

Research Article

An Improved Particle Swarm Optimization for the Automobile Spare Part Warehouse Location Problem

Zhen Yaobao,¹ Hu Ping,² and Yang Shu²

¹ School of Mechanical Engineering, Dalian University of Technology, Dalian, 116024, China

² School of Automotive Engineering, Dalian University of Technology, Dalian 116024, China

Correspondence should be addressed to Yang Shu; yangs_dl@126.com

Received 29 September 2013; Revised 2 November 2013; Accepted 2 November 2013

Academic Editor: Rui Mu

Copyright © 2013 Zhen Yaobao 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.

This paper deals with a real-life warehouse location problem, which is an automobile spare part warehouse location problem. Since the automobile spare part warehouse location problem is a very complex problem, particle swarm optimization is used and some improved strategies are proposed to improve the performance of this algorithm. At last, the computational results of the benchmark problems about warehouse location problems are used to examine the effectiveness of particle swarm optimization. Then the results of the real-life automobile spare part warehouse location problem also indicate that the improved particle swarm optimization is a feasible method to solve the warehouse location problem.

1. Introduction

The profitability of automobile spare part warehouse location for automobile spare part factories is important for several reasons. The last decade has witnessed greatly growth in the demand of automobile and commensurate with this growth is the large demand of automobile spare parts. The cost of the management and stock has been an excess burden for automobile spare part factories. Furthermore, the fixed cost of the vehicles involved in automobile spare part delivery is high enough. Automobile spare part warehouses are a part of an overall effort to gain place and time utility (a decision aid in warehouse site selection). It has some obvious advantages. For example, a warehouse can hold stocks to match the imbalance between supply and demand. And the travelling cost can be decreased by collecting from multiple sources into a single vehicle to the final destination. Therefore, these increasing pressures and the advantages of warehouse have caused these factories to look for an automobile spare part warehouse to minimize their management cost.

There have been many literatures on the location of warehouse. Michel and van Hentenryck [1] proposed a tabu search to solve the warehouse location problem which considered the fixed and the transportation costs from warehouse to

stores. In the algorithm, a linear neighborhood was used to improve the performance of the tabu search. Khumawala [2] proposed an uncapacitated warehouse location problem which aimed to minimize the fixed cost of warehouse and the warehouse operating costs and transportation costs from the warehouse to customers. Then he attempted to solve the warehouse location problem by using of an efficient heuristic procedure which was derived from the branching decision rules. Baker [3] introduced a partial dual algorithm to solve the capacitated warehouse location problem which is based on the properties of transportation problems. Ozsen et al. [4] proposed a model of the capacitated warehouse location with risk pooling, in which the fixed facility location, transportation, and inventory carrying costs are considered simultaneously. Then, they tried to use a Lagrangian relaxation algorithm to solve this problem.

From the literatures, it can be attained that warehouse location problem has been recognized by academics and practitioners. Most of them paid attentions to the fixed cost of warehouse or transportation cost from warehouse to customers. However, in the real-life automobile spare part warehouse problem, the transportation cost from the factories to warehouse should also be considered. However, only few literature works attempted to incorporate routing

from the suppliers to the warehouse and the warehouse to customers in location analysis. This paper attempted to solve a real-life automobile spare part warehouse location problem, in which there are some automobile spare part factories and their customers.

Therefore, the objective of the warehouse location problem is to minimize the sum of the transportation cost from the factories to the warehouse and from the warehouse to the customers and the fixed cost of warehouse. Based on the objective, it can be attained that the automobile spare part warehouse location problem is a generalization of well-known and difficult location problems. It is therefore a large and complex problem. Many literature works suggested that heuristic algorithm was often a first choice to solve this kind of complicated problems [5–12]. Among heuristic algorithms, particle swarm optimization (PSO) is a heuristic which is a population-based search method developed by Kennedy and Eberhart [13]. PSO simulates the social behaviors from bee swarm, fish school, or bird flock. Thus, in PSO, each solution of optimization problem is corresponding to the position of one particle in the searching space. PSO attains the search for optimum based on the mechanism which adapts to the global and local exploration. Most applications of PSO have indicated that PSO have concentrated on the optimization in continuous space. In recent years, PSO has been a very popular optimization method due to that PSO does not need the calculation of derivatives, but the information from each particle and the information between the whole particle population. Thus, PSO is less sensitive to the nature of the objective function, which can be suitable for stochastic objective functions and can easily escape from local minima. PSO has been successfully applied to solving the complex problem [14–16]. Therefore, PSO is used to solve the automobile spare part warehouse location problem in this paper.

