

Review Article

Bus Dispatching Interval Optimization Based on Adaptive Bacteria Foraging Algorithm

Zhong-hua Wei, Xia Zhao, Ke-wen Wang, and Yan Xiong

Beijing Key Laboratory of Traffic Engineering, Beijing University of Technology, Beijing 100124, China

Correspondence should be addressed to Zhong-hua Wei, weizhonghua@bjut.edu.cn

Received 4 September 2012; Accepted 14 November 2012

Academic Editor: Rui Mu

Copyright © 2012 Zhong-hua Wei et al. 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.

The improved bacterial foraging algorithm was applied in this paper to schedule the bus departing interval. Optimal interval can decrease the total operation cost and passengers' mean waiting time. The principles of colony sensing, chemotactic action, and improved foraging strategy made this algorithm adaptive. Based on adaptive bacteria foraging algorithm (ABFA), a model on one bus line in Hohhot city in China was established and simulated. Two other algorithms, original bacteria foraging algorithm (BFA) and genetic algorithm (GA), were also used in this model to decide which one could greatly accelerate convergence speed, improve searching precision, and strengthen robustness. The final result showed that ABFA was most feasible in optimizing variables.

1. Introduction

Traffic demand becomes increasingly higher with the great development of social economy and urbanization. With the large amount of private cars and limited road facilities, severe traffic congestion occurs inevitably. Municipal governments and road transport authority have strongly recommended public transportation for its higher passenger capacity and smaller coverage area. So it is necessary to improve the quality of bus dispatching management, and a wholesome and intelligent bus scheduling scheme is needed.

Bus interval scheduling is a complex optimization problem for its nonlinear and multiobjective characteristics. It requires traffic planners to take round consideration of multi-interests, such as bus operating enterprises and passengers. The relationship between the two is contradictory. Obviously, bus operating enterprises always try to dispatch buses as few as possible with the longest intervals and the highest profit, whereas passengers are the opposite. According to the two contradictory characteristics, an optimal bus dispatching interval model is urgent to be established to benefit both sides.

As to the optimal algorithms, lots of intelligent algorithms have been adopted to solve transportation optimal problems [1]. Intelligent algorithms enlightened by bacterium have become fashionable recently. Back to the history of bacteria foraging algorithm (BFA), it was initially proposed in 2002 by Passino [2] and applied to dealing with several engineering problems [3–5] successfully. However, it was limited to solo modal function optimization for its poor convergence behavior. So effort to pursue more adaptive algorithms goes on. Muñoz et al. [6] proposed some methods to simplify the algorithm while maintaining its core elements. These included the simplification of the algorithm architecture, the elimination of the bacteria colony scale, a clear adaptation rule for the step size, the use of a uniform distribution the position initialization, and the removal of the cell-to-cell communication. In 2008, Dasgupta et al. [7] mathematically analyzed the chemotactic step of a one-dimensional BFA and proposed the adaptive step size for BFA. In the same year, Chen et al. [8] analyzed how the run length unit parameter controlled the exploration and exploitation process of BFA and then employed the adaptive search strategy to significantly improve the performance of the original algorithm.

In this paper, the adaptive bacteria foraging algorithm was adopted to solve the complex bus interval dispatching problem. Many variables were taken into consideration such as dispatching interval, waiting time, load factor, economic efficiency, and trip service level. Among them, dispatching interval was chosen as the control variable. To compare which one was the best, genetic algorithm [9] (GA) and nonadaptive bacteria foraging algorithm (BFA) were employed in optimizing this model. The simulation results, focusing on minimizing two specific objective functions, depicted the advantages of the mentioned algorithm.

The rest of the paper is organized as follows. Section 2 gives brief views of basic and adaptive bacteria foraging algorithm. The theoretical optimal model for bus dispatching interval is built in Section 3. Based on this model, the simulations on ABFA, GA, and BFA are given in Section 4. Finally, Section 5 gives the conclusions.

