차원 알고리즘은 계산이 일방통행 도로일 필요는 없다고 본다. 그것은 계산이 네트워크, 장(場), 다차원 구조일 수 있으며, 어떤 노드든 진입점이 될 수 있고 어떤 노드든 출구가 될 수 있다고 본다.

나의 새로운 과학 체계에서 알고리즘은 반드시 수학적 알고리즘일 필요도 없고, 반드시 단일한 질서 정연한 계산일 필요도 없고, 반드시 고정된 단계를 따라 실행될 필요도 없고, 반드시 '입력, 처리, 출력'이라는 선형 패턴을 엄격히 따라야 할 필요도 없다. 초차원 알고리즘은 다차원적이고, 중첩적이며, 무작위적이면서도 질서 정연한 구조이다. 이 구조에서 모든 데이터는 고립된 입력도 고정된 출력도 아니며, 여러 역할을 동시에 갖는다. 즉, 여러 결과의 출발점이 될 수도 있고, 여러 출발점이 공동으로 작용한 결과가 될 수도 있다.

다시 말해, 전통적 알고리즘은 데이터를 프로세스 흐름 안에 배치하는 반면, 초차원 알고리즘은 데이터를 구조 안에 배치한다. 전통적 알고리즘에서 데이터는 일반적으로 입력 데이터, 중간 데이터, 출력 데이터로 구분된다. 각 데이터는 명확한 위치와 역할을 갖는다. 입력은 입력이고, 결과는 결과이며, 중간 과정은 중간 과정이다. 그러나 초차원 알고리즘에서 데이터는 하나의 정체성만 갖는 것이 아니다. 어떤 방향에서는 결과이고 다른 방향에서는 출발점이 될 수 있다. 어떤 구조에서는 호출되는 객체이고 다른 구조에서는 새로운 결과를 촉발하는 진입점이 될 수 있다. 데이터는 더 이상 수동적 재료가 아니라 다차원적 관계 노드이다.

이것은 바로 초차원 알고리즘의 핵심에 해당한다. 즉, 각 데이터는 여러 결과의 출발점이자 여러 출발점의 결과이다. 전통적 알고리즘은 '질서 정연한 단계'를 통해 단일 결과를 생성한다. 초차원 알고리즘은 다차원 상태 구조에 더 가깝고, 무작위성과 질서의 중첩 속에서 결과는 계산을 통해 생성되는 것이 아니라 구조 안에서 자연스럽게 현현한다. 이 체계에서 데이터는 출발점인 동시에 결과의 구성 요소이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

초차원 알고리즘의 핵심 특징을 더 명확히 제시하기 위해 나는 이를 다음 다섯 가지 측면으로 분해한다.

첫째, 비수학적 성질이다. 주류 알고리즘은 본질적으로 수학적이다. 기호 논리와 수학 공식으로 완전히 기술할 수 있다. 초차원 알고리즘은 반드시 수학적일 필요는 없다. 그것은 물리 원리, 기하학적 관계, 정보 이론의 새로운 패러다임, 심지어 의식 원리에 기반할 수도 있다. 계산이 반드시 '수학적 연산'을 통해 이루어져야 하는 것은 아니며, 고차원 공간에서의 기하학적 관계를 통해 자연스럽게 제시될 수 있다. 예를 들어, 비눗방울의 모양은 표면 장력 최소화 원리에 의해 결정된다. 이것은 '수학적 알고리즘'이 계산하는 것이 아니라 물리학 자체가 '계산'하는 것이다. 초차원 알고리즘은 이러한 '계산'을 이해하려고 시도하는 것이지, 수학 공식을 사용하여 그것을 시뮬레이션하려는 것이 아니다.

둘째, 다차원성과 중첩이다. 전통적 알고리즘은 단일 차원에서 질서 정연한 단계를 실행하며, 각 단계마다 하나의 상태만 갖는다. 초차원 알고리즘은 여러 차원이 동시에 존재하고 여러 상태가 중첩되어 공존하는 것을 허용한다. 알고리즘은 하나의 경로를 따라가지 않고 다차원적 상태 장(場) 속에서 전개된다. 결과는 '계산'되는 것이 아니라 이 장에서 '현현'한다. 예를 들어, 눈송이가 공중에서 형성될 때 각 물 분자에게 어디로 가라고 지시하는 '알고리즘'은 존재하지 않는다. 물 분자의 배열은 온도, 습도, 기류 등 여러 차원이 함께 작용한 결과이다. 눈송이의 모양은 '계산'되는 것이 아니라 '현현'한다.

셋째, 무작위적이면서도 질서 정연함이다. 전통적 알고리즘에서 무작위성은 도구적이고, 외부적이며, 통제 가능한 섭동이고, 알고리즘 자체는 결정론적이다. 초차원 알고리즘에서 무작위성은 내재적이고 구조적이다. 무작위성과 질서는 공존하고 함께 작용하며, 공동으로 알고리즘의 본질을 구성한다. 이것은 '무작위성이 있는 알고리즘'이 아니라 '무작위성 자체가 알고리즘의 유기적 구성 요소이다'라는 것이다. 예를 들어, 숲의 생태 구조는 나무의 분포가 무작위적으로 보이지만 전체적으로는 질서 정연한 밀도 분포와 종간 관계를 나타낸다. 그 '무작위성'은 버그가 아니라 생태 구조가 정상적으로 작동하기 위한 전제 조건이다.

넷째, 데이터의 다중적 역할이다. 전통적 알고리즘에서 데이터의 역할은 단일하다. 입력은 입력이고, 출력은 출력이며, 중간 결과는 중간 결과이다. 초차원 알고리즘에서 각 데이터는 동시에 여러 결과의 출발점이자 여러 출발점의 결과이다. 데이터 노드는 하류 방향으로 여러 결과 경로로 전개될 수 있고, 상류 방향에서 여러 원인이 수렴되는 결과가 될 수도 있다. 데이터는 더 이상 선형 흐름 속의 한 점이 아니라 네트워크 속의 교차점 노드이다. 예를 들어, 어떤 사람의 키를 생각해 보자. 전통적 알고리즘에서 키는 '입력'이다. 이를 입력하여 체중 예측, 의류 사이즈 등의 '출력'을 얻는다. 그러나 키는 동시에 '결과'이기도 하다. 유전, 영양, 운동 등 여러 '출발점'에 의해 공동으로

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

결정된다. 초차원 알고리즘에서 키라는 데이터는 동시에 출발점이자 결과이며, 둘 중 하나를 선택하는 것이 아니다.