The remainder of the paper is organized as follows. Section 2 introduces the automobile spare part warehouse location problem. In Section 3, PSO and some improvement strategies are presented. Some computational results are discussed in Section 4 and, lastly, the conclusions are provided in Section 5.

2. The Automobile Spare Part Warehouse Location Problem

The automobile spare part warehouse location problem can be described as follows. There are a set of M potential warehouses and a set of N customers (repair station). Each warehouse has a fixed cost and the transportation cost from production factories to warehouse and from warehouse to customers. The automobile spare part warehouse location problem is to find a subset of warehouses and an assignment of warehouses to the customers aiming to minimize the fixed and the transportation costs. Once the locations of the warehouses are determinate, these warehouses should be assigned to serve their closest warehouse. Figure 1 described the network about the automobile spare part delivery.

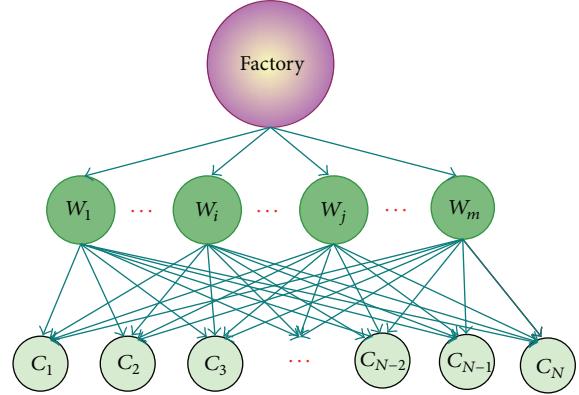


FIGURE 1: An example of the automobile spare part warehouse location problem.

From Figure 1, there are m depots for potential warehouse, that is, $w_1, \dots, w_i, \dots, w_j, \dots, w_m$. It is assumed that w_j is selected as the warehouse which should serve the customers. And it is possible for any set among $w_1, \dots, w_i, \dots, w_j, \dots, w_m$ to be the warehouse(s). Then, the automobile spare part warehouse location problem can be formulated as formulas (1a)–(1e):

$$\text{Min} \quad \left(\sum_{i=1}^M F_i y_i + \sum_{i=1}^M \sum_{j=1}^N C_{ij} \cdot d_{ij} \cdot x_{ij} + \sum_{i=1}^M \sum_{k=1}^K C_{ik} d_{ik} x_{ik} \right), \quad (1a)$$

$$\text{s.t.} \quad \sum_{i=1}^M F_i y_i \leq B, \quad (1b)$$

$$\sum_{j=1}^N x_{ij} \geq \sum_{k=1}^K x_{ik}, \quad (1c)$$

$$\sum_{i=1}^M \sum_{j=1}^N x_{ij} \leq 1, \quad (1d)$$

$$\sum_{i=1}^M \sum_{k=1}^K x_{ik} \leq 1, \quad (1e)$$

where

$$y_i = \begin{cases} 1 & \text{point } i \text{ is selected for the warehouse} \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

B is the maximum operative budget; M is the number of potential warehouse; N is the number of customers; K is the number of factories; C_{ij} and C_{ik} in formulas (1a)–(1e) stand for the costs from point i to j and from point i to k , respectively; in this paper, it is the unit transportation cost of unit distance and unit quantity. x_{ij} is the demand of customer i supplied from warehouse i . x_{ik} is the demand of warehouse i supplied from factory k . F_i is the fixed cost associated with warehouse i , where formulas (1a)–(1e) represent the objective function, as described above. Constraint (1b) ensures that

the fixed cost from selected warehouse cannot exceed the total planned budget. Constraint (1c) ensures the reliability of production of part factories. Constraints (1d)-(1e) ensure that each customer can be visited at most once.

