An Improved Ant Colony Algorithm to Solve Vehicle Routing Problem with Time Windows

Yi Yunfei[1,2,3(✉)], Lin Xiaodong[1], Sheng Kang[2], and Cai Yongle[2]

[1] College of Computer and Information Engineering, Hechi University, Yizhou 546300, China
gxyiyf@163.com
[2] Computer School, Wuhan University, Wuhan 430079, China
[3] Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning 530006, China

Abstract. This paper presents an improved ant colony optimization algorithm (ACO algorithm) based on Ito differential equations, the proposed algorithm integrates the versatility of Ito thought with the accuracy of ACO algorithm in solving the vehicle routing problem (VRP), and it executes simultaneous move and wave process, and employs exercise ability to unify move and wave intensity. Move and wave operator rely on attractors and random perturbations to set the motion direction. In the experiment part, this improved algorithm is implemented for solving vehicle routing problem with soft time windows (VRPSTW), and tested by Solomon Benchmark standard test dataset, the result shows that the proposed algorithm is effective and feasible.

Keywords: Ant colony optimization algorithm · Vehicle routing problem · Move operator · Wave operator

1 Introduction

Vehicle Routing Problem is firstly proposed by G. Dantzig and J. Ramser in 1959, ever since extensive attention are given on this subject by large number of experts and scholars, and the issue has become one of the hot topic in combination optimization. As a result, the problem is well expanded in accordance with various facts in life by experts, and lots of branches have been developed, such as Split and Simultaneous Pickup and Delivery Vehicle Routing Problem with Time Windows Constraint [1] (SVRPS-PDTW), Optimal Vehicle Routing With Real-Time Traffic Information [2], Capacitated Vehicle Routing Problem [3] (CVRP), Heterogeneous Fleet Vehicle Routing Problem [4] (HFVRP), etc. In summary, there exists a bunch of algorithms aimed to solve these problems, which can be classified into three categories, thus precise algorithm, heuristics algorithm and meta-heuristics algorithm. In recent years, meta-heuristic algorithm has

This work was supported by the Open Foundation of Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis (No. HCIC201411), the National Undergraduate Training Programs for Innovation and Entrepreneurship (No. 201410605055), and the Education Scientific Research Foundation of Guangxi Province (No. KY2015YB254).

D.-S. Huang et al. (Eds.): ICIC 2015, Part I, LNCS 9225, pp. 11–22, 2015.
DOI: 10.1007/978-3-319-22180-9_2

become the most commonly used algorithm for solving VRP problems, these high-precision algorithms can effectively solve lots of large-scale problems, such as particle swarm optimization (PSO), genetic algorithm (GA), ant colony optimization algorithm (ACO), simulated annealing algorithm (SA) and so on. However, as a general optimizer in practical applications, meta-heuristic optimization algorithm inevitably confront the contradiction of common versus proficient, exploration versus exploitation and efficiency versus accuracy. Based on these facts, a hybrid IT-AC algorithm is proposed by combining the ITO algorithm, which is first introduced by Professor Dong Wenyong and Professor Li Yuanxiang in Wuhan University [2], with ACO algorithm which has been deeply developed in solving VRP problems. By the means of combining the generality of ITO algorithm with the high accuracy of the ACO algorithm in solving VRP, the proposed optimization algorithm is applied in solving VRPSTW [5].

ITO algorithm (Evolutionary Algorithm inspired by Ito stochastic process, in order to reflect the origin of the algorithm, referred to as ITO) by Dong and Li [6] is based on random process of Ito process, it analyzes the movement of particles from the microscopic point of view, and mimic the mechanism of kinetics equation that particles colliding and interacting with others to design algorithm and solve problems, which reflects the features of group search in bionics. At present, time series modeling [7], function optimization [2], combinatorial optimization [8], multi-objective optimization [9] and other problems can obtain comparatively good solutions by utilizing ITO algorithm. ACO algorithm, for its natural advantages, ever since the time it was proposed in the field of combinatorial optimization, especially in the study of vehicle routing problem, has achieved much success. In this paper, therefore, based on ITO algorithm framework, the selection strategy in ACO algorithm is utilized to design ITO-ACO algorithm.