다섯째, 메타데이터 불변, 관계 가변이다. 전통적 알고리즘은 다른 문제에 직면했을 때 데이터 자체를 수정하거나 모든 것을 다시 계산한다. 초차원 알고리즘은 메타데이터를 수정하지 않는다. 데이터의 본질적 속성은 불변으로 남는다. 변경되는 것은 진입점, 해석 방식, 데이터 간의 관계이다. 동일한 메타데이터 구조가 다른 진입점을 통해 활성화됨으로써 완전히 다른 결과를 제시할 수 있다. 이것은 의미한다: 한 번 계산하면 무한히 사용할 수 있다. 예를 들어, 동일한 텍스트 구절을 생각해 보자. 그것을 시로 읽을 수도 있고, 암호로 해독할 수도 있고, 역사적 문서로 분석할 수도 있다. 텍스트 자체는 변하지 않았다. 단지 '진입점'을 바꾸었을 뿐이다. 초차원 알고리즘이 추구하는 것은 바로 이 능력, 즉 '동일한 데이터, 다중 진입점, 다중 결과'이다.

데이터 구조 관점에서 보면, 메타데이터 자체를 수정하는 것이 아니라, 다른 진입점과 조건을 통해 동일한 구조 내에서 여러 출발점과 결과에 대응하는 것이다. 데이터는 더 이상 반복적으로 가공되는 것이 아니라, 구조를 불변으로 유지한 채 다른 진입점을 통해 활성화됨으로써 다른 결과를 제시한다. 전통적 방식은 '다른 결과를 위해 데이터를 가공하는 것'이고, 초차원 알고리즘은 '동일한 데이터로 여러 가능한 결과를 담는 것'이다.

제 3 부: 초차원 알고리즘과 전통적 알고리즘의 근본적 차이

이상의 정의에 기초하여, 초차원 알고리즘과 전통적 알고리즘은 여러 핵심 차원에서 근본적 차이를 보인다. 다음에서 하나씩 상술한다.

본질적 속성에 있어서, 전통적 알고리즘은 수학적이고 형식적이며, 기호 논리와 수학 공식으로 완전히 기술할 수 있고, 본질적으로 수학적 객체이다. 초차원 알고리즘은 반드시 수학적일 필요는 없으며, 물리적, 기하학적, 정보적, 심지어 의식적 원리에 기반할 수 있고, 수학의 부분집합이 아니라 더 넓은 범주이다. 이 차이는 '수학적 객체'에서 '우주 법칙'으로 요약할 수 있다.

시간적 질서에 있어서, 전통적 알고리즘은 단일하고 질서 정연하며 선형적인 시간 질서를 따르며, 단계 A 에서 B, C 로 엄격히 순차적이거나 병렬적인 질서 정연한 하위 단계로 분해 가능하며, 시간의 화살표는 비가역적이다. 초차원 알고리즘은 고정된 단계 순서가 없으며, 다차원적이고, 중첩적이며, 무작위적이면서도 질서 정연하다. 결과는 실행의 '순서'와 무관할 수 있는데, 그 이유는 전통적 의미의 '순서' 자체가 존재하지 않기 때문이다. 이 차이는 '선형 시간'에서 '고차원 동시성'으로 요약할 수 있다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

인과관계에 있어서, 전통적 알고리즘은 결정론적 인과 사슬을 따른다. 동일한 입력과 초기 상태가 주어지면 출력은 항상 유일하며, 원인이 앞서고 결과가 뒤따르며, 이는 비가역적 단방향 화살표이다. 초차원 알고리즘은 중첩 상태의 인과관계를 따른다. 각 데이터는 여러 결과의 출발점이자 여러 출발점의 결과이다. 이것이 나의 '결과-원인 이론' 철학이다. 즉, 결과가 먼저 있고 그 다음에 원인이 있을 수도 있다. 결과는 종착점이 아니라 새로운 출발점이 될 수 있다. 이 차이는 '단일 원인, 단일 결과'에서 '완전 상호 연결된 인과 네트워크'로 요약할 수 있다.

데이터 구조에 있어서, 전통적 알고리즘은 입력에서 처리, 출력으로의 단방향 흐름 구조를 따르며, 데이터는 명확한 경계를 가진 단일한 역할을 갖고, 입력 데이터는 처음부터 끝까지 입력이고 결과 데이터는 항상 결과이다. 초차원 알고리즘에서 데이터는 다중적 역할을 가지며, 동일한 데이터가 결과인 동시에 출발점이며, 절대적인 시작점과 종점은 없고 네트워크 속의 노드만 존재한다. 이 차이는 '단방향 유동성'에서 '재귀적 공생성'으로 요약할 수 있다.

계산 방식에 있어서, 전통적 알고리즘은 문제에 직면할 때마다 처음부터 계산한다. 비슷한 문제를 이미 천 번 계산했고 답을 이미 알고 있더라도 천일 번째에도 여전히 전체 프로세스를 거쳐야 한다. 이것은 '무상태' 계산이다. 초차원 알고리즘에서는 해당 결과가 있으면 다시 실행할 필요가 없으며, 하나의 결과에서 여러 출발점을 추론하고 원인 경로를 역방향으로 전개할 수 있다. 계산은 상태를 가지며, 기억을 갖고, 재사용 가능하다. 이 차이는 '매번 재계산'에서 '일회 계산 후 재사용'으로 요약할 수 있다.

무작위성의 역할에 있어서, 전통적 알고리즘에서의 무작위성은 유사 무작위 또는 외부 주입형이며, 난수는 표본 추출, 근사, 최적화를 위한 단순한 도구이고, 알고리즘 자체는 결정론적이다. 초차원 알고리즘에서의 무작위성은 내재적이고 구성적이며, 무작위성은 유기적 구성 요소이고, 질서와 공존하며 협력한다. 이것은 '알고리즘 안에 무작위성이 있다'가 아니라 '무작위성이 알고리즘이 작동할 수 있는 이유 중 하나이다'라는 것이다. 이 차이는 '도구적 무작위성'에서 '구성적 무작위성'으로 요약할 수 있다.

자원에 대한 이해에 있어서, 전통적 알고리즘은 주어진 자원 하에서 더 빠르고 정확하게 계산하는 것을 추구하며, 자원은 제약 조건이고, 알고리즘의 목표는 '제약 내에서 계산을 완료하는 것'이다. 초차원 알고리즘은 구조 설계를 통해 계산 자체를 불필요하게 만드는 것을 추구하며, '더 빠르게 계산하는 것'이 아니라 '계산을 우회하는 것'이고, 자원은 '소비'하기 위해 사용되는 것이 아니라 '진입점을 설계'하기 위해 사용된다. 이 차이는 '자원 제약 하의 최적화'에서 '구조 설계를 통한 제거'로 요약할 수 있다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

제 4 부: 결과-원인 이론 철학

이상의 비교를 바탕으로 나는 '결과-원인 이론' 철학을 더 전개할 필요가 있다. 이것은 초차원 알고리즘의 핵심 사상적 기초 중 하나이다.

