

Research Article

A New Plant Intelligent Behaviour Optimisation Algorithm for Solving Vehicle Routing Problem

Godfrey Chagwiza 

Department of Applied Mathematics, National University of Science & Technology, P.O. Box AC939, Ascot, Bulawayo, Zimbabwe

Correspondence should be addressed to Godfrey Chagwiza; chagwizag@gmail.com

Received 20 June 2017; Revised 5 December 2017; Accepted 11 January 2018; Published 6 February 2018

Academic Editor: Leonilde Varela

Copyright © 2018 Godfrey Chagwiza. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new plant intelligent behaviour optimisation algorithm is developed. The algorithm is motivated by intelligent behaviour of plants and is implemented to solve benchmark vehicle routing problems of all sizes, and results were compared to those in literature. The results show that the new algorithm outperforms most of algorithms it was compared to for very large and large vehicle routing problem instances. This is attributed to the ability of the plant to use previously stored memory to respond to new problems. Future research may focus on improving input parameters so as to achieve better results.

1. Introduction

A lot of research has been done and is ongoing on development of optimisation algorithms that are motivated by natural processes. In this research, an attempt to develop a new plant intelligence behaviour optimisation (PIBO) algorithm motivated by intelligent behaviour of plants to survive and avoid competition is presented. Natural processes such as how plants fight competition and change morphology in response to environmental changes make it worthwhile to develop an algorithm inspired by plant intelligence. Plant intelligence starts with cell molecular network. Molecular connections result in a living organism. It is believed that metabolic steps act like Boolean computer logic gates such as Or, AND, and NOR [1] and act like chemical neurones [2, 3].

Pattern recognition is achieved through assembling chemical neuron together [4]. Computational elements in plant intelligence are taken to be proteins [5]. Plants can forage for their food through changing their behaviour. This is achieved by changing physiology, phenotype, and architecture [6]. Plants make decisions as they grow, thereby placing roots, shoots, and leaves at optimal points or positions depending on availability or positions of perceived resources [7, 8].

Overgrown branches can trigger decision to seal the vascular connection system and redirect resources to the rest of the plant [9]. The plants have an ability to sense

volume [10]. Plants can avoid competition and can respond to local opportunities that will be of use in future [11]. Roots plays a pivotal role in plant intelligence. The paper is highly motivated by adaptability of plants to severe conditions, ability to avoid competition, and ability to survive under dynamic conditions. Using this intelligence of plants to develop an optimisation algorithm can result in a better optimisation algorithm capable of adapting to conditions given. An optimisation algorithm that is flexible and adapting to different conditions is necessary with current advancement in technology. The proposed algorithm is different from other nature inspired algorithms in that it can redirect time resources to solutions that are feasible only. The remainder of the paper is arranged as follows. In Section 2, formulation of the new PIBO algorithm is presented. Computational experiments, results, and discussions are presented in Section 3, and conclusions are drawn in Section 4.

2. PIBO Algorithm Development

In this section an attempt to develop the new optimisation algorithm is presented. Basic steps of plant intelligence are used as steps of the new algorithm. Parameters and Variables section presents definitions of parameters and variables used.

2.1. Environmental Information. Environmental information comes in various forms that require different actions.

```

(1) START
(2) Objective function  $O(x_i) = (x_1, x_2, \dots, x_i)^T$ 
(3) Generate initial population of  $N$  environmental information  $x_i$ 
(4) Define perception,  $F_i$  for  $x_i$ 
(5) while  $N < \text{MaxGen}$  do
(6)   Get a plant randomly
(7)   Initiate signal transduction creating response  $r_i$ 
(8)   Evaluate  $r_i$  appropriateness by finding  $P(r_i)$ 
(9)   Set up the initial adaptive network,  $G_0$ 
(10)  Get  $\alpha, \beta, \tau$  and  $\eta$ 
(11)  if  $(\alpha x_i > 0 \wedge \beta x_i > 0 \wedge \tau x_i > 0) \wedge (\eta x_i > 0 \vee \eta x_i \leq 0)$  then
(12)    Find  $\max\{G(x_i)\}$ 
(13)    Choose  $(\alpha, \tau, \beta) > 0$  that produces the set  $\max\{P(G)\}$ 
(14)  else if  $\alpha = \beta = \tau \leq 0$  then
(15)    Set  $P(G) = 0$ 
(16)    Wait until  $(\alpha > 0 \wedge \beta > 0 \wedge \tau > 0) \wedge (\eta > 0 \vee \eta \leq 0)$ 
(17)  end if
(18)  Construct the set  $V = \{V_{1,1}^k, \dots, V_{ji}^k\}$  of the response instances
(19)  Generate the response tissue integration decision  $M(x_i, t)$ 
(20)  if  $M(x_i, t+1) > M(x_i, t)$  then
(21)    Choose the response  $V_{(x_i, t+1)}$ 
(22)  else if  $M(x_i, t+1) < M(x_i, t)$  then
(23)    Choose response  $V_{(x_i, t)}$ 
(24)  else if  $V_{(x_i, t)} \notin V$  then
(25)    Set  $M(x_i, t) = 0$  and stop, abandon the response decision and make another decision
(26)  end if
(27)  Evaluate fitness,  $A_i$ 
(28)  if  $A_i > A_i^*$  then
(29)    Produce intelligent phenotype,  $C_i$ 
(30)  else if  $A_i < A_i^*$  then
(31)    Set  $C_i = C_0$  and re-analyse  $F_i$ 
(32)  end if
(33)  Determine compactness,  $Z(x_i)$ , of  $C_i$ 
(34)  if  $Z(x_i) \geq Z(x_i^*)$  then
(35)    Set  $Z(x_i) = Z^*(x_i)$ 
(36)  else if  $Z(x_i) \leq Z(x_i^*)$  then
(37)    Set  $Z(x_i^*) = Z^*(x_i)$ 
(38)  end if
(39)  Find best current solution,  $O(x_i^*)$ 
(40) end while
(41) Return the results
(42) STOP

```

ALGORITHM 1

A population, N , of environmental information, x_i , is generated by various plant parts such as roots, leaves, and trunk. A population $x_i \in N$ is generated first before anything else. Line (3) of Algorithm 1 defines environmental information.