3. Improved Particle Swarm Optimization for the Automobile Spare Warehouse Location Problem

3.1. Particle Swarm Optimization. PSO is a population-based search method, proposed by Kennedy and Eberhart [13], in which each individual is taken as a particle. Each particle has a position in the space of solutions and fly at a certain speed. The particles can dynamically adjust their position according to the fly experiences from themselves and their populations, that is, three possible directions that a particle can follow: one is its own path, the second is its best position, and the third is the best position of the population. In this way, the particle's personal best position and the global best position can be always updated and kept. Thus, the principles of the algorithm can be described as follow.

In PSO, the position of each particle is a solution of the problem. The fitness of each particle is based on the objective function of the problem. The position (X_i) and the speed (V_i) of particle i can be denoted as $(X_{i1}, X_{i2}, \dots, X_{iD})$ and $(X_{i1}, X_{i2}, \dots, X_{iD})$, respectively. At the iteration t , if the best fitness of particle i is p_i^t and the best fitness of the whole population is p_g^t , then, the velocity and the position of the particle i are updated using the following formula:

$$v_i^{t+1} = w v_i^t + c_1 \text{rand1}(p_i^t - X_i) + c_2 \text{rand2}(p_g^t - X_i), \quad (3)$$

$$X_i^{t+1} = X_i^t + v_i^{t+1}, \quad (4)$$

where t is the iteration counter; c_1 and c_2 are the acceleration coefficients which determine the effect the experiences from the particle and the population. w is the inertia weight which is used to control the impact of previous histories of velocities on current velocity. The inertia weight is a parameter which affects the trade-off between exploration and exploitation.

3.2. Random Inertia Weight. According to (3), it can be found that inertia weight is one of the most important parameters of PSO algorithm. In order to balance the global search ability and local search ability, the value of the inertia weight should be adjusted to achieve the goal. If the value of the inertia weight w was larger, the particle would have stronger global searching ability. While the value of the inertia weight w was smaller, the local searching ability of the particle would be enhanced. If the value of the inertia weight can obey a certain distribution of random number, it will randomize particle movement according to (3). Hence, two different particles may move to different position in the subsequent iteration even though they have similar position, personal best, and global best. It is common to have high inertia weight at the beginning of PSO iteration to lead to the particles exploring larger solution space, and low weight at the end allows the population following the cognitive and social term to exploit

the personal best and global best in the final phase. Based on the above analysis, the proposed random inertia weight ω can be described as the following formula:

$$\omega = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot \text{rand}() + \sigma \cdot \text{randn}() \cdot \frac{t}{T_{\max}}, \quad (5)$$

where λ_{\min} is the minimum value of the proposed random inertia weight ω ; λ_{\max} is the maximum value of the proposed random inertia weight ω ; $\text{rand}()$ denotes the uniformly distributed random numbers between 0 and 1; σ is deviation, which is used to measure the deviation degree between the random inertia and its average value and can control the direction of the random inertia to expected value. $\text{randn}()$ is a random of normal distribution. t is the number of the current iteration, and T_{\max} is the maximum number of the iterations.

3.3. Acceleration Coefficients. From the formula (3), the acceleration coefficients c_1 and c_2 can also be attained which are used to control the acceleration weight for each particle to the best fitness p_i^t and the best fitness p_g^t of the whole population, respectively. Thus, acceleration coefficient with low value allows particles to roam far from target regions before being tugged back, while high values for high value easily result in abrupt movement towards, or past, target regions [13]. If $c_1 = c_2 = 0$, the particles are directed where their velocity indicates, which will lead to the algorithm fail in finding the optimal solution. If $c_1 > 0$ and $c_2 = 0$, all the particles follow the best particle. There is not social information exchange between particles and then the model becomes a cognition-only model. Thus it is difficult for the model to find the optimal solution. In most literature works, $c_1 = c_2$ is selected.