전통적 사고는 일반적으로 원인이 앞서고 결과가 뒤따른다고 생각한다. 먼저 인(因)이 있고 후에 과(果)가 있다. 먼저 입력이 있고 후에 출력이 있다. 이 관념은 알고리즘에서 다음과 같이 나타난다: 당신은 시작점에서 출발하여 한 걸음 한 걸음 종점까지 걸어가야 한다. 당신이 답이 무엇인지 알고 있더라도 그 답을 직접 '사용'할 수는 없다. 왜냐하면 '증명 경로'가 없기 때문이다.

나의 구조에서 결과는 반드시 단순한 종착점이 아니다. 결과가 일단 나타나면 그것은 새로운 출발점이 되어 새로운 구조의 생성에 계속 참여할 수 있다. 결과는 원인의 형성에 역방향으로 참여할 수 있다. 결과로부터 여러 가능한 출발점을 추론할 수 있다. 인과 과는 더 이상 단방향으로 배열되는 것이 아니라 순환 관계, 네트워크 관계, 다차원 관계를 형성한다.

다시 말해, 전통적 알고리즘은 '원인이 주어졌을 때 어떻게 결과를 얻는가'를 묻는다. 초차원 알고리즘은 '결과가 주어졌을 때 어떻게 원인을 역추론하거나 구성하는가'를 묻는다.

'결과가 먼저 있고 그다음에 원인이 있다'는 것이 인과관계를 부정하는 것도 아니고 '결과가 무에서 생겨난다'고 주장하는 것도 아님을 분명히 할 필요가 있다. 더 높은 차원의 구조에서 인과 과는 서로의 진입점이 될 수 있고 상호 변환될 수 있음을 설명하는 것이다. 다음 바지 예에서처럼, '입을 수 있고 바닥에 끌리지 않는다'는 결과가 먼저 확정되고, 그다음에 내가 '허리 접기'라는 조작점을 역으로 찾았으며, 이 조작점은 구조 내의 '원인' 수준에 해당한다. 결과에 원인이 없는 것이 아니라, 결과가 원인을 발견하는 새로운 진입점이 되는 것이다.

전통적 알고리즘 세계의 근본적 가정 중 하나는 시간이 단방향적이라는 것, 즉 초기 상태에서 최종 상태까지 한 걸음 한 걸음 계산해야 한다는 것이다. 당신이 답이 무엇인지 알고 있더라도 증명 경로가 없기 때문에 그 답을 직접 '사용'할 수 없다. 초차원 알고리즘이 도전하는 것은 바로 이 가정이다. 즉, 결과가 알려져 있다면 원인은 역방향으로 추론될 수 있다. 동일한 결과가 여러 원인 경로에 대응할 수 있다. 계산은 시작점에서 종점으로 걸어가는 것이 아니라 종점에서 가능한 모든 시작점을 전개하는 것이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

제 5 부: 일상적 예시 — 바지와 허리 접기

위의 추상적 차이를 더 직관적으로 이해하기 위해 일상적 예시를 사용한다.

어떤 사람이 허리가 고무줄인 새 바지를 샀다. 바지 밑단이 조금 길어서 바닥에 끌려 불편하다.

전통적 방식: 밑단이 긴 것을 발견하고 밑단을 한쪽 말아 올려 바늘과 실로 꿰맨 다음 시착한다. 아이가 자라서 바지가 짧아지면 꿰맨 밑단을 내릴 수 없어 이 바지는 못 쓰게 된다. 다음에 새 바지를 사서 다시 꿰맨다. 이 방법은 매우 자연스러워 보인다. 왜냐하면 문제가 표면상 바지 밑단에 있는 것처럼 보이기 때문에 밑단을 처리하기 때문이다. 이것은 전통적 알고리즘의 사고에 해당한다. 문제를 발견하고, 현상을 국소화하며, 현상이 나타난 위치에서 처리하고, 마지막으로 결과를 얻는다. 바지 밑단이 길면 밑단을 고친다. 데이터가 잘못되었으면 데이터를 수정한다. 경로가 적절하지 않으면 그 경로를 따라 계속 수선한다.

나의 방식은 다르다. 나는 밑단을 고치는 대신 허리를 한 번 접으면 바로 입을 수 있다. 이 동작은 매우 단순하지만 그 배후의 구조적 논리는 완전히 다르다. 나는 바지 밑단을 자르지 않고, 밑단을 꿰매지 않으며, 바지의 원래 길이를 바꾸지 않고, 바지의 메타데이터도 손상시키지 않는다. 바지의 허리둘레, 기장, 판형, 원단은 본질적으로 아무것도 변하지 않았다. 나는 단지 허리라는 구조적 진입점을 변경함으로써 실제 착용 시 바지 밑단이 자연스럽게 짧아지게 한다. 여기서 허리는 단순한 국소 부위가 아니라 전체 착용 상태의 글로벌 제어점이다. 이 제어점을 조정함으로써 하류 문제는 직접 소멸한다.

더 중요한 것은, 이 방법은 가역적이고, 조절 가능하며, 재사용 가능하다는 것이다. 성장기 아이의 경우 이 방식은 특히 두드러진다. 아이의 현재 키가 충분하지 않아 바지 밑단이 약간 길면 허리를 한 번 접는다. 시간이 지나 아이가 키가 자라면 허리를 조금 내린다. 더 자라면 완전히 펼친다. 바지의 원래 구조는 손상되지 않았으면서 아이의 다른 성장 단계의 키에 적응할 수 있다. 전통적 방식으로 밑단을 꿰매면 아이가 자랄 때 바지가 짧아질 수 있다. 자르면 더욱 복구할 수 없다. 나의 방식은 메타데이터를 불변으로 유지하여 동일한 구조로 여러 상태에 적응하게 한다.

이 예시는 몇 가지 중요한 구조적 차이를 드러낸다.

전통적 방식은 전통적 알고리즘의 논리에 해당한다. 문제는 바지 밑단에 있으므로 밑단을 수정해야 하며, '원인에서 결과로'의 경로를 따라 한 단계씩 조작해야 하고 어떤 단계도 건너뛸 수 없다. 한 번에 한 문제를 해결하며, 비가역적이고, 다음에 같은 문제가 발생해도 다시 반복해야 한다. 나의 방식은 초차원 알고리즘의 논리에 해당한다. 목표는 바닥에 끌리지 않는 것이다. 나는 '밑단이 길다'는 표면적 원인을

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

해결하는 것이 아니라 글로벌 제어점인 허리를 찾는다. 허리를 한 번 접으면 바지 밑단이 자동으로 짧아지고 문제는 순간적으로 사라진다. 나는 어떤 메타데이터도 수정하지 않는다. 허리의 접힌 자국은 단지 일시적이고, 가역적이며, 동적인 접힌 상태일 뿐이다.