2.2. Plant Perception. Plants perception, sometimes referred to as biocommunication, is a way how plants communicate. Plants have the ability to sense and respond to the environment. This can lead to adjustment in morphology, physiology, and phenotype. Reactions to chemicals, light, gravity, parasites, and many other factors have been noted in plants. Perception frequency, F_i , for environmental information, x_i , is defined. This will be used in future encounters with the same information. Equation (1) represents perception

frequency where γ is drawn from a uniform distribution, $U[0, 1]$. In Algorithm 1, line (4) presents plant perception:

$$F_i = h(x_{i,\min}) + \gamma [h(x_{i,\max}) - h(x_{i,\min})]. \quad (1)$$

2.3. Signal Transduction. Signal transduction occurs when the extracellular molecule activates a receptor that in turn triggers biochemical events resulting in a response. Signal transduction (line (7) of Algorithm 1), r_i , is affected by stimulus location, type, duration, and intensity. Equation (2) shows the signal transduction level of the environmental information by the plant:

$$r_i = \frac{1}{(s_{\text{type}} + s_{\text{loc}} + s_{\text{dur}} + s_{\text{int}})} x_i \times 100\%. \quad (2)$$

2.4. Assessment. The plant will assess the information transmitted in order to make an informed decision. This is an important step of the algorithm because it is the one that affects the action to be taken. Assessment of x_i is carried out to determine the effect of the environmental information, and (3) presents assessment function. It is shown that $0 \leq P(r_i) \leq 1$ from (3). The smaller the value of $P(r_i)$ is, the greater the chance that the adaptive representational network, G , is formulated is. Research suggests that memory in plants cannot be constructed without first learning. Plants learn complex things such as stress, condition of drought, cold, heat, flooding, and many more through reinforcement [31–34], and this is the reason for using previous environmental information when carrying out current assessment.

$$P(r_i) = \frac{x_i^*}{x_i}, \quad x_i^* \leq x_i, \quad x_i \neq 0. \quad (3)$$

2.5. Adaptive Representational Network. Adjustment in the morphology or physiology and phenotype may take place depending on the type of environmental information at hand. The network $G = (B, E)$ will be formulated where B is the set of components (morphology, phenotype, and physiology) and E are edges. This network presents what is to be done. Use of Boolean operators AND, OR, and NOR is carried out at this stage of the algorithm. To ensure that the adaptive representational network formulated is useful, set α , β , and τ to be nonnegative. The tolerance optimiser does not matter if it is negative.

2.6. Tissue Integration. The state of all nodes in the adaptive representational network can be modelled as follows:

$$\bar{Y}_i^{k+1} = Y_i(x_1^k, x_2^k, \dots, x_I^k), \quad i \in I. \quad (4)$$

The variable $\bar{Y}_i^{[.]}$ is updated according to

$$\bar{Y}_i^{k+1}(T_i^k) = Y_i(x_1^k[V_{1i}^k], x_2^k[V_{2i}^k], \dots, x_I^k[V_{Ii}^k]), \quad (5)$$

$$k \in \mathbb{N}, \quad j = 1, 2, \dots, J,$$

where (6) and (7) represent T_i^k and V_{ji}^k , respectively.

$$T_i^k = \min_{jl} \{T_j^l \geq t\}, \quad \text{for some } k \quad (6)$$

$$V_{ji}^k = \max_l \{T_j^l : T_i^k > T_j^l\} = V. \quad (7)$$

The response tissue integration decision is then presented by

$$M(x_i, t) = \max \{\bar{Y}_i^1, \dots, \bar{Y}_i^k\}. \quad (8)$$

2.7. Fitness and Intelligent Phenotype Production. In this step, the algorithm will evaluate the fitness of tissue integration decision. Information gathered from learning previous environmental information reinforces the evaluation criteria. This means that the previous environmental information

intensity, x_i^* , leading to the optimal tissue integration decision, A_i^* , should be smaller than or equal to the current environmental information intensity, x_i , so that the current fitness, A_i , can lead to the intelligent phenotype production, C_i .

2.8. Compactness and the Solution. Compactness of the intelligent phenotype produced is important for the plant to achieve the goal. The algorithm ranks the current solution and the best solution is printed. This solution must be better than the previous solution produced for the same environmental information. The PIBO algorithm is presented as in Algorithm 1.

Figure 1 represents the algorithm flow chart of the new PIBO algorithm.