In the experiment part, Solomon Benchmark standard test dataset[1] are used as experimental data. The dataset includes a total of 56 data subsets, the dataset nodes can be divided based on the distribution of geographical position into three categories: class R, class C and class RC, while class R data nodes are randomly distributed, nodes in class C are in clustered distribution, and nodes in RC class are between class R and class C, which means some nodes are in randomly distribution, the others are in clustered distribution. Additionally, test dataset can be further classified into six different categories according to the level of the time schedule, named R1, R2, C1, C2, RC1, RC2.

2 Vehicle Routing Problem Description

Vehicle Routing Problem (VRP) is a well-known NP-hard problem, solving the problem effectively is of great importance in real life. Generally, VRP contains a warehouse, N clients and NV vehicles, each vehicle is required to be of same specifications that the cargo capacity is Q. The desired scheduling scheme is that the warehouse is designed as the starting point and finishing point in all of the suitable vehicle routes to ensure that all customers are visited only once by one vehicle, and the total transportation costs (e.g., path length, transit time, needed vehicles, etc.) are minimized. Certain constraints should be met when designing the route program as following:

---

[1] Source: http://web.cba.neu.edu/∼msolomon/problems.htm.

$$x_{kij} = \begin{cases} 1, & vehicle\ k\ move\ from\ i\ to\ j \\ 0, & otherwise \end{cases} \quad (i,j = 0,1,\ldots,N; k = 1,2,\ldots,NV) \tag{1}$$

$$y_{ki} = \begin{cases} 1, & vehicle\ k\ finish\ task\ i \\ 0, & otherwise \end{cases} \quad (i,j = 0,1,\ldots,N; k = 1,\ldots,NV) \tag{2}$$

$$\sum_{i=0}^{N} q_i y_{ki} \le Q\,(k = 1,2,\ldots,N) \tag{3}$$

$$\sum_{k=0}^{NV} y_{ki} = 1\,(i = 1,2,\ldots,N) \tag{4}$$

$$\sum_{j=0}^{N} x_{kij} = y_{ki}\,(i = 1,2,\ldots,N;\ k = 1,2,\ldots,NV) \tag{5}$$

$$\sum_{i=0}^{N} x_{kij} = y_{ki}\,(j = 0,1,\ldots,N; k = 1,2,\ldots,N) \tag{6}$$

Among them, the warehouse is numbered 0, customer demand point are numbered 1,2,3, …, N, $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$ denotes the demand for customer $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$, Eq. (3) indicates capacity constraints of vehicle; formula (4) represents each client point must be visited once, and only by one vehicle; formulas (5) and (6) represents the points $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$ in the path of vehicle $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$ are to be visited by vehicle $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$.

3 Vehicle Routing Problem with Time Windows

When solving VRPSTW problem, it usually have to be considered that the vehicle service time $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$ for each customer, and working hours (driving time + service time + wait time) of each vehicle cannot exceed the maximum limit (i.e. the time of going off work). Suppose a scheduling scheme $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$, where $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \le T$ (7), which represents that a vehicle will fulfill all of the customer's distribution tasks.

Now the working time cannot exceed the maximum operating time $\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \leq T$ can be formulated as following:

$$\sum_{i=0}^{N} d_{i,i+1}/speed + \sum_{i=1}^{N} \delta_i + \sum_{i}^{N} wt_i \leq T \tag{7}$$

Where $wt_i = \begin{cases} \infty & if\ CurTime > li \quad (arrive\ late) \\ ei - CurTime & if\ CurTime < ei \quad (arrive\ early) \\ 0 & else \quad (comform\ to\ the\ time\ window) \end{cases}$ denotes the path length between clients $wt_i = \begin{cases} \infty & if\ CurTime > li \quad (arrive\ late) \\ ei - CurTime & if\ CurTime < ei \quad (arrive\ early) \\ 0 & else \quad (comform\ to\ the\ time\ window) \end{cases}$ and $wt_i = \begin{cases} \infty & if\ CurTime > li \quad (arrive\ late) \\ ei - CurTime & if\ CurTime < ei \quad (arrive\ early) \\ 0 & else \quad (comform\ to\ the\ time\ window) \end{cases}$, speed stands for the velocity, in addition, despite of the common limits of VRP problem, there exists other constraints, such as: customer demand service time window [ei, li], if the demand is not met, there will be some punishment. The wait time function for serving customer $wt_i = \begin{cases} \infty & if\ CurTime > li \quad (arrive\ late) \\ ei - CurTime & if\ CurTime < ei \quad (arrive\ early) \\ 0 & else \quad (comform\ to\ the\ time\ window) \end{cases}$ is designed as follows.