The functions of c_1 and c_2 are changed so that the influence of the two factors can vary during the iterations. Therefore, the value of acceleration coefficient is also optimized to improve the performance of the algorithm. The following formulas for c_1 and c_2 can be used in this paper [17]:

$$c_1 = c_1^{\min} + \frac{c_1^{\max} - c_1^{\min}}{T_{\max}} \times t, \quad (6)$$

$$c_2 = c_2^{\min} + \frac{c_2^{\max} - c_2^{\min}}{T_{\max}} \times t,$$

where, c_1^{\min} , c_1^{\max} , c_2^{\min} , and c_2^{\max} denote the minimum and maximum values of c_1 and c_2 , respectively. t is the number of the current iteration and T_{\max} is the maximum number of the iterations.

3.4. Crossover Operation. Crossover operation is a reproduction operation in GA, which is used to exchange genetic information between two parent chromosomes at a predefined probability [10, 18, 19]. In this paper, crossover operation is used to help PSO algorithm to reach further solution space. In the crossover operation, half of the particles with better fitness are selected to enter into the next generation in each iteration. Simultaneously, the positions and speeds

of the former half-particles with better fitness substitute the corresponding vectors of the latter half-particles with worse fitness. However, during the crossover operation, the individual extrema of the latter is unchanged. In the crossover operation, any two of the latter half-particles will perform the crossover operation like genetic algorithm to children. Then half of the latter with better fitness enter into the next generation again. The crossover operation can increase the particle diversity, jump out of the local optimum, and accelerate the convergence speed.

The positions and speed vectors of the children particles are shown as follows:

$$\begin{aligned} \text{child}_1(x) &= p \times \text{parent}_1(x) + (1.0 - p) \times \text{parent}_2(x), \\ \text{child}_2(x) &= p \times \text{parent}_2(x) + (1.0 - p) \times \text{parent}_1(x). \end{aligned} \quad (7)$$

Among them, x is a D -dimensional position vector. $\text{Child}_k(x)$ and $\text{parent}_k(x)$ show the positions of the child particles and the parent particles, respectively. p is a D -dimensional even-distributed random vector, that is, crossover probability which is between 0 and 1.

Crossover operation can help the PSO algorithm to search in a larger space; however, in the late iteration, frequent crossover operation will make the algorithm run with more computing time and hard to be converged. Therefore, an adaptive method to adjust the crossover probability is used in this paper. The adaptive method can be found as follow:

$$P_c = P_{c0} + \sum_{k=1}^N \left[\frac{|f' - \bar{f}|}{(f_{\max} - \bar{f})} \right]^{[(t \times 2)/T_{\max}] + 1}, \quad (8)$$

where P_{c0} is the initial value of the crossover probability. f_{\max} and \bar{f} represent the best fitness and the average fitness of the population until now. f' is the current best fitness.

4. Case Studies

This paper attempts to use an improved PSO algorithm to solve the automobile spare part warehouse location problem. To examine the feasibility of the improved PSO algorithm (IPSO), some benchmarks for uncapacitated warehouse location from the standard OR library are selected in this paper. Then, the IPSO is used to solve a real-life automobile spare part warehouse location problem. The following will describe the two examples, respectively.

4.1. The Well-Known Uncapacitated Warehouse Location. In order to examine the performance of the proposed IPSO in this paper, the instances from the standard OR library [1] are selected. The information of the test problems can be shown in Table 1. VC++.NET 2003 is used to achieve the IPSO proposed in this paper, operating environment is the Pentium IV 2.93 GHz processor and 3 GB for the Windows platform. Large number of experiments to determine the parameters is set to the following: the number of particle is 50. The maximum number of the iterations is set to 1000. Also $c_1^{\min} = 0.5c_2^{\min} = 0.5c_2^{\max} = 2c_1^{\max} = 2$, $\lambda_{\min} = 0.5\lambda_{\max} = 0.95$, and $P_{c0} = 4$. In order to test the performance of the algorithm,

TABLE 1: The information of the test problems.