Relevance estimation and value calibration (REVAC) introduced by Nannen and Eiben [35] is used to assist tuning the control parameters. Racing of Maron and Moore [36] and sharpening used by Bartz-Beielstein et al. [37] are incorporated into the REVAC algorithm to fine-tune the parameters. Default values of the REVAC algorithm are used. The best values of tuning the tolerance optimiser are $\eta = -0.1$ and $\eta = 0.13$, throughput accelerator, $\tau = 0.077$, and filter parameter, $\alpha = 0.89$. These values of the input parameters are used to execute the PIBO algorithm using three sets of benchmark problems. Figures 2 and 3 show the graphical presentation of the tolerance optimiser of the best two values.

3. Computational Experiments

This section presents computational experiments of the PIBO algorithm and its results are compared to those available in literature. The algorithm is implemented to solve three sets of known benchmark vehicle routing problems, 12 very large-scale vehicle routing problems (VLVRP) in Li et al. (2005) [28], 20 large-scale vehicle routing problems (LVRP) in Golden et al. (1998) [38], and 14 small- or medium-sized vehicle routing problems (SVRP) in Christofides and Eilon (1969) [39]. Computational experiments are performed in MATLAB 7.0.4 on a PC with AMD E-300 APU with Radeon™ @1.30 GHz and 4.00 GB RAM. A total of 30 runs are performed for each set of benchmark problems. Table 1 presents information of the problem instances that are used in this paper.

Computational results of the PIBO algorithm are shown in Table 2. Two best values of tolerance optimiser, η , are used. The third column (avail sol) presents the available solution, time (m) is the time in minutes, and number of feasible solutions is presented in the succeeding column. The average solutions are presented by the column labeled “avg sol” and best solutions by the column titled “best sol,” and D is the deviation. The results show that the PIBO algorithm performs better when $\eta = 0.13$ as compared to $\eta = -0.1$.

Table 3 show results of the PIBO algorithm compared to the best results available in the literature for the SVRP problems. The results show that PIBO algorithm is promising. The average deviation from the previous best solution obtained by executing the PIBO algorithm to achieve optimality is

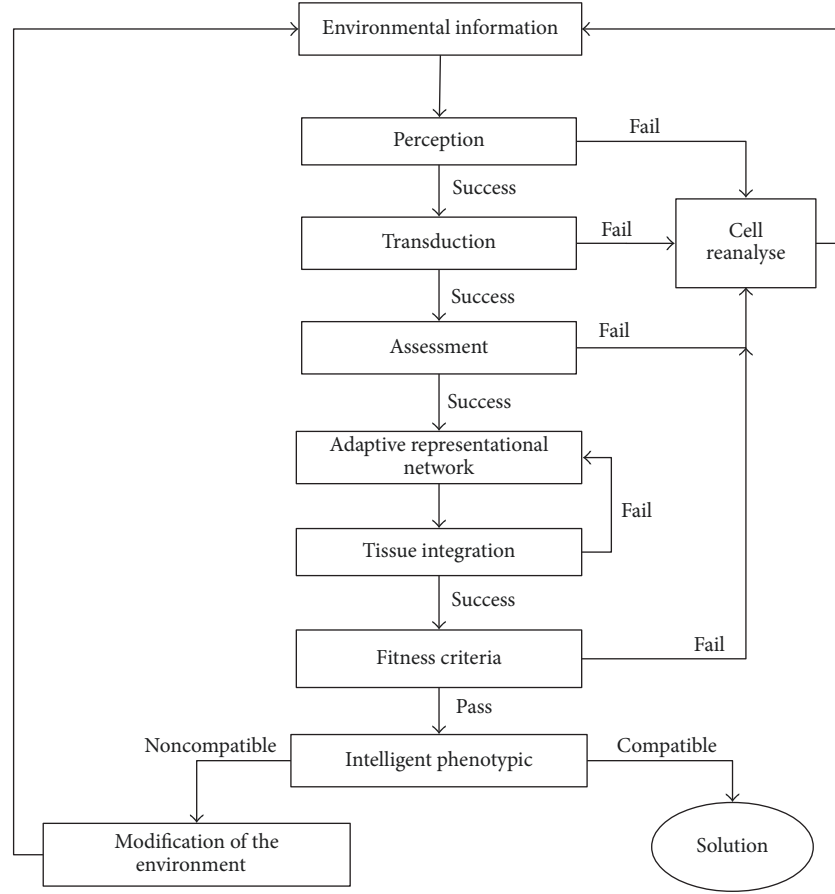
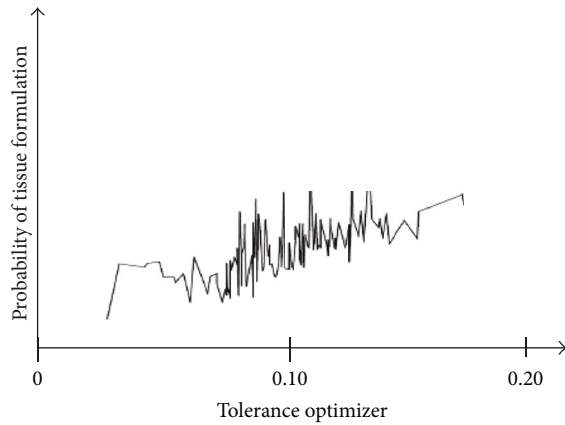
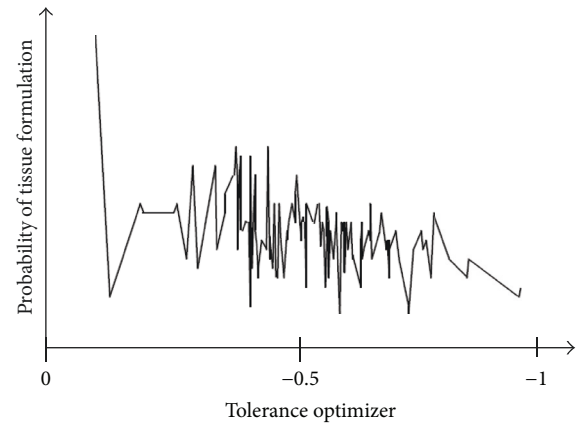


FIGURE 1: Algorithm flow chart.