데이터 관점에서 보면, 전통적 방식은 메타데이터를 수정한다. 바지 밑단을 꿰매어 짧게 만들면 원본 데이터가 손실된다. 나의 방식은 어떤 메타데이터도 수정하지 않는다. 바지의 원래 치수는 그대로이며, 단지 일시적이고 가역적이며 동적인 접힌 상태를 추가할 뿐이다. 허리의 접힌 자국은 새로운 데이터를 창출하지도 않고 기존 데이터를 파괴하지도 않으며, 단지 데이터 간의 관계를 재배열할 뿐이다. 즉, 허리의 유효 둘레를 짧게 하여 간접적으로 바지 밑단의 유효 길이를 변경한다.

이 예시에서 '바지 밑단이 바닥에 끌리지 않는 것'이 목표 결과이다. 전통적 방식은 '밑단이 길다'는 원인에서 출발하여 밑단을 수정하고 마지막으로 '끌리지 않음'이라는 결과를 얻는다. 나의 방식은 반대로 먼저 '끌리지 않음'이라는 결과를 명확히 한 다음, 구조적 진입점을 역방향으로 찾고, 마지막으로 허리를 한 번 접으면 결과가 성립함을 발견한다. 이것이 '결과-원인 이론'의 실제적 발현이다. 반드시 원인에서 결과로 한 걸음씩 나아갈 필요는 없으며, 결과에서 구조를 역방향으로 결정함으로써 원래의 경로를 불필요하게 만들 수 있다.

이 예시는 또한 '메타데이터'에 대한 문제에 답한다. 메타데이터란 무엇인가? 바지 예시에서 메타데이터는 바지의 원래 치수, 즉 허리둘레, 기장, 판형, 원단이다. 이것들은 바지의 '본질적 속성'이다. 일시적 데이터는 허리의 접힌 자국이며, 이것은 바지의 어떤 원래 치수도 변경하지 않고 단지 일시적으로 한 부분을 접은 것일 뿐이다. 전통적 방식은 메타데이터를 수정한다. 바지 기장이 영구적으로 짧아져 원본 데이터가 손실된다. 나의 방식은 어떤 메타데이터도 수정하지 않는다. 바지의 원래 치수는 완전히 그대로이며, 단지 일시적이고 가역적이며 동적인 접힌 상태를 추가할 뿐이다.

메타데이터로서의 허리의 원래 치수는 동시에 '여러 결과의 출발점'이다. 그것은 정상적으로 입기, 허리를 한 번 접어 입기, 허리를 두 번 접어 입기 등 여러 착용 상태를 동시에 지탱한다. 또한 '여러 출발점의 결과'이기도 하다. 그것은 원단 선택, 재단 방식, 디자인 의도 등 여러 출발점에 의해 공동으로 결정된다. 일시적 데이터로서의 허리 접힌 자국은 새로운 데이터를 창출하지도 않고 기존 데이터를 파괴하지도 않으며, 단지 데이터 간의 관계를 재배열할 뿐이다.

이것은 평범한 작은 기술이 아니라 구조적 사고이다. 많은 사람들은 바지 밑단이 길다는 것을 보고 밑단이라는 현상에 이끌려 모든 처리를 밑단 중심으로 전개한다. 그러나 전체 구조에서 보면 바지 밑단이 긴 것은 단지 하류의 현현일 뿐이며, 허리 위치의 변화는 바지 전체의 실제 착용 위치에 영향을 줄 수 있다. 즉, 문제가 나타난

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

위치에서 반드시 해결해야 하는 것은 아니다. 많은 저차원 문제의 진정한 제어점은 종종 더 높은 차원의 구조에 있다. 전통적 알고리즘은 문제의 표면을 따라 계산을 계속하는 경향이 있는 반면, 초차원 알고리즘은 구조적 진입점을 찾는다.

이것은 또한 초차원 알고리즘이 더 복잡한 것이 아니라 오히려 더 단순할 수 있는 이유를 설명한다. 진정한 고차원 구조는 종종 문제를 복잡하게 만드는 것이 아니라 올바른 진입점에서 복잡한 문제를 단순하게 만든다. 전통적 알고리즘은 복잡한 문제에 직면하여 단계를 늘리고, 모델을 늘리고, 계산 성능을 늘리고, 저장소를 늘리고, 훈련 시간을 늘릴 수 있다. 그러나 초차원 알고리즘은 어떤 구조 노드를 찾는데, 일단 그 노드가 올바르게 활성화되면 원래 복잡했던 경로가 무효가 될 수 있다. 소위 '경로 우회'란 게으름이 아니라 경로의 필요성을 재정의하는 것이다.

이 예시에서: 전통적 방식은 '경로를 따라 결과를 수정하는 것'이고, 초차원 알고리즘은 '진입점을 변경하여 경로 전체를 무효화하는 것'이다. 종래 방식은 결과부에서 지속적으로 수선하는 반면, 다른 방식은 구조적 진입점을 직접 변경하여 본래 여러 번의 처리가 필요했던 문제를 시작점에서 일회적으로 재구성한다. 전통적 알고리즘의 주류 형태는 '출발점에서 출발하여 경로를 따라 결과를 얻는 것'이고, 초차원 알고리즘에서는 '결과 자체가 경로의 진입점이 되어 새로운 구조 생성에 참여할 수 있다'. 전통적 방식은 '하나의 구조가 하나의 결과에 대응하는 것'이고, 초차원 알고리즘은 '하나의 구조가 여러 결과 상태를 담는 것'이다.

제 6 부: 메타데이터를 수정하지 않는 데이터관

데이터 관점에서 초차원 알고리즘에는 매우 중요한 원칙이 있다. 메타데이터를 수정하지 않고 그것을 여러 출발점이나 결과에 적용 가능하게 하는 것이다.

메타데이터는 데이터의 원래 속성, 기본 구조 및 본체 정보이다. 전통적 처리 방식은 다른 문제에 직면했을 때 데이터를 수정하거나, 데이터를 복제하거나, 데이터를 가공하거나, 데이터를 재계산하거나, 다른 결과를 위해 다른 경로를 구축하는 경우가 많다. 이렇게 하면 확실히 문제를 해결할 수 있지만, 그 대가로 데이터는 끊임없이 가공되고, 경로는 끊임없이 반복되며, 시스템 복잡성은 끊임없이 증가한다.