2. Adaptive Bacteria Foraging Algorithm

2.1. Basic Bacteria Foraging Algorithm

To forage food, a bacterium needs to communicate with the group with its sensing abilities. By generating common knowledge, developing group identity and recognizing the identity of other colonies, bacterium engages in group decision-making [10]. A sort of collective intelligence subsequently occurs. And this wit consists of four steps: chemotaxis, swarming, reproduction, and elimination and dispersal [11–13].

(1) Chemotaxis

Basically, chemotaxis is a foraging strategy that implements a type of local optimization where the bacteria try to climb up the nutrient concentration [7]. Swimming and tumbling are two aspects of this step. The bacterium's flagella rotation determines its foraging direction in an anticipated route (swimming) or an unexpected one (tumbling). $\varphi(j)$ stands for the unit random length in the direction of a tumble, that is,

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\varphi(j), \quad (2.1)$$

where $\theta^i(j, k, l)$ means the i th bacteria's j th chemotactic behavior at the k th reproductive and l th elimination step. $C(i)$ represents the step size adopted in the random route in a tumble. And C is labeled as run length unit.

(2) *Swarming*

It is assumed that bacterium tends to inform others of its optimal food path, so group can locate food area more swiftly. Swarming gathers bacterium into high-density group and then forage food in a type of concentric circle, that is,

$$J_{cc} = \sum_{i=1}^S J_{cc}^i(\theta, \theta^i(j, k, l)) = \sum_{i=1}^S \left[-d_{\text{attract}} \exp \left(-w_{\text{attract}} \sum_{m=1}^P (\theta_m - \theta_m^i)^2 \right) \right] + \sum_{i=1}^S \left[h_{\text{repellent}} \exp \left(-w_{\text{repellent}} \sum_{m=1}^P (\theta_m - \theta_m^i)^2 \right) \right], \quad (2.2)$$

where $J_{cc}^i(\theta, \theta^i(j, k, l))$ is the cost function value to be minimized; S is the bacterium scale; P is the number of parameters to be optimized. d_{attract} , w_{attract} , $h_{\text{repellent}}$, and $w_{\text{repellent}}$ are different coefficients that are to be chosen carefully.

(3) *Reproduction*

The least healthy bacterium vanishes, and the other healthiest bacterium splits into two and are released in the same position. This makes the population of bacteria constant.

(4) *Elimination and Dispersal*

Possibly in the current surroundings, bacterium's life span changes slightly either by nutrient supplements or other unexpected influence. Accidents can kill some bacteria colonies in one second. But this has the effect of assisting in chemotaxis.

2.2. Adaptive Bacterial Foraging Algorithm

BFA is adaptive by adjusting the run length step parameter dynamically during its execution process to balance the exploration or exploitation search [14]. Each bacterium has two different foraging states in ABFA model.

- (1) Exploration: imprecisely explore the regions never gone before quickly in the search space in a large run length unit.
- (2) Exploitation: exploit the potential regions slowly in its immediate vicinity in a small run length unit.

The self-adaptive search is given in pseudocode in Table 1 below.

2.3. Algorithm Flowchart of ABFA

The flowchart of ABFA is listed below.

Table 1: Pseudocode for dynamic self-adaptive strategy [17, 18].

```

(1) FOR (each bacterium  $i$ ) IN PARALLEL
(2) IF (Criterion-1) then
(3)  $C^i(t+1) = C^i(t)/\alpha;$  // exploitation
(4)  $\varepsilon^i(t+1) = \varepsilon^i/\beta;$ 
(5) ELSE IF (Criterion-2) then
(6)  $C^i(t+1) = C_{\text{initial}};$  // exploration
(7)  $\varepsilon^i(t+1) = \varepsilon_{\text{initial}};$ 
(8) ELSE
(9)  $C^i(t+1) = C^i(t);$ 
(10)  $\varepsilon^i(t+1) = \varepsilon^i;$ 
(11) END IF
(12) END FOR IN PARALLEL

```

Where Criterion-2: exploration state; Criterion-1: exploitation state; t : the current iterations; α , β : default constants; $C^i(t)$: current run-length unit of the i th bacteria; $\varepsilon^i(t)$: the desired precision in the current iteration of the i th bacteria; C_{initial} : the original run length unit; $\varepsilon_{\text{initial}}$: the original precision.