FIGURE 2: Graphical representation of $\eta = 0.13$.FIGURE 3: Graphical representation of $\eta = -0.1$.

0.07, while the best result so far is 0.00 seconds obtained by a computer with processor speed of 2 G. The algorithm outperformed most of the results in literature and might even perform better if it is executed on a PC with a faster processor.

In Table 4, PIBO algorithm with $\eta = 0.13$ is compared to the results available in literature for the LVRP set of benchmark problems. The best solutions obtained by the

PIBO algorithm outperform all the other solutions they have been compared to for this set of benchmark problems. Finally, results obtained by executing PIBO to solve VLVRP are presented in Table 5. Again, the results show that PIBO algorithm outperforms all the other algorithms it has been compared with. The best average results deviation was obtained by Xiao et al. (2014) [26] as $D = 0.11$. There is a significant difference

TABLE 1: Information of the SVRP, LVRP and VLRP benchmark problems instance.

Type	Problem	Number of customers	Number of vehicles	Vehicles capacity	Maximum route length	Service time
SVRP	1	50	5	160		
	2	75	10	140		
	3	100	8	200		
	4	150	12	200		
	5	199	17	200		
	6	50	6	160	200	10
	7	75	11	140	160	10
	8	100	9	200	230	10
	9	150	14	200	200	10
	10	199	18	200	200	10
	11	120	7	200		
	12	100	10	200		
	13	120	11	200	720	50
	14	100	11	200	1040	90
LVRP	1	240		550	650	
	2	320		700	900	
	3	400		900	1200	
	4	480		1000	1600	
	5	200		900	1800	
	6	280		900	1500	
	7	360		900	1300	
	8	440		900	1200	
	9	255		1000	∞	
	10	323		1000	∞	
	11	399		1000	∞	
	12	483		1000	∞	
	13	252		1000	∞	
	14	320		1000	∞	
	15	396		1000	∞	
	16	480		1000	∞	
	17	240		200	∞	
	18	300		200	∞	
	19	360		200	∞	
	20	420		200	∞	
VLRP	1	560	10			
	2	600	15			
	3	640	10			
	4	720	10			
	5	760	20			
	6	800	11			
	7	840	20			
	8	880	10			
	9	960	10			
	10	1040	10			
	11	1120	10			
	12	1200	11			

TABLE 2: Computational results of the SVRP, LVRP and VLVRP benchmark problems instance.

Type	Problem	Avail best	Time (min)	$\eta = -0.1$			D% best	Time (min)	No. feasible sol	Best sol	$\eta = 0.13$			D%
				No. feasible sol	Avg sol	Best sol					No. feasible sol	Avg sol	Best sol	
SVRP	1	524.61	0.15*	30	524.61	524.6	0.00	0.15*	30	523.09	523.07	0.29		
	2	835.26	0.32*	30	834.09	834.09	0.14	0.32*	30	834.01	834	0.00		
	3	826.14	0.37*	29	825	824.9	0.15	0.35*	30	824.87	824.87	0.02		
	4	1028.42	0.80*	30	1028.05	1028.05	0.04	0.78*	30	1028.04	1028.02	0.04		
	5	1291.29	1.08*	27	1291.32	1291.31	0.00	1.07*	29	1292.01	1292.01	-0.06		
	11	1042.11	0.48*	30	1042.06	1042.02	0.01	0.48*	30	1041.97	1041.97	0.01		
	12	819.56	0.34*	30	819.6	819.56	0.00	0.33*	30	819.03	819.02	0.07		
	1	5627.54	1.47	30	5554.01	5553.87	1.31	1.43	30	5551.98	5548.02	0.05		
	2	8447.92	2.31	30	8447.92	8444.11	0.05	2.3	30	8440.99	8440.99	0.08		
	3	11036.22	3.2	30	11025.10	11025.1	0.10	3.2	30	11022.01	11019.76	0.15		
	4	13624.52	4.3	30	13602.34	13601.1	0.17	3.9	30	13548.9	13544.06	0.03		
	5	6460.98	1.1	30	6459.07	6459.03	0.03	1	30	6458.63	6455.52	0.08		
LVRP	6	8412.8	1.42	30	8409.30	8408.54	0.05	1.02	30	8402.33	8400.31	0.15		
	7	10181.75	1.76	30	10177.98	10177.02	0.05	1.44	30	10167.71	10152.98	0.12		
	8	11643.9	2.55	30	11638.02	11638.02	0.05	2.43	30	11635.7	11634.44	0.08		
	9	580.48	4.02	30	580.02	580.02	0.08	3.91	30	567.2	565.75	0.00		
	10	738.73	1.5	30	737.98	737.98	0.10	1.02	30	733.33	732.08	0.21		
	11	914.75	3.72	30	912.22	912.04	0.30	3.22	30	909.14	909.14	0.31		
	12	1106.33	4.2	30	1106.01	1106.01	0.03	3.78	30	1088.4	1088.4	0.02		
	13	857.19	1.7	30	857.22	857.22	0.00	1.4	30	853.49	853.02	0.01		
	14	1080.55	2.4	30	1080.03	1080.03	0.05	2.3	30	1077.54	1077.54	0.21		
	15	1340.24	3	30	1338.11	1338.1	0.16	3	30	1333.61	1333.52	0.21		
	16	1616.33	4.98	28	1614.20	1614.19	0.13	4.98	28	1614.2	1614.2	0.09		
	VLVRP	17	707.76	1.1	30	702.82	702.8	0.70	0.9	30	702.66	701.76	0.00	
18		995.39	1.49	30	991.04	991.01	0.44	1.49	30	991.01	991.01	0.34		
19		1366.14	2.2	30	1364.90	1364.9	0.09	2.22	30	1355.82	1355.76	0.02		
20		1819.99	2.48	30	1815.81	1815.78	0.23	2.45	30	1811.11	1811.02	0.09		
1		16212.74	5.90	30	16219.81	16219.81	-0.04	5.77	30	16211.81	16211.81	0.01		
2		14597.18	6.70	30	14567.09	14555.74	0.28	6.67	30	14553.78	14553.78	0.30		
3		18801.12	7.65	30	18889.05	18855.99	-0.29	7.56	30	18888.41	18888.41	-0.46		
4		21389.33	9.45	30	21373.91	21372.77	0.08	9.45	30	21370.99	21370.99	0.09		
5		16902.16	11.32	30	16880.01	16784.76	0.69	11.02	30	16878.05	16878.04	0.14		
6		23971.7	12.03	30	23965.70	23965.7	0.03	12.00	30	23964.17	23964.15	0.03		
7		17488.74	12.64	30	17473.11	17472.21	0.09	12.76	30	17467.63	17467.63	0.12		
8		26565.92	13.23	30	26566.38	26566.38	0.00	13.20	30	26565.04	26565.04	0.00		
9	29154.34	15.57	30	29144.93	29144.93	0.03	15.55	30	29141.09	29141.08	0.05			
10	31742.51	18.09	29	31746.72	31744.92	-0.01	18.66	30	31746.05	31746.03	-0.01			
11	34330.84	20.10	30	34335.01	34335	-0.01	20.11	30	34332.76	34332.76	-0.01			
12	36919.24	26.77	28	36912.51	36911.34	0.02	26.22	28	36912.03	36912.03	0.02			