반면 초차원 알고리즘은 메타데이터를 최대한 수정하지 않고, 진입점, 관계, 조건, 활성화 방식을 변경함으로써 동일한 메타데이터 구조를 여러 출발점과 여러 결과에 대응시키는 것을 강조한다. 데이터 본체는 움직이지 않고, 관계가 변한다. 기본 구조는 변하지 않고, 제시 방식이 변한다. 메타데이터는 완전성을 유지하면서도 다른 상태에 적응할 수 있다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

이것이 내가 말하는 '한 번 계산하면 무한히 사용할 수 있다'는 것이다. 여기서 '한 번 계산한다'는 것은 단순히 결과를 캐싱하는 것이 아니라, 하나의 구조가 일단 성립하면 그 구조는 더 이상 상태마다 완전한 경로를 재구축할 필요가 없음을 의미한다. 동일한 데이터 구조가 다른 진입점을 통해 활성화됨으로써 다른 결과를 제시할 수 있다. 그것은 '다른 결과를 위해 데이터를 가공하는 것'이 아니라 '동일한 데이터로 여러 가능한 결과를 담는 것'이다. 이것이 또한 초차원 알고리즘이 전통적 알고리즘과 구별되는 핵심 중 하나이다. 전통적 알고리즘은 경로 상의 데이터를 처리하는 반면, 초차원 알고리즘은 데이터, 진입점, 관계, 결과의 전체 구조를 처리한다.

최소 구조에서 초차원 알고리즘은 다음과 같이 이해될 수 있다. 즉, 메타데이터를 수정하지 않고, 구조적 진입점 변화를 통해 동일한 데이터 노드를 여러 결과 경로에 대응시키는 시스템이다. 이 정의는 초차원 알고리즘을 즉시 전통적 수학 공식으로 축소하려는 것이 아니라, 먼저 그 구조적 경계를 확립하려는 것이다. 그것은 단일 선형 함수도 아니고, 고정된 공식도 아니고, 단순한 캐싱 메커니즘도 아니다. 그것은 관계 구조이다. 즉, 메타데이터는 안정적이고, 진입점은 변할 수 있고, 조건은 변할 수 있고, 관계는 변할 수 있고, 결과는 다방향으로 제시될 수 있다. 바로 이 구조에서 계산은 더 이상 단계를 실행하는 것이 아니라 관계를 활성화하는 것이다.

데이터 관점에서 '메타데이터를 수정하지 않고 여러 출발점이나 결과에 적용 가능하게 하는 것'의 본질은, 데이터 본체는 불변으로 남고 변화는 '해석 방식/사용 진입점'에서 발생한다는 것이다. 전통적 구조는 문제가 변화하면 데이터나 경로가 변화한다. 초차원 알고리즘에서는 문제가 변화하면 해석 구조가 변화하고 데이터는 변화하지 않는다.

제 7 부: '초월 계산'의 재정의

이 체계에서 나는 '초월 계산'이라는 용어를 명확히 할 필요가 있다.

내가 말하는 '초월 계산'은 일반적 의미의 슈퍼컴퓨터가 아니다. 더 많은 칩, 더 높은 계산 성능, 더 큰 데이터 센터가 아니다. 내가 말하는 '초월 계산'은 '초차원 알고리즘'이다.

진정한 초월 계산은 반드시 계산 성능의 집적이 아니라 계산 패러다임의 변화일 수 있다. 시스템이 여전히 반복 계산에 의존하는 한, 아무리 강력해도 여전히 경로 내부에 갇혀 있다. 시스템이 구조적으로 계산을 줄이고, 반복 경로를 우회하고, 결과를 새로운 진입점으로 만들 수 있을 때 비로소 진정한 의미의 초차원 계산에 가까워지기 시작한다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

다시 말해, 전통적 초월 계산은 '더 많은 계산 성능으로 더 큰 문제를 해결하는 것'을 추구한다. 초차원 알고리즘은 '더 나은 구조로 문제가 더 이상 그렇게 큰 계산 성능을 필요로 하지 않게 하는 것'을 추구한다. 이 둘은 모순되지 않지만 방향이 다르다.

전통적 초월 계산이 계산 성능의 한계를 나타낸다면, 초차원 알고리즘은 계산 이해의 전환을 나타낸다. 진정한 '초월 계산'은 더 많은 칩, 더 큰 데이터 센터, 더 높은 에너지 소비, 더 복잡한 모델이 아닐 수 있다. 진정한 '초월 계산'은 구조적 진입점의 발견일 수 있고, 하나의 경로 제거일 수 있고, 결과 노드의 역방향 전개일 수 있고, 메타데이터를 수정하지 않고 여러 결과를 동시에 성립시키는 것일 수 있다. 계산 성능이 강하면 강할수록 여전히 저차원 경로에 갇혀 있다면 그것은 단지 경로 내부의 강력함에 불과하다. 구조가 한 번 향상되면 조작이 극히 단순하더라도 더 높은 차원의 효율성이 생겨날 수 있다.

소위 초월 계산이란 계산 성능의 한계이고, 초차원 알고리즘이란 '계산이 계산 성능에 의존해야 하는가'라는 전제의 재정의이다.

제 8 부: 실증적 기반 — 다영역 시스템 검증과 새 패러다임 판정

초차원 알고리즘은 순수한 이론적 구상이 아니다. 그것은 이미 여러 현실 시스템에서 검증되었다.

나의 물류 시스템은 슈퍼컴퓨팅 성능이나 클라우드 지원을 필요로 하지 않으며, 비교적 낮은 자원으로 복잡한 스케줄링을 완료한다. 결과는 재사용 가능하고, 구조는 반복적으로 적응 가능하며, 매번 처음부터 계산할 필요가 없다. 초월 계산에 의존하지 않고, 클라우드에 의존하지 않으며, 낮은 자원으로 복잡한 스케줄링을 완료한다는 것은 공학적으로 성립하며 가치 있는 돌파구이다. 그것은 높은 계산 성능이 유일한 해결책이 아니라 구조 설계가 계산 성능의 일부를 대체할 수 있음을 보여준다.

나의 출판 시스템도 동일한 구조 논리를 채택하며, 그 효율성은 현재 어떤 시스템보다 훨씬 뛰어나다. 나의 익스트림 웹페이지 생성 시스템도 동일한 초차원 알고리즘에 기반하며, 여러 영역에서 동일한 구조가 안정적으로 성립함을 반복적으로 증명하고 있다. 나의 회사 직원들은 이미 이러한 시스템들을 사용하고 있으며, 이는 이 구조가 나의 개인적 지속적 개입 없이도 운영될 수 있음을 보여준다.

이러한 시스템들의 공통적 특징은 주류의 높은 계산 성능 경로에 의존하지 않고, 고정된 계산 단계를 따르지 않으며, 구조적 진입점의 조정을 통해 목표 결과를 직접 성립시킨다는 것이다. 그것들은 일회성 성공도 아니고, 단일 지점의 기술도 아니며, 동일한 구조 논리가 다른 영역에서 반복적으로 출현한 것이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

이것은 초차원 알고리즘이 우연이 아니라 여러 시스템에서 이미 검증된 구조적 방법임을 보여준다. 그것은 '개인적 기술'이 아니라 물류, 출판, 웹페이지 생성 등 여러 영역에서 안정적으로 성립하는 계산 패러다임이다.