Bench	Size	Bench	Size	Bench	Size
Cap71	16×50	MO1	100×100	MR1	500×500
Cap72	16×50	MO2	100×100	MR2	500×500
Cap73	16×50	MO3	100×100	MR3	500×500
Cap74	16×50	MO4	100×100	MR4	500×500
Cap101	25×50	MO5	100×100	MR5	500×500
Cap102	25×50	MP1	200×200	MS1	1000×1000
Cap103	25×50	MP2	200×200	MS2	1000×1000
Cap104	25×50	MP3	200×200	MS3	1000×1000
Cap131	50×50	MP4	200×200	MS4	1000×1000
Cap132	50×50	MP5	200×200	MS5	1000×1000
Cap133	50×50	MQ1	300×300	MT1	2000×2000
Cap134	50×50	MQ2	300×300	MT2	2000×2000
Cap a	100×1000	MQ3	300×300	MT3	2000×2000
Cap b	100×1000	MQ4	300×300	MT4	2000×2000
Cap c	100×1000	MQ5	300×300	MT5	2000×2000

the results of the proposed IPSO algorithm are compared with those of tabu search [1]. Table 2 is the results from these algorithms for solving the warehouse location problems from the standard OR library.

From Table 2, it can be found that most results of IPSO algorithm have been close to the ones of tabu search, which indicate that the proposed IPSO algorithm is suitable for solving a warehouse location problem.

4.2. The Automobile Spare Part Warehouse Location Problem. The PSO algorithm is examined by the warehouse location problems from the standard OR library, which indicates that IPSO algorithm is suitable for solving the warehouse location problem. Then a real-life automobile spare part warehouse location problem needs to be solved by the PSO algorithm. In the real-life warehouse location problem, there is one automobile company that wants to large its business. Thus, the company wants to build some warehouses in a chosen area. Then the automobile spare parts will be sent from the company to the warehouses and from the warehouses to customers (e.g., repair station). There is one warehouse in the chosen area and three alternative points for selection. In the chosen area, there are four potential points for selection and ten customers need to be served. If the horizontal and vertical coordinates of the factory are assumed as (0,0), then, the information of the problem is shown in Tables 3, 4, and 5.

Then, the improved PSO continues calculating the automobile spare par warehouse location problem ten times. However, in the real-life seafood product delivery routing problems, the distance between two points is based on the length of the routes. The results are shown in Figure 2. It can be found that the optimization results are stable and the difference between the optimum and the worst plan is less than 2.1%. Furthermore, the computing time is from 320 to 410 seconds, which is acceptable for the larger real-life problem. Therefore, the proposed PSO has an excellent convergence performance to solve the automobile spare part

TABLE 2: Computational results using IPSO algorithm and tabu search.

Bench	Tabu search	IPSO	Bench	Tabu search	IPSO
Cap71	932 615.75	932 615.75	MP4	2633.56	2633.56
Cap72	977 799.40	977 799.40	MP5	2290.16	2290.16
Cap73	1 010 641.45	1 010 641.45	MQ1	3591.27	3591.27
Cap74	1 034 976.97	1 034 976.98	MQ2	3543.66	3543.66
Cap101	796 648.44	796 648.44	MQ3	3476.81	3476.82
Cap102	854 704.20	854 704.20	MQ4	3742.47	3742.47
Cap103	893 782.11	893 782.11	MQ5	3751.33	3751.33
Cap104	928 941.75	928 941.75	MR1	2349.86	2349.86
Cap131	793 439.56	793 439.56	MR2	2344.76	2344.76
Cap132	851 495.32	851 495.33	MR3	2183.24	2183.24
Cap133	893 076.71	893 076.71	MR4	2433.11	2433.11
Cap134	928 941.75	928 941.75	MR5	2344.35	2344.35
Cap a	17 156 454.4	17 156 454.48	MS1	4378.63	4378.63
Cap b	12 979 071.5	12 979 071.58	MS2	4658.35	4658.35
Cap c	11 505 594.3	11 505 594.33	MS3	4659.16	4659.16
MO1	1156.91	1156.91	MS4	4536.00	4536.01
MO2	1227.67	1227.67	MS5	4888.91	4888.91
MO3	1286.37	1286.37	MT1	9176.51	9176.51
MO4	1177.88	1177.88	MT2	9618.85	9618.85
MO5	1147.60	1147.61	MT3	8781.11	8781.11
MP1	2460.10	2460.10	MT4	9225.49	9225.49
MP2	2419.32	2419.32	MT5	9540.67	9540.67
MP3	2498.15	2498.15			

TABLE 3: The information of warehouse.