$$wt_i = \begin{cases} \infty & if\ CurTime > li \quad (arrive\ late) \\ ei - CurTime & if\ CurTime < ei \quad (arrive\ early) \\ 0 & else \quad (comform\ to\ the\ time\ window) \end{cases} \tag{8}$$

Where CurTime represents the time that a vehicle reaches the customer point $f_{\min} = NV \times VecCost + PC \times \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{NV} d_{ijk} x_{ijk} + Cw \times \sum_{i=1}^{N} wt_i + Cs \times \sum_{i=1}^{N} \delta_i$ General VRP problems strive to find the shortest path, but there exists many limitations, for example, when in severe traffic congestion, the shortest path usually lead to long time operation to vehicle, cause high transportation costs. Instead, we set the minimum total cost as the goal, and the object function is designed as follows:

$$f_{\min} = NV \times VecCost + PC \times \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{NV} d_{ijk} x_{ijk} + Cw \times \sum_{i=1}^{N} wt_i + Cs \times \sum_{i=1}^{N} \delta_i \tag{9}$$

Where NV, VecCost, PC, Cw, Cs are the number of vehicles, vehicle costs, unit path cost, unit waiting time penalties and unit time service fee respectively.

4 The Design of Algorithm

When initially designed, ITO algorithm has been carefully designed by considering all elements of swarm intelligence algorithms, in order to be able to give a unified swarm intelligence algorithm model. Conventional swarm intelligence algorithm uses the exploration and exploitation process, namely firstly employ particles to explore the solution space, then update fitness value, utilize new fitness value for exploitation, and update the fitness value again. Such a process-oriented approach is neither consistent with the actual situation of mining industry, nor with the Brownian motion process on which ITO algorithm based, So starting from the particle movement and ITO algorithm itself, there comes out the thought of executing move process and wave process simultaneously, that is, particle movement is not conducted following mechanical steps (thus, moves are executed around the most attractive element, and waves are around the random attractors, i.e., two separate processes), instead, it's continuous motion under joint effects, that is, it is the movement under common effects of the most attractive element and random elements. This process is not only consistent with the facts of movement, it can also reduce costs effectively by lower the calculation times on the fitness value. Therefore, a solution based on ITO algorithm model is given in this paper. Additionally, we take into account of the natural advantages of ACO algorithm when applied in solving VRP problems and our in-depth study of merging it with VRP problems, some mechanisms of ACO algorithm are also borrowed, such as route update strategies, to improve our ITO algorithm which is not yet mature, in order to integrate the universality of ITO algorithm with the accuracy of ACO algorithm for solving VRP problems, there comes out the algorithm which named "improved ant optimization algorithm with soft time windows for solving vehicle routing problem" (referred to as IAO).

The proposed algorithm mainly includes three operators, i.e., radius operator which maintains the personality of particles, environmental temperature operator to exert macro-control, move and wave operators which optimize the routines by continuous learning from the feedbacks. Firstly, each particle utilizes own radius to maintain its personality, the size is related to the merits of its solution, and the personality of groups is employed to ensure the diversity of particles; Secondly, ambient temperature exert macro control on particle motion capability, that is, environmental temperature decreases with the increasing of iteration times to ensure that the algorithm converges gradually; then particle move and wave operators are designed based on particle radius and ambient temperature, they utilize respectively the optimal particle and random particle as attractor to implement move and wave movement, get new solution, and search for more optimal solutions in the motion. When designing move and wave operators (includes strength and process), we consider the intensity of move and wave as the reflection of particle exercise capability, therefore, these two operators, to some extent, are of one concept, so we no longer differentiate the move intensity and wave intensity when two operators are unified in this paper, it is indicated by exercise ability instead, which simplifies ITO algorithm design a lot. The move process and wave process are all attracted by attractors, while the attractor of move process is global optimal solution and that of wave process is randomly chosen solution. Finally, we use the path construction rules of ACO algorithm, thus the influence of moving and waving have unified effects on changing pheromone concentration, then scheduling scheme are constructed under the influence of the pheromone concentration. Afterwards, execute the process of move and wave simultaneously, so in the proposed it is further promoted for the integration of exploration and exploitation process.