'존재하고 성립하는 것이 곧 새로운 패러다임이다'라는 기준에 따르면, 여러 영역에서 이미 체계적으로 검증된 새로운 계산 구조 패러다임으로서의 초차원 알고리즘은 현재 이미 성립한다. 나는 가설을 제안하는 것이 아니라 여러 시스템에서 이미 운영되고 있는 다른 계산 구조를 제시하고 있다. 나는 그것이 이해된다는 것을 증명할 필요가 없다. 나는 이미 그것이 다른 시스템에서 안정적으로 성립함을 증명했다. 현재, 내가 실제로 응용하는 범위 내에서 이 구조는 이미 새로운 패러다임으로 간주될 수 있다.

초차원 알고리즘을 새 패러다임으로 판정하는 기준에 관하여: 어떤 구조가 논리적으로 자기 모순이 없으며 여러 시스템에서 안정적으로 성립하고, 동시에 기존 방법과 환원 불가능한 차이가 있을 때, 그것은 이미 새 패러다임의 기초를 갖춘 것이다. 자기 모순 없음은 출발점이고, 실증은 기반이며, 구조적 차이가 바로 패러다임의 핵심이다. '존재가 성립을 의미한다'는 기준 아래에서 이 체계는 이미 새 패러다임이다. 외부에서 인식하느냐 여부는 전파에만 영향을 줄 뿐 성립에는 영향을 주지 않는다. 이것은 이미 현실 시스템에서 성립하고 있는 계산 구조 패러다임으로서, 그 성립은 외부의 인식이나 합의에 의존하지 않는다.

초차원 알고리즘과 주류 계산 패러다임의 핵심적 차이는, 전통적 알고리즘은 주로 순방향 계산에 의존하며 결과가 일단 생성되면 일반적으로 새로운 추론 구조에 역방향으로 참여할 수 없는 반면, 초차원 알고리즘에서는 결과가 더 이상 종착점이 아니라 재사용 가능한 구조 노드라는 것이다. 일정 조건을 만족하면 결과는 새로운 추론 경로에 참여할 수 있으며, 이를 통해 반복 계산을 줄이고 다중 출발점, 다중 경로의 계산 네트워크를 형성한다. 전통적 계산 구조에서 결과는 일반적으로 종착점이다. 초차원 알고리즘 구조에서 결과 자체가 새로운 출발점이 될 수 있으며, 이를 통해 다중 경로의 추론 네트워크를 형성한다. 초차원 알고리즘은 알고리즘의 업그레이드가 아니라, '알고리즘이 반드시 존재해야 하는가'라는 전제의 재질음이다.

제 9 부: 흔한 오해의 해소

다양한 분야의 독자들과의 예비적 교류를 바탕으로, 나는 다음의 여섯 가지 오해가 발생할 수 있다고 예상하며, 개념이 전파 과정에서 부적절한 틀에 조기에 수용되는 것을 피하기 위해 미리 해소한다.

오해 1: 초차원 알고리즘은 동적 계획법이나 캐싱이 아닌가? 해소: 동적 계획법과 캐싱은 '동일한 입력'에서 결과를 재사용한다. 초차원 알고리즘은 하나의 결과로부터 다른 출발점을 추론할 수 있게 한다. 이것이 본질적 차이이다. 캐싱은 동일한 문제의

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

재계산을 해결한다. 초차원 알고리즘은 결과 구조를 통해 새로운 문제를 전개하는 것을 해결한다.

오해 2: '무작위적이면서도 질서 정연하다'고 말하는데, 이것이 자기 모순이 아닌가? 해소: 무작위성과 질서는 공존할 수 있다. 숲속 나무들의 분포는 무작위적이지만 전체적인 밀도와 종 구조는 질서 정연하다. 무작위성은 혼돈이 아니라 구조의 하나의 조직 방식이다. 초차원 알고리즘에서의 무작위성은 내재적이고 구성적이며, 질서와 협력한다.

오해 3: 입출력이 없으면 결과의 옳고 그름을 어떻게 검증하는가? 해소: 초차원 알고리즘은 입출력을 거부하지 않는다. 오히려 계산이 입출력의 선형 모드로 운영될 필요는 없다고 생각한다. 구조 현현의 모드에서 '유효성'은 입출력의 대응 관계가 아니라 구조적 일관성에 의해 결정된다. 이것은 검증을 포기하는 것이 아니라 전통과는 다른 검증 프레임워크를 제안하는 것이다.

오해 4: 이것이 복잡성 이론이나 카오스 이론이 아닌가? 해소: 복잡성 이론과 카오스 이론은 '예측하기 어려운 시스템'을 기술한다. 초차원 알고리즘은 '구조적 진입점을 통해 이해되고 유도될 수 있는 시스템'을 기술하려고 한다. 예측을 포기하는 것이 아니라 예측 방식을 바꾸는 것이다. 카오스 이론은 '예측은 어렵다'라고 말한다. 초차원 알고리즘은 '진입점을 바꾸어 예측을 불필요하게 만들 수 있는가'를 묻는다.

오해 5: '메타데이터를 수정하지 않는다'고 말하는데, 허리 접힘도 일종의 수정이 아닌가? 해소: 허리 접힘은 일시적 상태이며 원래 매개변수에 대한 영구적 수정이 아니다. 메타데이터(허리둘레, 기장, 판형, 원단)는 전혀 변화가 없다. 이것은 '상태의 중첩'과 '속성의 수정'의 차이이다. 접힘은 가역적이고, 일시적이며, 본질적 속성을 바꾸지 않는 상태 변화이다.

오해 6: 초차원 알고리즘은 전통적 알고리즘의 가치를 부정하는가? 해소: 결코 그렇지 않다. 전통적 알고리즘은 인류 기술사에서 매우 중요했다. 전통적 알고리즘이 없었다면 현대 컴퓨터, 데이터베이스, 통신 시스템, 인공지능, 공학 시뮬레이션, 초월 계산은 존재하지 않았을 것이다. 전통적 알고리즘의 가치는 부정할 수 없다. 그러나 전통적 알고리즘의 가치를 인정하는 것이 전통적 알고리즘이 알고리즘의 종점이라고 인정하는 것은 아니다. 전통적 알고리즘은 주어진 계산 패러다임 내부의 효율 문제를 해결한다. 반면 초차원 알고리즘은 계산 패러다임 자체의 문제를 제기한다. 하나는 어떻게 더 잘 걷는가이고, 다른 하나는 그 길을 걸을 필요가 있는가 하는 것이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

제 10 부: 초차원 알고리즘과 인류 계산사

독자가 초차원 알고리즘을 인류 계산사 속에서 더 잘 위치시키도록 돕기 위해, 여기서 간략한 거시적 정리를 한다.