*Time recorded in seconds in order to show differences.

TABLE 3: Comparison of the results of different algorithms for the SVRP benchmark problem instances.

Problem	SA	SB	SC	SD	SE	SF	SG	SH	SI	SJ	SK	SL	SM	SN	SO	SP	PIBO
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.29
2	0.00	0.00	0.40	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.29	0.00	0.15	0.00	0.00	0.16	0.00	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.02
4	0.41	0.24	0.47	0.31	0.75	0.00	0.00	0.76	0.00	0.11	0.12	0.26	0.00	0.10	1.00	0.29	0.04
5	1.90	1.77	2.09	0.69	2.54	1.56	0.00	1.24	0.00	0.45	1.01	2.64	0.00	0.22	1.58	1.07	0.03
11	3.01	0.00	0.07	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.01
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07
Avg	0.76	0.29	0.47	0.14	0.51	0.22	0.00	0.31	0.00	0.08	0.17	0.44	0.00	0.05	0.42	0.19	0.07
CPU	2 G	400 M	200 M	1 G	400 M	400 M	2 G	733 M	2.8 G	3 G	2.4 G	2.4 G	2.4 G	2.8 G	2.8 G	1.6 G	1.3 G
Time (min)	24.62	5.22	3.84	5.19	21.25	6.96	0.08	28.91	0.03	17.43	5.84	4.54	23.74	5.19	8.21	-	4.51

Notes. SA, Cordeau et al. (2001) [12]; SB, Tarantilis and Kiranoudis (2002) [13]; SC, Toth and Vigo (2003) [14]; SD, Prins (2004) [15]; SE, Berger and Barkaoui (2004) [16]; SF, Tarantilis (2005) [17]; SG, Mester and Braysy (2005) [18]; SH, Ergun et al. (2006) [19]; SI, Mester and Braysy (2007) [20]; SJ, Pisinger and Ropke (2007) [21]; SK, Derigs and Kaiser (2006) [22]; SL, Derigs and Kaiser (2006) [22]; SM, Nagata (2007) [23]; SN, Prins (2009) [24]; SO, Lin et al. (2009) [25]; SP, Xiao et al. (2014) [26].

TABLE 4: Comparison of the results of the LVRP benchmark problems instance.

