On the Performance and Adoption of Search-Based Microservice Identification with `toMicroservices`

Luiz Carvalho (PUC-Rio), Alessandro Garcia (PUC-Rio), Thelma Elita Colanzi (UEM), Wesley K. G. Assunção (UTFPR), Juliana Alves Pereira (PUC-Rio), Baldoino Fonseca (UFAL), Márcio Ribeiro (UFAL), Maria Julia de Lima (Tecgraf/PUC-Rio), Carlos Lucena (PUC-Rio)

August, 2020

1 Introduction

In this document, we present additional information and results of the paper *On the Performance and Adoption of Search-Based Microservice Identification with toMicroservices* published in the *36th IEEE International Conference on Software Maintenance and Evolution.*

2 JMetal Details

We used JMetal[1] in version 5.9 to instantiate NSGA-III and additional algorithms.

3 Quality Indicators: IGD and HV

We rely on two sets of solutions: (i) PF_{known} is the set of non-dominated solutions found by an algorithm, eliminating the dominated ones, (ii) PF_{true} is conceptually known as the set with ideal solutions for a problem.

Hypervolume (HV) measures the area of the objective space from a reference point to a front of solutions. This indicator enables us to analyze both closeness and diversity of a Pareto front. In this study, we use the HV computed by a recursive and dimension-sweep algorithm[2]. To compute HV we normalized each

[1] `https://github.com/jMetal/jMetal`
[2] `http://lopez-ibanez.eu/hypervolume`

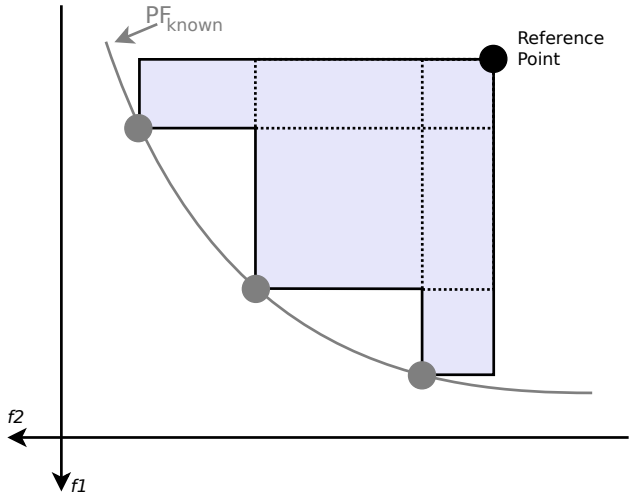

Figure 1: Hypervolume

PF_{known} between 0 and 1, and adopted a reference point with the value of 1.1 for all five objectives. Pareto fronts with high values of HV are the best since their solutions are far from the reference point. Figure 1 presents the computation of HV visually.

Inverted Generational Distance (IGD) measure the convergence/closeness between PF_{known} and PF_{true}. IGD is an indicator based on GD, but with the goal of evaluating the distance from PF_{true} to PF_{known}, i.e., the inverse of which is considered by GD. Wherever GD is an error measure used to examine the distance of the solutions found by an algorithm (PF_{know}) to the best solutions known (PF_{true}). Values of IGD closer to 0 are desired, which indicates that the solutions of both PF_{approx} and PF_{true} are close to each other. Figure 1 introduces the computation of IGD visually.

4 Interview Questions

Our interview was divided into three phases. The first phase is related to the participant knowledge about the analyzed feature in the legacy system. Table 1 presents the questions of this phase. All the questions are in four points Likert scale : (i) I don't know, (ii) I know little, (iii) I know, and (iv) I know a lot.

After, in the second phase we aimed to discover the adoptability of microservices. Table 2 shows the questions that inquire developers about the microservices generated by `toMicroservices`. The identifier (1 to 4) assigned to each microservice is chosen in a random way. The five points Likert scale is: (i) I would not adopt strongly, (ii) I would not adopt, (iii) I adopt partially, (iv) I adopt, and (v) I adopt strongly. The participant was instructed to adopt partially when modified the microservice with less than 20% of modification as a

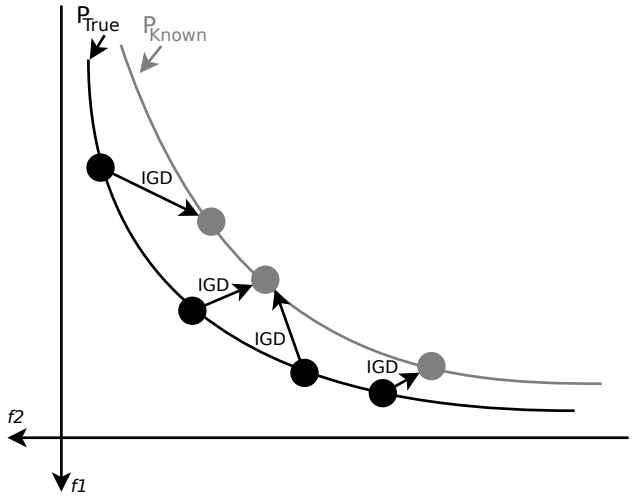

Figure 2: Inverted Generational Distance

move methods from or to another microservice.

Finally, in the third phase we inquired the participants of the interview about their previous experience as shown in Table 3.

Table 1: Measurement of the knowledge level about the feature under analysis

Question	**Response Type**
What is your level of knowledge related to the **Authentication** feature?	Four point Likert scale
What is your level of knowledge related to the **Algorithm** feature?	Four point Likert scale
What is your level of knowledge related to the **Project** feature?	Four point Likert scale

5 Interview Results

Regarding the knowledge about the features, all the median to the three features is 3 on our Likert scale, what means that the participants know the evaluated features. No developer replied that she doesn't know the feature.

Table 4 presents the results for two scenarios: Scenario-5MS and Scenario-10MS with five and ten microservice candidates, respectively. Microservices whose grades were 3, 4 or 5 (last column of Table 4) were considered (partially or

fully) adoptable.

Table 2: Adoptability questions

Question	**Response Type**
Would you adopt Microservice 1?	Five point Likert scale
Justify your answer about the adoption of Microservice 1 by pointing out the positive or negative points that you identified.	Open
Would you adopt Microservice 2?	Five point Likert scale
Justify your answer about the adoption of Microservice 2 by pointing out the positive or negative points that you identified.	Open
Would you adopt Microservice 3?	Five point Likert scale
Justify your answer about the adoption of Microservice 3 by pointing out the positive or negative points that you identified.	Open
Would you adopt Microservice 4?	Five point Likert scale
Justify your answer about the adoption of Microservice 4 by pointing out the positive or negative points that you identified.	Open
What other criteria could be observed during the process of identifying microservices for microservices architecture?	Open
What was your biggest difficulty when analyzing the proposed solutions?	Open

Table 3: Developers' background

Question	**Response Type**
What is your academic background?	Open
How long time have you been developing the software analyzed?	Years

Table 4: Results of the Qualitative Evaluation

Participant	**Years of experience in the system**	**Recognizable features**	**New recognizable features**	**Microservice's grades**
Scenario-5MS: Architectures with 5 microservices				
P1	0.5	5	2	3,2,4,5
P2	2	3	2	3,2,4,1
P3	2	4	1	2,4,2,4
P4	20	7	6	1,1,1,1
Scenario-10MS: Architectures with 10 microservices				
P5	13	6	4	5,3,2,1
P6	8	4	2	1,5,1,4
P7	1	5	3	3,3,2,4
P8	3	5	3	2,4,4,3