Step 1. Initialize parameters like bacterium scale S , custom constants n , α , and β , run length step C_{initial} and precision $\varepsilon_{\text{initial}}$, and position of bacterium colony.

Step 2 (chemotaxis and swarming). In this process, the signaling concentration (fitness value) released on every path will be calculated. Bacterium will choose paths that own the highest fitness value.

Step 3 (reproduction). Sort by the bacteria colony's fitness values. The best half of the population undergoes reproduction. And the rest are eliminated to accelerate convergence speed.

Step 4 (elimination and dispersal). The wholesome adaptive bacteria colony migrate to other spaces randomly to expand the diversity of the colony.

Step 5 (judgment). If the current iteration is lower than the max iterations, then go to Step 2; if not, the algorithm loop ends.

The algorithm flowchart is given below in Figure 1, where, S —the colony scale; t —the current iterations; N_s —the max iterations in the current nutrient gradient; X_i —the current location in searching space of the i th bacteria; flag^i —the number of times that bacteria's fitness value has no continuous improvement.

3. Theoretical Optimal Model Building

Bus scheduling is a complex optimal problem influenced by varied external environment. According to the specific and available data on bus dispatching, the following hypothesis was made to establish a simulated model [15].

- (1) The type of buses was identical. They were well operated and dispatched strictly according to bus schedule, with no accidents on the road.