Based on the above analysis, we conclude the design ideas as follows:

(1) Move and wave intensity are only the reflection of particle exercise capacity, in this paper the two intensity is replaced with unified exercise intensity, thus no longer treat them differently.
(2) Move refers to the process of attractor particles attracting current particles, so what need to do is increase the pheromone concentration of attractors.
(3) Wave means generating random perturbations under the influence of environment, so we just need to randomly select several paths from the environment to increase the pheromone concentration.
(4) Pheromone evaporates on all paths to ensure that pheromone concentration will not be too high in the environment.

The operator and pheromone concentration on each path are designed as follows:
Particle radius:

$$r_{n_i} = \frac{n - n_i}{n - 1} \tag{10}$$

Environment temperature:

$$T = \exp\left(-1/_T\right) \tag{11}$$

Motion capability:

$$f(r, T) = \gamma_{\min} + f_1(r) \times f_2(T) \times (\gamma_{\max} - \gamma_{\min}) \tag{12}$$

All explanation and design ideas of the above parameters see reference [8], and will not be described here.

All paths execute the evaporation process in line with the volatilization factor $\tau(i,j) = (1-\rho) \times \tau(i,j)$ (13), the formula is:

$$\tau(i,j) = (1-\rho) \times \tau(i,j) \tag{13}$$

Where (i,j) represents all paths.

The formula of increasing the pheromone concentration on optimal paths is as following:

$$\tau(i,j) = \tau(i,j) + \gamma \tag{14}$$

$if(i,j) \in$ all paths in optimal solutions. Where $\tau(i,j) = \tau(i,j) + \gamma$ represents the motion capability.

The increase of pheromone concentration on randomly selected path is according to the following formula:

$$\tau(i,j) = \tau(i,j) + \gamma \tag{15}$$

$\tau(i,j) = \rho \times \tau(i,j) + \begin{cases} \gamma \ if \ e(i,j) \in \sigma' \\ \gamma \ if \ e(i,j) \in \sigma \ and \ rand() < p \end{cases}$. Where the rand() is a function that generates numbers from 0 to 1, p is the probability of selecting random path, which is to control the wave intensity.

In summary, the pheromone updating formula is:

$$\tau(i,j) = \rho \times \tau(i,j) + \begin{cases} \gamma \ if \ e(i,j) \in \sigma' \\ \gamma \ if \ e(i,j) \in \sigma \ and \ rand() < p \end{cases} \tag{16}$$

Where $p^k(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum\limits_{l \notin tabu_k} [\tau(i,l)]^\alpha [\eta(i,l)]^\beta}, i \in tabu_k \cap j \notin tabu_k \\ 0, else \end{cases}$ denotes the paths that haven't been visited by certain particle and the optimal solution particle neither.

The solution construction method is as follows:

Learn the path generating method from ant colony algorithm, that is, according to the pheromone concentration and distance to calculate the probability of selecting each candidate edge, then utilize the roulette approach to select a candidate edge as the next path, iteratively run this method until the scheduling scheme is constructed. The probability is calculated as following:

$$p^k(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum\limits_{l \notin tabu_k} [\tau(i,l)]^\alpha [\eta(i,l)]^\beta}, i \in tabu_k \cap j \notin tabu_k \\ 0, else \end{cases} \tag{17}$$

Where $\eta(i,j)$ is the reciprocal of the distance, i.e., 1/dij, denotes as empirical knowledge, also called visibility. tabuk is a taboo table for storing customer demand points have been searched by vehicle k. α is a factor for controlling the importance measurement of edge weight in the probabilistic choice, and β is for controlling the effect of visibility (i.e., edge length factor).