Problem	LA	LB	LC	LD	LE	LF	LG	LH	LI	LJ	LK	LL	LM	PIBO
1	1.93	0.29	0.69	0.00	4.27	4.27	0.42	0.24	0.30	0.30	2.99	0.57	-0.01	0.05
2	1.24	0.01	0.25	0.00	0.34	0.34	0.25	0.18	0.03	0.00	0.64	0.20	0.06	0.08
3	3.32	0.00	0.99	0.00	0.07	0.07	0.10	0.04	0.51	0.00	2.98	0.21	0.00	0.15
4	9.44	0.55	0.98	0.00	0.10	0.05	0.08	0.83	0.80	0.00	3.76	0.00	0.02	0.03
5	3.66	0.00	0.26	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.79	0.00	0.00	0.08
6	6.54	0.00	1.51	0.00	0.03	0.03	0.04	0.01	0.00	0.00	1.67	0.00	0.00	0.15
7	3.59	0.14	1.06	0.14	1.14	1.14	0.00	0.84	0.14	0.14	2.37	0.84	0.14	0.12
8	3.37	1.59	2.38	0.17	1.99	1.96	0.60	1.95	1.98	0.00	2.94	0.97	0.17	0.08
9	2.22	1.10	1.34	0.50	6.92	6.92	0.80	0.80	1.44	0.99	1.07	0.80	0.00	0.00
10	1.75	1.63	1.46	0.38	8.42	6.23	1.38	0.97	1.20	0.76	1.38	0.90	0.04	0.21
11	2.33	1.37	1.22	0.40	9.44	7.88	0.87	0.89	1.65	0.84	1.09	0.65	0.17	0.31
12	3.69	3.12	1.96	0.08	10.34	9.28	1.15	1.12	1.08	0.88	1.75	0.97	0.25	0.02
13	1.35	0.92	0.93	0.22	10.28	8.01	0.87	0.97	0.92	0.60	1.18	0.52	0.24	0.01
14	1.45	1.22	1.59	0.07	9.52	6.91	1.37	1.43	1.41	0.81	1.69	0.77	0.00	0.21
15	2.18	1.34	1.58	0.37	10.17	9.05	1.47	1.11	1.22	0.91	1.22	1.20	0.36	0.21
16	2.23	1.16	1.19	0.39	8.62	7.83	1.41	1.56	1.89	1.11	1.64	0.89	0.35	0.09
17	0.47	0.14	0.56	0.00	5.09	2.58	0.16	0.25	0.27	0.15	0.64	0.16	0.17	0.00
18	2.15	0.35	1.50	0.34	11.05	8.25	0.71	0.73	0.95	0.68	2.26	1.52	0.52	0.34
19	2.55	0.08	1.20	0.05	6.53	5.74	0.59	1.26	1.12	0.40	1.38	1.21	0.24	0.02
20	5.27	0.16	1.70	0.01	9.32	6.49	0.59	1.65	1.46	0.60	1.97	1.09	0.21	0.09
Avg	3.04	0.76	1.22	0.16	5.68	4.65	0.65	0.84	0.92	0.46	1.77	0.67	0.15	0.11
CPU	200 M	900 M	1 G	2.8 G	3 G	3 G	3 G	2.4 G	2.4 G	2.8 G	2.8 G	2.93 G	1.6 G	1.3 G
Avg time (min)	17.55	49.3	1.33	24.35	0.18	1.01	107.61	109.63	111.10	66.9	118.98	284.4	-	2.65

Notes. LA, Toth and Vigo (2003) [14]; LB, Reimann et al. (2004) [27]; LC, Li et al. (2005) [28]; LD, Mester and Braysy (2007) [20]; LE, Kytöjoki et al. (2007) [29]; LF, Kytöjoki et al. (2007) [29]; LG, Pisinger and Ropke (2007) [21]; LH, Derigs and Kaiser (2006) [22]; LI, Derigs and Kaiser (2006) [22]; LJ, Prins (2009) [24]; LK, Lin et al. (2009) [25]; LL, Chen et al. (2010) [30]; LM, Xiao et al. (2014) [26].

between the result obtained by PIBO and the average best result in literature.

Statistical analysis of the results is done to establish significance differences between them, if any. Analysis of variance (ANOVA) is performed on the results obtained by the 30 runs performed. Results show that the results obtained by the PIBO algorithm are significantly different from those

that are available in literature with the probability value $p = 0.02$ and $p = 0.00$ for the LVRP and VLVRP, respectively, but insignificantly different ($p = 0.65$) for the SVRP instances. Since the solution space is very large for these problems, memory of previous best solution is important. The ability of the plant to change morphology, physiology, and phenotype in response to the environment and conditions and storage

TABLE 5: Comparison of the results of the VLVRP benchmark problems instance.

Problem	VA	VB	VC	VD	VE	VF	PIBO
1	1.10	2.41	0.00	0.19	0.05	0.01	0.01
2	0.37	0.23	0.00	0.40	0.40	0.14	0.30
3	1.09	0.19	0.00	0.11	0.05	0.00	-0.46
4	1.85	0.62	0.00	0.06	0.06	0.01	0.09
5	1.47	0.00	1.14	3.72	2.7	0.65	0.14
6	0.91	0.18	0.00	0.36	0.1	0.04	0.03
7	1.91	0.71	0.00	4.34	4.26	-0.32	0.12
8	0.15	0.85	0.00	0.1	0.1	0.00	0.00
9	0.09	0.86	0.02	0.04	0.04	0.01	0.05
10	0.74	0.71	0.00	0.2	0.2	0.00	-0.01
11	3.02	1.28	0.00	0.06	0.06	0.00	-0.01
12	1.36	1.24	0.03	0.29	0.29	0.77	0.02
Average	1.28	0.58	0.10	0.82	0.69	0.11	0.02
CPU	1 G	3 G	2.8 G	3 G	3 G	1.6 G	1.3 G
Avg time (min)	2.72	7.7	8	0.04	0.13	-	5.03

Notes. VA, Li et al. (2005) [28]; VB, Pisinger and Ropke (2007) [21]; VC, Mester and Braysy (2007) [20]; VD, Kytöjoki et al. (2007) [29]; VE, Kytöjoki et al. (2007) [29]; VF, Xiao et al. (2014) [26].

of memory of previous experiences may have assisted the PIBO algorithm to be more efficient when solving these problems.