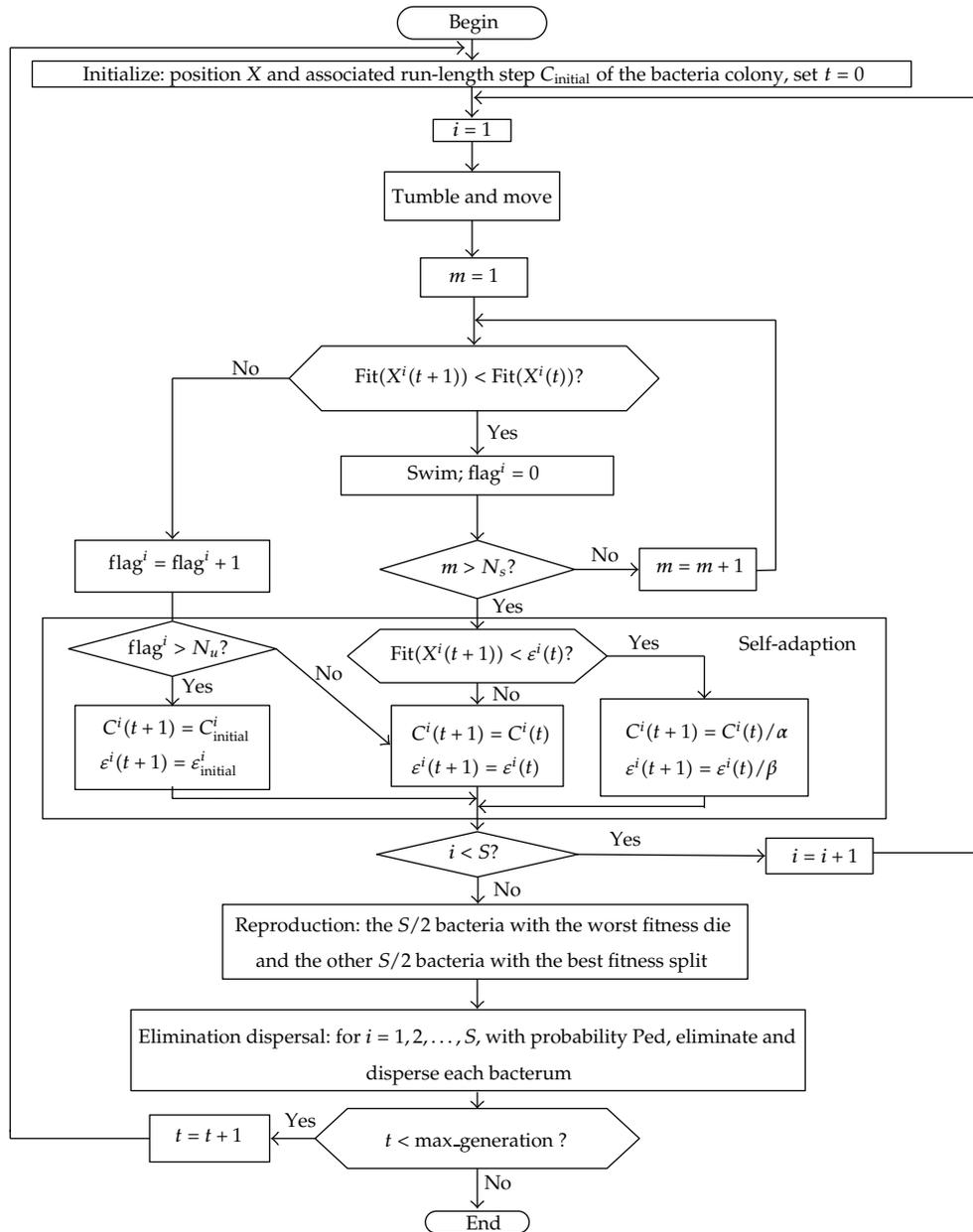


Figure 1: Flowchart of ABFA [15].

- (2) Passengers' traffic flow volume at a bus line was independent of each other; their arrival distribution followed a uniform distribution.
- (3) The dispatching interval of every two successive buses was identical in a given period.
- (4) The trip cost unit for passengers and operation cost unit for bus operating enterprises were fixed.

A certain bus line with its upstream travel direction was selected as the scheduling object. The whole operating time was divided into several one-hour intervals. And the theoretical optimization model was given below.

(1) Control Variable

Bus dispatching interval was chosen as control variable.

(2) Objective Function

Minimum bus operating cost and minimum passengers' waiting cost were two optimal objectives. In formula (3.1), f_1 represented the operation cost bus enterprises invested one day; in (3.2), f_2 represented the trip cost passengers invested one day; in (3.3), f represented the final minimum function with weighting coefficients given:

$$f_1 = \frac{\sum_{k=1}^K (t_k / \Delta t_k)}{t_s / ((t_{\max} + t_{\min}) / 2)}, \quad (3.1)$$

$$f_2 = \frac{\sum_{k=1}^K \sum_{j=1}^J m_j \times \rho_{kj} \Delta t_k^2 / \sum_{k=1}^K \sum_{j=1}^J U_{kj}}{(t_{\max} + t_{\min}) / 2}, \quad (3.2)$$

$$f = \min(\eta \times f_1 + \tau \times f_2), \quad (3.3)$$

where t_k is time duration at k th interval; $k = \max/\min$ means the upper or lower bound of bus interval; $k = s$ means the whole operating time span one day; Δt_k is the k th dispatching interval; m_j is passenger volume at the j th site; ρ_{kj} , u_{kj} is the density of passenger flow or passenger volume arriving at the j th site in the k th interval; U_{kj} is the getting off at the j th site in the k th interval; η , τ is weighting coefficient.

(3) Standard Constraint Condition

Formula (3.4) represented the mean bus load factor to be more than 75%; formula (3.5) represented dispatching interval to be between the lower and upper bounds; (3.6) represented condition to ensure profitability for bus enterprises.