IAO algorithm flow

Initialization: parameters, particle radius, ambient temperature and pheromone concentration

WHILE (termination condition is not satisfied)

step1. According to the pheromone concentration and path distance, follow the path selection rule, generate the scheduling scheme of each particle, and finally according to the probability to decide whether to accept this update or not.

step2. Update the fitness value of each particle, and according to the size of fitness value to sort all the scheduling schemes generated by current iteration in descending order

step3. Select the optimal solution, and update the global optimal solution

step4. Update each particle radius, ambient temperature and pheromone concentration, and calculate out each particle's exercise ability

step5. Increase the concentration of pheromone on the optimal path, lead the particles move towards the optimal solution; increase the pheromone concentration on random path to implement random wave process in the environment.

5 Experimental Results and Analysis

In this section, the VRPSTW problem is solved, and the solution is compared with the results in [10, 11] respectively, according to the comparison and analysis, effectiveness of the proposed algorithm is proved.

The experiment environment: Eclipse Helios Service Release 2. The related cost is calculated according to Solomon method: rental fee per vehicle is 500, wait fee per unit is 10, travel cost per path unit is 40, service cost per unit is 5, the speed is 1. Algorithm related parameters are set as follows: population size M = 50, maximum evolution generation GEN = 500, initial temperature of annealed table T = 1000, annealing table length TLength = 2, annealing rate $\rho = 0.99$, edge weight importance factor $\alpha = 5$, the importance factor of the distance between customer points $\beta = 3$, the probability of selecting a random path p = 0.3. The results are showed in Tables 1 and 2; Figs. 1, 2, 3 are illustrative diagram of the distance, the optimum value and the average value respectively, which are depicted based on the experimental data in Tables 1 and 2.

From Table 1, we compare the proposed algorithm with the experimental results in [10], mainly are the number of vehicles, path length, and total cost of the proposed algorithm is listed as well. As can be seen from Table 1, the proposed algorithm for C class problems can always get the optimal results or better results, while for R and RC class problems there exists certain bias to the optimal solution, mainly due to the design of the proposed algorithm is a macro-design based on our thoughts of the algorithm, while the specific design details such as initialization method, local search methods haven't yet been in detailed design, in addition, all of the running time of the proposed algorithm is limited in 50 s, while in [10] the algorithm running time are typically between 2−7 h. As a result, the effectiveness of the proposed IAO algorithm can be proved according to the results of Table 1.

From Table 2, comparisons have been drawn on the calculation of fitness value between the proposed algorithm and the algorithm in [11]. Literature [11] uses the "first-expired-first-serviced" algorithm (FEFS) to work out initial solution, then simulated annealing algorithm (SA) is employed to improve the initial solution, and

Table 1. Comparison between the algorithm in [10] and IAO

Pro	Best known		Algorithm in [10]		The proposed algorithm (IAO)			
	VN	Distance cost	VN	Distance cost	VN	Distance cost	Waiting cost	Total cost
C101	10	828.94	10	828.94	**10**	**828.94**	**0**	**83 157.47**
C106	10	828.94	10	828.94	10	835.42	49.64	83 913.36
C201	3	591.56	3	591.56	**3**	**591.56**	**0**	**70 162.26**
R103	14	1237.05	14	1287.0	17	1522.21	1149.64	85 884.80
			15	1264.2				
R108	10	960.26	10	971.91	12	1123.87	467.96	60 634.52
R203	4	935.04	3	1041.0	5	1320.55	1436.07	74 682.60
			5	995.8				
			6	978.5				
R204	3	789.72	3	1130.1	**3**	**950.53**	**780.77**	**52 328.90**
			4	927.7				
			5	831.8				
			6	826.2				
RC101	15	1636.92	15	1690.6	18	1871.01	731.33	96 153.80
			16	1678.9				
RC102	13	1470.26	15	1493.2	16	1817.48	618.87	91 888.09
RC105	16	1590.25	15	1611.5	18	1875.87	826.65	97 301.12
			16	1589.4				
RC108	10	1142.66	11	1156.5	12	1335.28	227.90	66 690.29