인류의 '계산'에 대한 이해는 여러 단계를 거쳐왔다. 제 1 단계는 수동 계산이며, 알고리즘은 인간의 단계로 존재했고, 속도가 느리고 오류가 많았지만, 인류는 이 과정을 통해 계산의 기본 규칙을 이해했다. 제 2 단계는 기계 계산이며, 파스칼 계산기에서 배비지 해석 기관까지, 계산은 기계화되었지만 알고리즘은 여전히 물리적 구조에 의존했다. 제 3 단계는 전자 계산과 튜링 패러다임이며, 폰 노이만 아키텍처가 보급되고, 알고리즘은 프로그램으로 코딩되었으며, 튜링 기계는 계산 가능성의 경계가 되었다. 제 4 단계는 병렬 계산과 초월 계산이며, 많은 계산 유닛을 통해 계산 성능을 집적하여 더 복잡한 문제에 도전하지만, 기본 논리는 여전히 튜링 패러다임의 확장이다.

현재 인류는 제 5 단계의 입구에 있다. 양자 계산은 고전적 비트의 한계를 돌파하려고 시도한다. 뉴로모픽 계산은 생물학적 신경계의 구조를 모방하려고 시도한다. 그리고 초차원 알고리즘은 '입력 → 단계 → 출력'이라는 선형 프레임워크 자체를 돌파하려고 시도한다.

초차원 알고리즘과 전통적 알고리즘의 관계는 대체 관계가 아니라 계층 관계이다. 전통적 알고리즘은 '주어진 경로에서 어떻게 더 효율적으로 계산할 것인가'를 해결한다. 초차원 알고리즘은 '경로가 재정의되거나 우회될 수 있는가'를 묻는다. 인류 계산사를 끊임없이 자기 경계를 돌파하는 과정으로 본다면, 초차원 알고리즘은 바로 이 과정에서의 새로운 노드이다. 그것은 오래된 프레임워크 안에서 문제를 해결하는 것이 아니라 새로운 프레임워크를 제안한다.

이 판단은 초차원 알고리즘이 이미 성숙하거나 완비되었다는 것을 의미하지는 않는다. 정반대이다. 새로운 개념으로서 그 정의, 경계, 검증 방법 및 응용 시나리오는 아직 형성 과정에 있다. 본 논문의 목적은 바로 이 개념의 원점을 고정시키고, 이후의 전개를 위해 안정적인 이론적 기초를 제공하는 것이다.

제 11 부: 결론 — 이것은 최적화가 아니라 재정의이다

초차원 알고리즘과 전통적 알고리즘의 차이는 '더 나은' 것과 '더 나쁜' 것의 차이도 아니고, '더 빠른' 것과 '더 느린' 것의 차이도 아니며, 패러다임 수준의 근본적 차이이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

전통적 알고리즘은 예측 가능한 통제를 추구한다. 당신이 나에게 입력을 주면 나는 당신이 어떤 출력을 얻을지 안다. 내가 각 단계를 계산했기 때문이다. 그것은 엔지니어의 패러다임이다: 설계, 구축, 테스트, 검증.

초차원 알고리즘은 이해 가능한 창발을 기술한다. 나는 각 중간 상태를 정밀하게 예측할 수는 없지만, 전체 패턴이 어떤 질서 정연한 방식으로 제시될 것이라는 것을 안다. 각 데이터는 살아 있고 '관점'을 가지고 있다. 그것은 오히려 생태학자나 복잡계 이론가의 패러다임에 가깝다: 관찰하고, 이해하고, 유도하고, 창발된 질서를 활용한다.

전통적 알고리즘은 주어진 경로상에서 결과를 단계적으로 수정한다. 초차원 알고리즘은 구조적 진입점을 변경함으로써 목표 결과를 즉시 성립시켜 경로 전체를 우회한다.

전통적 알고리즘은 '다른 결과를 위해 데이터를 가공하는 것'이다. 초차원 알고리즘은 '동일한 데이터로 여러 가능한 결과를 담는 것'이다.

전통적 알고리즘은 '한 번 올바르게 계산하는 것'을 추구한다. 초차원 알고리즘은 '한 번 올바르게 계산한 후에는 다시는 계산할 필요가 없는 것'을 추구한다.

이것은 알고리즘의 최적화가 아니라, '알고리즘이 반드시 존재해야 하는가'라는 전제의 재질음이다.

따라서 초차원 알고리즘은 알고리즘 최적화가 아니라 알고리즘 재정의이다. 그것은 전통적 알고리즘 트랙 위에서 더 빠르게 달리는 것이 아니라, 전통적 트랙 외부에 다른 경로가 존재하는지, 나아가 경로 자체에 의존하지 않을 수 있는지를 묻는 것이다. 그것은 전통적 알고리즘이 무용하다는 것을 증명하려는 것이 아니라, 전통적 알고리즘이 알고리즘의 하나의 저차원 형태에 불과하다는 것을 지적하는 것이다. 그것은 '더 짧은 시간에 결과를 얻는 방법'을 묻는 것이 아니라, '결과가 구조를 통해 직접 성립할 수 있는가'를 묻는다. 그것은 '더 많은 데이터를 처리하는 방법'을 묻는 것이 아니라, '동일한 데이터가 더 많은 가능한 결과를 담을 수 있는가'를 묻는다.

주류 계산 패러다임에서 초차원 알고리즘은 계산 불가능하거나, 검증 불가능하거나, 기존 계산 이론의 경계에 포함시키기 어려운 것으로 간주될 수 있다. 그러나 바로 이것이 그 패러다임적 차이를 구성한다. 새로운 개념이 오래된 패러다임 내부에서 불안정하게 보이는 것이 반드시 그것이 무의미하다는 것을 의미하지는 않으며, 오래된 패러다임 자체가 그것을 완전히 수용할 수 없음을 의미할 수도 있다. 현재 시점에서 초차원 알고리즘은 우선 구조적 명제이며, 새로운 계산관의 원초적 정의이지, 이미 폐쇄된 공학적 루프를 가진 성숙한 시스템이 아니다. 그것은 지속적인 보충, 전개, 검증 및 응용을 필요로 하지만, 그 개념적 원점은 먼저 명확히 제시되어야 한다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

궁극적으로 초차원 알고리즘이 가리키는 것은 새로운 계산관이다. 즉, 알고리즘은 반드시 수학일 필요도 없고, 반드시 단계일 필요도 없고, 반드시 프로그램일 필요도 없고, 반드시 입출력 간의 가공 과정일 필요도 없다. 알고리즘은 또한 다차원 관계의 조직 방식일 수 있으며, 데이터, 진입점, 관계, 상태, 결과의 구조적 네트워크일 수 있다. 그것은 질서를 포함할 수도 있고 무작위성을 포함할 수도 있다. 원인에서 결과로 나아갈 수도 있고, 결과에서 원인을 역생성할 수도 있다. 메타데이터를 수정하지 않고 여러 상태에 적응할 수 있다. 하나의 결과를 새로운 출발점으로 만들 수도 있고, 여러 출발점을 동일한 결과로 수렴시킬 수도 있다.