Standard constraint condition

$$\frac{\sum_{k=1}^K \sum_{j=1}^J U_{kj}}{Q \times \sum_{k=1}^K (t_k / \Delta t_k)} \geq 75\%, \quad (3.4)$$

$$t_{\min} \leq \Delta t_k \leq t_{\max}, \quad (3.5)$$

$$\frac{\sum_{k=1}^K \sum_{j=1}^J U_{kj}}{\sum_{k=1}^K (t_k / \Delta t_k)} \geq 2.5 \times L, \quad (3.6)$$

where Q is the rated passenger capacity of a bus; K is time set $K = \{1, \dots, k, \dots, K\}$, K is the total time intervals; J is site set $J = \{1, \dots, j, \dots, J\}$, J is the amount of bus sites of a line; L is the length of the total bus line.

Table 2: Passenger volume in each site.

Time period	Passengers get in	Passengers get off	Time period	Passengers get in	Passengers get off
06:01–07:00	98	52	13:01–14:00	69	78
07:01–08:00	74	105	14:01–15:00	85	90
08:01–09:00	83	97	15:01–16:00	49	45
09:01–10:00	54	52	16:01–17:00	63	81
10:01–11:00	85	69	17:01–18:00	78	93
11:01–12:00	52	44	18:01–19:00	89	96
12:01–13:00	48	45	19:01–20:00	79	59

Table 3: Adjacent site's distance and average running time between sites.

Site no.	Adjacent site's distance (m)	Running time (min)	Site no.	Adjacent site's distance (m)	Running time (min)
1	0	0	18	397	1.87
2	406	2.375	19	562	3.6
3	308	1.28	20	491	2.85
4	562	3.7	21	374	1.73
5	461	2.75	22	259	1.25
6	364	1.54	23	376	1.65
7	347	1.37	24	549	3.2
8	570	3.3	25	523	3.0
9	440	2.5	26	1100	6.7
10	389	2.1	27	563	3.25
11	607	3.7	28	395	1.95
12	709	4	29	397	1.94
13	533	3.1	30	426	2.75
14	604	3.4	31	407	2.25
15	552	3.5	32	714	5.1
16	385	1.75	33	199	0.8
17	546	3.2	34	424	2.5

4. Practical Simulation Based on Optimal Model

A bus line in its upstream travel direction in Hohhot city in China was selected as the scheduling object. Several relevant data had been investigated and listed as follows: the number of total sites was $J = 34$ sites; the operation period was 6:00–20:00 and divided into $K = 14$ hours; the lower and upper bounds of bus dispatching interval were $t_{\min} = 2$ min and $t_{\max} = 15$ min, with the rated passenger capacity $Q = 100$ persons.

4.1. Passenger Traffic Flow and Basic Data on Bus Site Facilities

Data on passenger volume in each site of the upstream line were shown in Table 2 and adjacent sites distance and average running time between them were displayed in Table 3.

Table 4: Bus dispatching interval schedule based on ABFA.

Interval	Bus departure schedule									Departure times
1	6:00	:04	:08	:12	:16	:20	:24	:28	:32	15
	:36	:40	:44	:48	:52	:56				
2	7:00	:03	:06	:09	:12	:15	:18	:21	:24	20
	:27	:30	:33	:36	:39	:42	:45	:48	:51	
	:54	:57								
3	8:00	:04	:08	:12	:16	:20	:24	:28	:32	15
	:36	:40	:44	:48	:52	:56				
4	9:00	:05	:10	:15	:20	:25	:30	:35	:40	12
	:45	:50	:55							
5	10:00	:05	:10	:15	:20	:25	:30	:35	:40	12
	:45	:50	:55							
6	11:00	:05	:10	:15	:20	:25	:30	:35	:40	12
	:45	:50	:55							
7	12:00	:05	:10	:15	:20	:25	:30	:35	:40	12
	:45	:50	:55							
8	13:00	:06	:12	:18	:24	:30	:36	:42	:48	10
	:54									
9	14:00	:06	:12	:18	:24	:30	:36	:42	:48	10
	:54									
10	15:00	:05	:10	:15	:20	:25	:30	:35	:40	12
	:45	:50	:55							
11	16:00	:04	:08	:12	:16	:20	:24	:28	:32	15
	:36	:40	:44	:48	:52	:56				
12	17:00	:03	:06	:09	:12	:15	:18	:21	:24	20
	:27	:30	:33	:36	:39	:42	:45	:48	:51	
	:54	:57								
13	18:00	:04	:08	:12	:16	:20	:24	:28	:32	15
	:36	:40	:44	:48	:52	:56				
14	19:00	:06	:12	:18	:24	:30	:36	:42	:48	10
	:54									
Total bus departure times										190