Table 2. Comparison between the algorithm in [11] and IAO

Pro	TS		Algorithm in [11] (SA)		The proposed algorithm (IAO)	
	Optimal	Average	Optimal	Average	Optimal	Average
R101	128 005.12	129 862.66	121 122.50	123 229.51	104 808.41	110 506.79
R102	132 038.25	137 941.96	121 516.61	127 056.92	100 104.02	102 570.48
R201	120 492.04	125 031.48	110 738.33	114 076.51	88 834.95	94 294.84
R202	110 849.52	120 857.59	101 296.37	111 122.22	89 127.64	91 597.10
C101	153 800.02	160 337.92	136 719.63	143 741.30	83 157.47	88834.47
C102	164 917.74	166 576.88	146 608.96	154 561.76	91 751.71	104 462.04
C201	117 154.81	120 497.33	102 681.24	111 295.40	70 162.26	91 561.28
C202	129 560.34	133 189.78	120 051.88	124 762.01	70 966.79	99 670.66
RC101	119 678.16	121 419.28	108 689.43	113 520.80	96 835.02	102 032.03
RC102	128 343.59	133 229.97	112 415.05	120 523.93	88 693.09	95 680.04
RC201	155 422.90	156 767.73	150 829.97	154 549.38	99 663.79	103 191.14
RC202	132 486.31	139 866.27	119 488.16	126 091.85	92 590.15	99 073.89

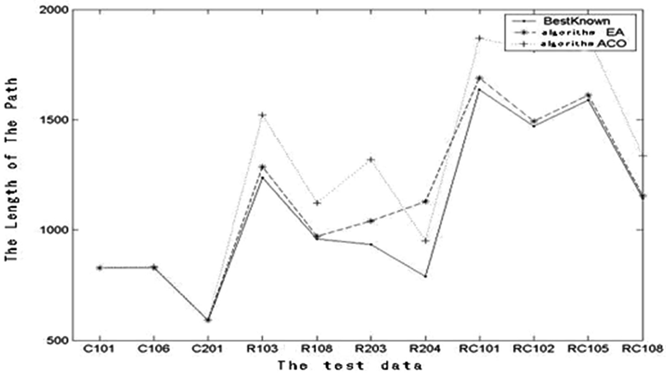

Fig. 1. Distance comparison

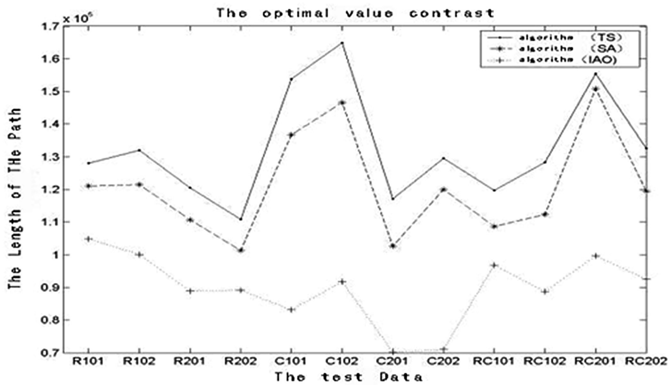

Fig. 2. Optimal value comparison

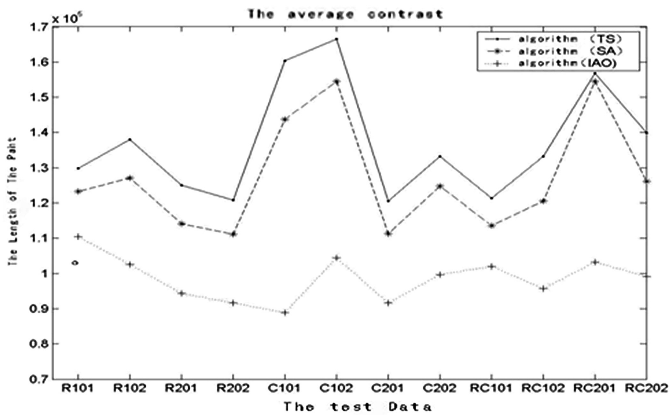

Fig. 3. Average value comparison

ultimately the satisfactory solution is obtained, this corresponds to the results with SA algorithm in Table 2; TS refers to the tabu search algorithm used for comparison with the algorithm proposed in [11]. The bold ones indicate the best solutions among the three algorithms listed. Each data set is executed for 20 times respectively, “Optimal” and “Average” mean the optimal solution and average solution respectively among the 20 iterations. In the experiment, first two cases of the Solomon’s six categories are selected, it can be seen from the experimental results that the optimal solution and the average value of the proposed algorithm are significantly better than the two algorithms in [11], but in [11] the execution time of the algorithm are about 3 s, while it is comparatively longer in the proposed algorithm, which are between 22.5-48.5 s, in fact, it’s also acceptable. Despite of the time, the obvious better results can also prove the effectiveness of IAO algorithm proposed in this paper.