4. Conclusion

A new optimisation algorithm called PIBO is formulated and executed to solve some benchmark vehicle routing problems. The development of the algorithm is motivated by plant intelligence behaviour. Results show that the PIBO algorithm outperforms most of the results reported in literature for LVRP and VLVRP problem instances. The PIBO algorithm is a promising optimisation algorithm to solve large vehicle routing problems which have a very large solution space. The new algorithm has an ability to adjust to changing problem conditions. It is suggested that future research be directed towards improving input parameters and testing applicability of the algorithm to a wider range of problems.

Parameters and Variables

- α : Filter parameter
- β : Temporary pattern generator
- τ : Throughput accelerator
- η : Tolerance optimiser
- γ : A random vector affecting decision making and $\gamma \sim U[0, 1]$
- A_i : Fitness of the tissue integration decision
- A_i^* : Previous fitness of tissue integration in response to the environmental information
- B : Set of components
- C_i : Intelligent phenotype
- E : Edges
- F_i : Perception

- G : Adaptive representational network
- h : Action taken in response to environmental information
- $h(x_{i,\min})$: Minimum effect of action h responding to environmental information x_i
- $h(x_{i,\max})$: Maximum effect of action h responding to environmental information x_i
- i : Index identifying an instant, $i \in I$
- k, l : Indexes identifying the protein
- $M(x_i, t)$: Tissue integration decision
- $O(x_i)$: Objective function
- $P(r_i)$: Assessment function of the signal transduction
- r_i : Signal transduction response level
- s_{dur} : Stimulus duration expressed as a percentage
- s_{int} : Stimulus intensity expressed as a percentage
- s_{loc} : Stimulus location expressed as a percentage
- s_{type} : Stimulus type expressed as a percentage
- T_i^k : Updating time at node q
- T_j^d : Updating time for the next node j
- V : Set of response activities
- V_{ji}^k : Most recent instant when node j is updated
- x_i : Environmental information
- x_i^* : Previous environmental information that triggered action
- $Y(x_i^k)$: Regulating function with respect to environmental information x_i
- $\bar{Y}_i^{[\cdot]}$: Protein
- $Z(x_i)$: Compactness of the solution.

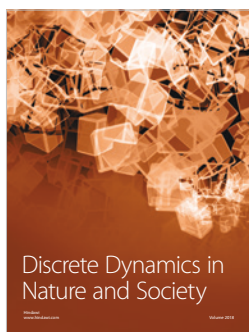
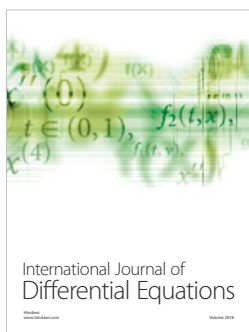
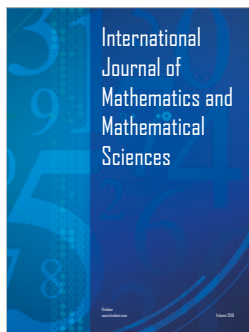
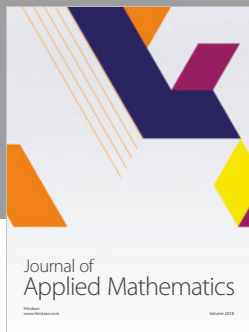
Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