4.2. Simulation Results and Analysis

The bus dispatching interval optimal model was built and simulated on ABFA comparison with GA and BFA. First, basic settings were initialized: the population scale was 40. The maximum iterations was 1000. $\alpha = \beta = 10$. $C_{\text{initial}} = 0.1$. $\epsilon_{\text{initial}} = 100$. The parameters of BFA and GA were similarly set. Three graphs showing convergence trend were given in Figure 2.

It could be drawn from Figure 2 that ABFA was the best one to accelerate convergence speed, improve searching precision, and strengthen robustness. So with the desired algorithm, latest bus interval was calculated in Table 4. Each bus departure interval was

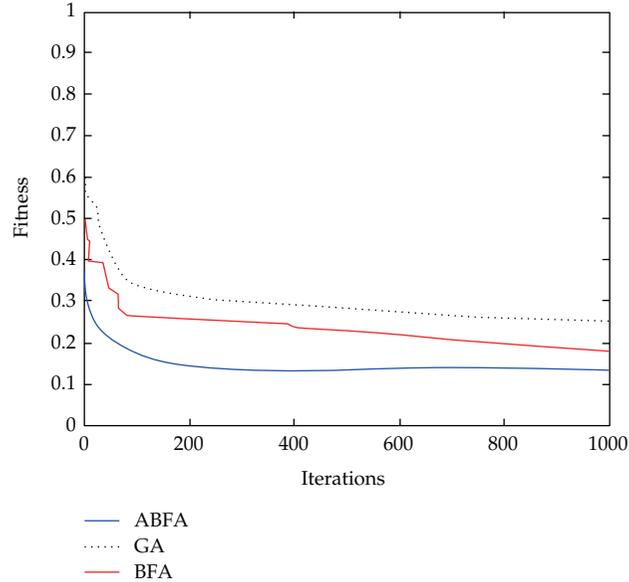


Figure 2: The convergence trend based on 3 algorithms [16].

$\Delta t_1 = 4; \Delta t_2 = 3; \Delta t_3 = 4; \Delta t_4 = 5; \Delta t_5 = 5; \Delta t_6 = 5; \Delta t_7 = 5; \Delta t_8 = 6; \Delta t_9 = 6; \Delta t_{10} = 5;$
 $\Delta t_{11} = 4; \Delta t_{12} = 3; \Delta t_{13} = 4; \Delta t_{14} = 6.$

5. Conclusions

In this paper, an interval optimal model was applied into one bus line in Hohhot city in China. The bus interval dispatching issue was considered as a nonlinear and multiobjective optimization. The enterprise profits, load factor, and dispatching interval were particularly chosen as the primary parameters in this optimization. Two relevant objective functions were defined. The proposed optimal algorithm, ABFA, combined colony sensing, chemotactic action, and improved foraging strategy to solve the distributed bacterial optimization. For comparison purpose, two other algorithms, GA and BFA, were employed to decide which was the best one. Furthermore, extensive sight would be thrown years ahead to evaluate the practical merits of ABFA in traffic and transportation optimization problems.