Item	W1	W2	W3	W4
Horizontal coordinate	23	88	72	45
Vertical coordinate	74	46	90	16
Land price	1.95	1.8	1.35	2.25

TABLE 4: The transportation cost from warehouses to customers and factories (Yuan/T).

Item	W1	W2	W3	W4
1	45	37.5	28.5	30
2	39	32.25	49.5	27
3	38.25	40.5	30	34.5
4	29.25	30.75	34.5	32.25
5	46.5	40.5	38.25	31.5
6	33.6	35.4	39.6	39
7	42	24	52	24.5
8	28	30	30	36
9	35	16	32	35
10	36	18	38	30
Factory	33.6	300	39.6	240

warehouse location problem. The optimized results of the problem can be seen in Table 6.

TABLE 5: The information of customers.

Customers	Horizontal	Vertical	Demand
1	25	80	3
2	30	23	0.8
3	70	60	0.75
4	92	51	3
5	42	64	1.05
6	87	45	2.1
7	39	7	1.5
8	28	71	3
9	55	20	4.5
10	14	56	1.5

TABLE 6: The optimized results of the automobile spare part warehouse location problem.

Item	W1	W2	W3	W4
The served customers	(1,5,8,10)	(3,4,6)	()	(2,7,9)
The treatment capability	5.2	3.9	0.0	4.6

From Table 6, it can be attained that the potential warehouse 3 is not suitable for warehouse and the potential warehouses 1, 2, and 4 are selected for warehouse location.

5. Conclusions

The automobile spare part warehouse location problem is the warehouse selection in potential depots in order to meet the demand of the customers and keep the automobile spare part factory competitiveness in a chosen area. Thus, the automobile spare part warehouse location problem is attempted to minimize the total cost which includes the transportation cost (from factories to warehouse and from warehouse to customers) and the fixed cost while meeting the constraints. Since the automobile spare part warehouse location problem is difficult to be solved. PSO is selected in this paper and some improved strategies are used to improve the performance of PSO. The computational results of some benchmark instances for uncapacitated warehouse location problem from the standard OR library suggest that the improved PSO is effective to solve the warehouse location problem. The results of the automobile spare part warehouse location problem can also indicate that the improved PSO is an effective method for the automobile spare part warehouse location problem.

The main contribution of this paper is to test the feasibility of PSO for the automobile spare part warehouse location problem. Our future research work will be on the models with stochastic demand, model and solve the model by other convenient methods, and possibly have added a capacity constraint on the multiple central warehouses.

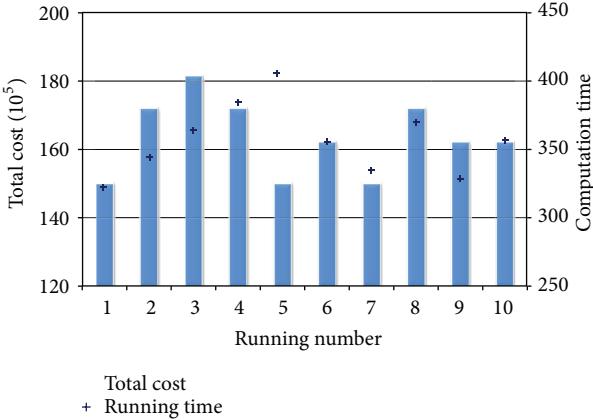


FIGURE 2: Computing results of the improved PSO after running 10 times.

Acknowledgments

This work was supported by the National Science Foundation for Postdoctoral Scientists of China 2013M530924 and the National Natural Science Foundation of China 51208079 and 11272075.