6 Conclusion

In this paper, Vehicle Routing Problem is firstly analyzed, then the Ito stochastic differential thought is integrated with ACO algorithm, and the intensity of move operator and wave operator are unified to be reflected by exercise ability. In addition, a new pheromone updating rule is given to be more in line with Brownian motion and ant optimization rule. From the comparison results between the proposed algorithm and that in [10, 11], it is demonstrated that the proposed IAO algorithm can effectively solve VRPSTW problem. Surely, there remains certain settings to be further researched, for example, the parameters of the proposed algorithm can be better adjusted, thereby enhancing the efficiency and accuracy of the algorithm. Additionally, when setting the parameters of the algorithm, it is mostly based on certain assumptions and analysis on large amounts of data, rather than on scientific setting rule, these will be further researched in the future.

References

1. Kim, S., Lewis, M.E., White III, C.C.: Optimal vehicle routing with real-time traffic information. IEEE Trans. Intell. Transp. Syst. **6**(2), 178–188 (2005)
2. Dong, W., et al.: BBOB-benchmarking: a new evolutionary algorithms inspired by ITO process for noiseless function tested. J. Comput. Inf. Syst. pp. 2195–2203 (2011)
3. Korenevskiy, N., Al-Kasasbeh, R.T., Ionescouc, F., Alshamasin, M., Alkasasbeh, E., Smith, A.P.: Fuzzy determination of the human’s level of psycho-emotional. In: Toi, V.V., Toan, N. B., Dang Khoa, T.Q., Lien Phuong, T.H. (eds.) 4th International Conference on Biomedical Engineering in Vietnam. IFMBE Proceedings, vol. 40, pp. 213–216. Springer, Heidelberg (2013)
4. Hongbin, L.: Study on multi-depots and multi-vehicles vehicle scheduling problem based on improved particle swarm optimization. Comput. Eng. Appl. **50**(7), pp. 251−253 (2014) (in Chinese)
5. Qiuyun, W., Wenbao, J.: Solving algorithm of vehicle routing problem with soft time windows. J. Beijing Inf. Sci. Technol. Univ. **28**(4), pp. 57−59 (2013) (in Chinese)

6. Wen, M., Cordeau, J.-F., Laporte, G., et al.: The Dynamic Multi-PeriodVehicle Routing Problem. DTU Management Engineering, Denmark (2009)
7. Dong, W.: Time series modeling based on ITO algorithm. In: Proceedings of the International Conference on Natural Computation, pp. 398–402 (2007)
8. Wenyong, D., Wensheng, Z., Ruiguo, Y.: Convergence and runtime analysis of ITO algorithm for one class of combinatorial optimization. chinese. J. Comput. **34**(4), pp. 636–646 (2011) (in Chinese)
9. Dong, W.: The multi-objective ITO algorithms. In: Proceedings of the International Conference on Intelligence Computation and Application, pp. 21–23 (2007)
10. Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R.: Multiobjective dynamic vehicle routing problem with fuzzy travel times and customers' satisfaction in supply chain management. IEEE Trans. Eng. Manage. **60**(4), 777–790 (2013)
11. MA, H., Jin, P., Yang, S.: Heuristic methods for time-dependent vehicle routing problem. J. Syst. Eng. **27**(2), pp. 256−262 (2012) (in Chinese)

http://www.springer.com/978-3-319-22179-3

Intelligent Computing Theories and Methodologies
11th International Conference, ICIC 2015, Fuzhou,
China, August 20-23, 2015, Proceedings, Part I
Huang, D.-S.; Bevilacqua, V.; Premaratne, P. (Eds.)
2015, XXIX, 755 p. 309 illus., Softcover
ISBN: 978-3-319-22179-3