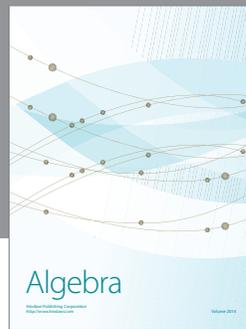
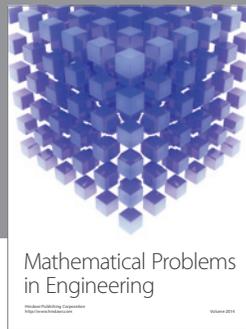
Acknowledgment

This work is supported by funded projects: National Key Basic Research Program of China (SN: 2012CB723303).

References

- [1] I. A. Farhat and M. E. El-Hawary, "Dynamic adaptive bacterial foraging algorithm for optimum economic dispatch with valve-point effects and wind power," *IET Generation, Transmission and Distribution*, vol. 4, no. 9, pp. 989–999, 2010.
- [2] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

- [3] E. Daryabeigi, M. Moazzami, A. Khodabakhshian, and M. H. Mazidi, "A new power system stabilizer design by using smart bacteria foraging algorithm," *Canadian Conference on Electrical and Computer Engineering*, Article ID 6030547, pp. 713–716, 2011.
- [4] M. Tripathy, S. Mishra, L. L. Lai, and Q. P. Zhang, "Transmission loss reduction based on FACTS and bacteria foraging algorithm," in *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature*, pp. 222–231, 2006.
- [5] D. H. Kim and J. H. Cho, "Adaptive tuning of PID controller for multivariable system using bacterial foraging based optimization," in *Proceedings of the 3rd International Atlantic Web Intelligence Conference on Advances in Web Intelligence*, pp. 231–235, 2005.
- [6] M. A. Muñoz, S. K. Halgamuge, W. Alfonso, and E. F. Caicedo, "Simplifying the bacteria foraging optimization algorithm," in *Proceedings of the 6th IEEE World Congress on Computational Intelligence*, pp. 1–7, 2010.
- [7] S. Dasgupta, A. Biswas, A. Abraham, and S. Das, "Adaptive computational chemotaxis in bacterial foraging algorithm," *IEEE Computer Society*, vol. 6, pp. 64–72, 2008.
- [8] H. Chen, Y. Zhu, and K. Hu, "Self-adaptation in bacterial foraging optimization algorithm," in *Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering (ISKE '08)*, pp. 1026–1031, November 2008.
- [9] F. A. Kidwai, "A genetic algorithm based bus scheduling model for transit network," *Proceedings of the Eastern Asia Society For Transportation Studies*, vol. 5, pp. 477–489, 2005.
- [10] E. B. Jacob, I. Becker, Y. Shapira, and H. Levine, "Bacterial linguistic communication and social intelligence," *Trends in Microbiology*, vol. 12, no. 8, pp. 366–372, 2004.
- [11] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 61–73, 2005.
- [12] W. J. Tang, Q. H. Wu, and J. R. Saunders, "Bacterial foraging algorithm for dynamic environments," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1324–1330, 2006.
- [13] S. D. Müller, J. Marchetto, S. Airaghi, and P. Koumoutsakos, "Optimization based on bacterial chemotaxis," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 16–29, 2002.
- [14] R. P. Gendron and J. E. R. Staddon, "Searching for cryptic prey: the effect of search rate," *American Naturalist*, vol. 121, no. 2, pp. 172–186, 1983.
- [15] G. Zhengwei, H. L. Pang, and D. W. Wang, "Adaptive bacterial foraging optimization and its application for bus scheduling," *Journal of System Simulation*, vol. 23, no. 6, pp. 1151–1160, 2011.
- [16] S. Sumathi, T. Hamsapriya, and P. Surekha, *Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*, 2008.
- [17] H. Chen, Y. Zhu, and K. Hu, "Adaptive bacterial foraging optimization," *Abstract and Applied Analysis*, vol. 2011, Article ID 108269, 7 pages, 2011.
- [18] S. Dasgupta, S. Das, A. Abraham, and A. Biswas, "Adaptive computational chemotaxis in bacterial foraging optimization: an analysis," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 919–941, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