References

- [1] L. Michel and P. van Hentenryck, “A simple tabu search for warehouse location,” *European Journal of Operational Research*, vol. 157, no. 3, pp. 576–591, 2004.
- [2] B. M. Khumawala, “An efficient heuristic procedure for the capacitated warehouse location problem,” *Naval Research Logistics Quarterly*, vol. 21, no. 4, pp. 609–623, 1974.
- [3] B. M. Baker, “A partial dual algorithm for the capacitated warehouse location problem,” *European Journal of Operational Research*, vol. 23, no. 1, pp. 48–56, 1986.
- [4] L. Ozsen, C. R. Coullard, and M. S. Daskin, “Capacitated warehouse location model with risk pooling,” *Naval Research Logistics*, vol. 55, no. 4, pp. 295–312, 2008.
- [5] G. Chen, K. Govindan, and Z. Yang, “Managing truck arrivals with time windows to alleviate gate congestion at container terminals,” *International Journal of Production Economics*, vol. 141, no. 1, pp. 179–188, 2012.
- [6] G. Chen and Z. Yang, “Optimizing time windows for managing export container arrivals at Chinese container terminals,” *Maritime Economics & Logistics*, vol. 12, no. 1, pp. 111–126, 2010.
- [7] B. Y. Chen, W. H. K. Lam, A. Sumalee, and H. Shao, “An efficient solution algorithm for solving multi-class reliability-based traffic assignment problem,” *Mathematical and Computer Modelling*, vol. 54, no. 5–6, pp. 1428–1439, 2011.
- [8] B. Y. Chen, W. H. K. Lam, A. Sumalee, Q. Q. Li, H. Shao, and Z. X. Fang, “Finding reliable shortest paths in road networks under uncertainty,” *Networks & Spatial Economics*, vol. 13, no. 2, pp. 123–148, 2013.
- [9] B.-Z. Yao, C.-Y. Yang, J.-B. Yao, and J. Sun, “Tunnel surrounding rock displacement prediction using support vector machine,” *International Journal of Computational Intelligence Systems*, vol. 3, no. 6, pp. 843–852, 2010.
- [10] B. Yu and Z. Z. Yang, “An ant colony optimization model: the period vehicle routing problem with time windows,” *Transportation Research E*, vol. 47, no. 2, pp. 166–181, 2011.
- [11] B. Yu, Z.-Z. Yang, and B. Yao, “An improved ant colony optimization for vehicle routing problem,” *European Journal of Operational Research*, vol. 196, no. 1, pp. 171–176, 2009.
- [12] B. Yu, Z.-Z. Yang, P.-H. Jin, S.-H. Wu, and B.-Z. Yao, “Transit route network design-maximizing direct and transfer demand density,” *Transportation Research C*, vol. 22, pp. 58–75, 2012.
- [13] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.
- [14] A. W. Mohammed, N. C. Sahoo, and T. K. Geok, “Solving shortest path problem using particle swarm optimization,” *Applied Soft Computing Journal*, vol. 8, no. 4, pp. 1643–1653, 2008.
- [15] T.-L. Lin, S.-J. Horng, T.-W. Kao et al., “An efficient job-shop scheduling algorithm based on particle swarm optimization,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 2629–2636, 2010.
- [16] T. J. Ai and V. Kachitvichyanukul, “A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery,” *Computers & Operations Research*, vol. 36, no. 5, pp. 1693–1702, 2009.
- [17] Y. Marinakis, G. R. Iordanidou, and M. Marinaki, “Particle swarm optimization for the vehicle routing problem with stochastic demands,” *Applied Soft Computing*, vol. 13, no. 4, pp. 1693–1704, 2013.
- [18] B. Yu, Z. Yang, and C. Cheng, “Optimizing the distribution of shopping centers with parallel genetic algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 2, pp. 215–223, 2007.
- [19] B. Yu, Z. Yang, and C. Cheng, “Optimizing the distribution of shopping centers with parallel genetic algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 2, pp. 215–223, 2007.