- [1] A. Arkin and J. Ross, "Computational functions in biochemical reaction networks," *Biophysical Journal*, vol. 67, no. 2, pp. 560–578, 1994.
- [2] A. Hjelmfelt and J. Ross, "Chemical implementation and thermodynamics of collective neural networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, no. 1, pp. 388–391, 1992.
- [3] M. Okamoto, T. Sakai, and K. Hayashi, "Switching mechanism of a cyclic enzyme system: role as a "chemical diode"," *BioSystems*, vol. 21, no. 1, pp. 1–11, 1987.
- [4] A. Hjelmfelt, F. W. Schneider, and J. Ross, "Pattern recognition in coupled chemical kinetic systems," *Science*, vol. 260, no. 5106, pp. 335–337, 1993.
- [5] D. Bray, "Protein molecules as computational elements in living cells," *Nature*, vol. 376, no. 6538, pp. 307–312, 1995.
- [6] J. P. Grime and J. M. L. Mackey, "The role of plasticity in resource capture by plants," *Evolutionary Ecology*, vol. 16, no. 3, pp. 299–307, 2002.
- [7] T. Yamada, T. Okuda, M. Abdullah, M. Awang, and A. Furukawa, "The leaf development process and its significance for reducing self-shading of a tropical pioneer tree species," *Oecologia*, vol. 125, no. 4, pp. 476–482, 2000.
- [8] H. Honda and J. B. Fisher, "Tree branch angle: Maximizing effective leaf area," *Science*, vol. 199, no. 4331, pp. 888–890, 1978.
- [9] J. Henriksson, "Differential shading of branches or whole trees: Survival, growth, and reproduction," *Oecologia*, vol. 126, no. 4, pp. 482–486, 2001.
- [10] H. J. Schenk, R. M. Callaway, and B. E. Mahall, "Spatial Root Segregation: Are Plants Territorial?" *Advances in Ecological Research*, vol. 28, no. C, pp. 145–180, 1999.
- [11] O. Falik, P. Reides, M. Gersani, and A. Novoplansky, "Self/non-self discrimination in roots," *Journal of Ecology*, vol. 91, no. 4, pp. 525–531, 2003.
- [12] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928–936, 2001.
- [13] C. D. Tarantilis and C. T. Kiranoudis, "BoneRoute: an adaptive memory-based method for effective fleet management," *Annals of Operations Research*, vol. 115, pp. 227–241, 2002.
- [14] P. Toth and D. Vigo, "The granular tabu search and its application to the vehicle-routing problem," *INFORMS Journal on Computing*, vol. 15, no. 4, pp. 333–346, 2003.
- [15] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.
- [16] J. Berger and M. Barkaoui, "A new hybrid genetic algorithm for the capacitated vehicle routing problem," *Journal of the Operational Research Society*, vol. 54, no. 12, pp. 1254–1262, 2004.
- [17] C. D. Tarantilis, "Solving the vehicle routing problem with adaptive memory programming methodology," *Computers & Operations Research*, vol. 32, no. 9, pp. 2309–2327, 2005.
- [18] D. Mester and O. Braysy, "Active guided evolution strategies for large-scale vehicle routing problems with time windows," *Computers & Operations Research*, vol. 32, no. 6, pp. 1593–1614, 2005.
- [19] Ö. Ergun, J. B. Orlin, and A. Steele-Feldman, "Creating very large scale neighborhoods out of smaller ones by compounding moves," *Journal of Heuristics*, vol. 12, no. 1–2, pp. 115–140, 2006.
- [20] D. Mester and O. Braysy, "Active-guided evolution strategies for large-scale capacitated vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 10, pp. 2964–2975, 2007.
- [21] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, 2007.
- [22] U. Derigs and R. Kaiser, "Applying the attribute based hill climber heuristic to the vehicle routing problem," *European Journal of Operational Research*, vol. 177, no. 2, pp. 719–732, 2006.
- [23] Y. Nagata, "Edge assembly crossover for the capacitated vehicle routing problem," in *Evolutionary computation in combinatorial optimization*, vol. 4446 of *Lecture Notes in Comput. Sci.*, pp. 142–153, Springer, Berlin, 2007.
- [24] C. Prins, "A grasp x evolutionary local search hybrid for the vehicle routing problem," in *Bio-inspired Algorithms for the Vehicle Routing Problem*, pp. 35–53, Springer, New York, NY, USA, 2009.
- [25] S.-W. Lin, Z.-J. Lee, K.-C. Ying, and C.-Y. Lee, "Applying hybrid meta-heuristics for capacitated vehicle routing problem," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1505–1512, 2009.
- [26] Y. Xiao, Q. Zhao, I. Kaku, and N. Mladenovic, "Variable neighbourhood simulated annealing algorithm for capacitated vehicle routing problems," *Engineering Optimization*, vol. 46, no. 4, pp. 562–579, 2014.
- [27] M. Reimann, K. Doerner, and R. F. Hartl, "D-ants: Savings based ants divide and conquer the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 4, pp. 563–591, 2004.
- [28] F. Li, B. Golden, and E. Wasil, "Very large-scale vehicle routing: new test problems, algorithms, and results," *Computers & Operations Research*, vol. 32, no. 5, pp. 1165–1179, 2005.
- [29] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau, "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 9, pp. 2743–2757, 2007.
- [30] P. Chen, H.-K. Huang, and X.-Y. Dong, "Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1620–1627, 2010.
- [31] H. Bohnert, "Abiotic stress," in *Encyclopedia of Life Sciences*, pp. 1–9, John Wiley & Sons, Ltd, Chichester, UK, 2007.
- [32] J. R. Dinneny, T. A. Long, J. Y. Wang et al., "Cell identity mediates the response of Arabidopsis roots to abiotic stress," *Science*, vol. 320, no. 5878, pp. 942–945, 2008.
- [33] C. J. Frost, M. C. Mescher, J. E. Carlson, and C. M. De Moraes, "Plant defense priming against herbivores: Getting ready for a different battle," *Plant Physiology*, vol. 146, no. 3, pp. 818–824, 2008.
- [34] L. A. C. J. Voesenek and R. Pierik, "Plant science: Plant stress profiles," *Science*, vol. 320, no. 5878, pp. 880–881, 2008.
- [35] V. Nannen and A. E. Eiben, "A method for parameter calibration and relevance estimation in evolutionary algorithms," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference 2006*, pp. 183–190, USA, July 2006.
- [36] O. Maron and A. W. Moore, "The Racing Algorithm: Model Selection for Lazy Learners," *Artificial Intelligence Review*, vol. 11, no. 1–5, pp. 193–225, 1997.
- [37] T. Bartz-Beielstein, K. E. Parsopoulos, and M. Vrahatis, "Design and analysis of optimization algorithms using computational statistics," *ANACM. Applied Numerical Analysis and Computational Mathematics*, vol. 1, no. 3, pp. 413–433, 2004.

- [38] B. L. Golden, E. A. Wasil, J. P. Kelly, and I. Chao, "The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results," in *Fleet Management and Logistics*, pp. 33–56, Kluwer, Boston, 1998.
- [39] N. Christofides and S. Eilon, "An Algorithm for the Vehicle-Dispatching Problem," *Operational Research Quarterly*, vol. 20, no. 3, pp. 309–318, 1969.



Submit your manuscripts at
www.hindawi.com