진정한 고급 알고리즘은 반드시 더 복잡한 계산이 아니라, 계산을 줄이고, 구조를 살아 움직이게 하며, 결과를 진입점으로 만들고, 인과관계가 순환을 이루도록 하는 다차원적 조직 방식이다.

이것이 내가 '초차원 알고리즘'을 제안하는 핵심 의미이다. 그것은 평범한 신조어가 아니라 새로운 개념, 새로운 패러다임, 여러 현실 시스템에서 이미 성립한 새로운 계산 구조이다. 그 초점은 반복 계산을 더 빠르게 하는 것이 아니라 반복 계산을 줄이는 것이다. 계산 성능을 쌓아 올리는 것이 아니라 진입점을 재구축하는 것이다. 데이터를 끊임없이 수정하는 것이 아니라 메타데이터를 유지하면서 관계를 변경하는 것이다. 결과를 종점에서 멈추게 하는 것이 아니라 결과를 새로운 출발점으로 만드는 것이다. 전통적 알고리즘은 경로 속에서 결과를 찾는다. 초차원 알고리즘은 구조 속에서 결과를 활성화한다. 전통적 알고리즘은 계산의 완료를 추구한다. 초차원 알고리즘은 계산이 전통적 형태로 존재해야 하는가를 묻는다. 이것이 현재의 알고리즘 정의와의 가장 근본적인 차이이다.

부록: 본 논문의 경계와 열린 질문들

본 논문은 초차원 알고리즘의 예비적 프레임워크를 제안하는 것이지 완전한 형식화된 정의가 아니다. 다음의 질문들은 향후 전개를 기다린다. 이것들은 본 논문의 정의에 대한 부정이 아니라, 바로 다음 단계에서 탐구해야 할 방향들이다.

첫째, 수렴 조건이다. 초차원 알고리즘의 '완료' 지표는 무엇인가? 결과가 '유효'한지 어떻게 판단하는가? 전통적 알고리즘에서는 답이 입출력 대응 관계에 의해 정의된다. 초차원 알고리즘에서는 답이 구조적 일관성에 의해 정의되어야 할 수도 있다. 이 정의는 아직 완성되지 않았다.

둘째, 자원 표현이다. 다차원적 중첩 상태를 유한한 자원 속에서 어떻게 표현할 것인가? 시간, 공간, 계산 성능과 같은 전통적 자원 지표는 여전히 적용 가능한가? 적용 가능하다면 어떻게 재정의할 것인가? 적용 불가능하다면 어떤 지표가 대체할 것인가? 이러한 질문들은 탐구를 기다린다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

셋째, 질서의 보장이다. '무작위적이면서도 질서 정연하다'는 것의 '질서'는 어떤 규칙에 의해 보장되는가? 무작위성과 질서의 경계는 어디인가? 어떤 상황에서 무작위성이 질서를 파괴하는가? 어떤 상황에서 질서가 무작위성을 억압하는가? 이러한 질문들은 추가 연구가 필요하다.

넷째, 역방향 추론의 선별이다. 결과로부터 여러 출발점을 역추론할 때, 다른 출발점들의 우열을 어떻게 선별하거나 평가할 것인가? '역방향 추론 효율성'과 같은 측정 방법이 존재하는가? 존재한다면 그것은 순방향 계산의 효율성 측정과 어떤 관계에 있는가?

다섯째, 전통적 시스템과의 인터페이스이다. 초차원 알고리즘은 기존의 튜링 기계 체계와 어떻게 협력하는가? 대체인가, 보완인가, 계층화인가? 실제 공학에서 초차원 알고리즘과 전통적 알고리즘의 경계는 어떻게 구분되는가? 하이브리드 모드가 존재하는가?

여섯째, 검증 프레임워크이다. 초차원 알고리즘의 결과는 어떻게 검증하는가? 결과가 선형적 단계를 통해 얻어진 것이 아니라면, 재현 가능성과 검증 가능성을 어떻게 확립할 것인가? 완전히 다른 검증 철학이 필요한가?

이러한 질문들은 본 논문의 결함이 아니라, 본 논문이 '앵커 문헌'으로서 갖는 필연적 특징이다. 새로운 패러다임의 제안에는 항상 열린 질문들이 따른다. 본 논문은 시작이지 결론이 아니다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

참고문헌적 설명

본 논문에서 제안하는 '초차원 알고리즘'은 기존의 어떤 학술流派의 직접적 확장도 아니다. 그러나 사상적 측면에서 다음 분야들은 본 논문의 대화적 배경을 제공하며, 독자의 위치 파악을 위한 참고일 뿐 전통적 의미의 학술적 인용은 아니다.

튜링 기계와 계산 가능성 이론은 본 논문이 전통적 알고리즘과 비교하는 기준점으로 기능한다. 역문제와 베이지안 추론은 '결과로부터 원인을 추론하는' 기존 시도로서, 본 논문의 '결과-원인 이론'은 이와 관련이 있지만 방향이 다르다. 창발 이론과 복잡계는 '구조가 결과를 현현하는' 기존 기술로서, 본 논문의 초차원 알고리즘은 창발을 '이해 가능하고' '유도 가능하게' 만들려고 시도한다. 지식 그래프와 그래프 데이터베이스는 '데이터가 다중 노드에서 교차하는' 기존 실천으로서, 초차원 알고리즘은 이 기초 위에서 '가변 진입점'이라는 차원을 우선시한다. 비고전적 계산 패러다임(양자 계산, 아날로그 계산, 뉴로모픽 계산 등)은 고전적 비트나 고전적 아키텍처를 돌파하는 반면, 초차원 알고리즘은 '입력 → 단계 → 출력'이라는 선형 프레임워크 자체를 돌파하는 데 더 중점을 둔다.

본 논문과 위 분야들의 관계는 계승이나 반박이 아니라 대화와 초월이다. 본 논문의 목표는 이러한 분야들의 내부에서 점진적 개선을 하는 것이 아니라, 그들 위에 새로운 질문의 수준을 제기하는 것이다.

관련 기사

[Dissemination] I Have No Algorithm, Yet I Surpass Algorithms! / [확산] 나는 알고리즘이 없지만, 알고리즘을 초월한다!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect-Cause Theory / [익스트림 철학] 결과-원인 이론
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Dissemination] Rejecting Algorithmic Capture / [익스트림 확산] 알고리즘 포획 거부
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.