

ITO-based evolutionary algorithm to solve traveling salesman problem

Wenyong Dong^{1,2}, Kang Sheng¹, Chuanhua Yang³, Yunfei Yi^{1,4,*}

¹Computer School, Wuhan University, Wuhan, China

²Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, USA

³Changjiang Wuhan Waterway Bureau, Wuhan, China

⁴Department of Computer and Information Science, Hechi University, Yizhou, China

E-mail: gxyiyf@163.com

Abstract. In this paper, a ITO algorithm inspired by ITO stochastic process is proposed for Traveling Salesmen Problems (TSP), so far, many meta-heuristic methods have been successfully applied to TSP, however, as a member of them, ITO needs further demonstration for TSP. So starting from designing the key operators, which include the move operator, wave operator, etc, the method based on ITO for TSP is presented, and moreover, the ITO algorithm performance under different parameter sets and the maintenance of population diversity information are also studied.

1. Introduction

ITO algorithm is based on Ito stochastic process by simulating the kinetic law of the particles collide with each other in imitation of the particle system to design the algorithm and problem solving. It's a new algorithm analyzing the law of motion of the particle from the microscopic point of view, and then be proposed through abstraction and simulation methods. ITO algorithm on the one hand reflects the characteristics of the population search in biomimetic evolutionary algorithm, and in these algorithms the solution is usually represented (encoded) as a particle and a large number of particles build up the object processed by ITO algorithm, namely particle system; on the other hand, we use ITO process theory to analyze the algorithm, such as use the Ito stochastic integral method to establish the dynamic equations of the algorithm, and we usually design two key operators: the move rate and the wave rate according to the law of macromolecular movement proposed by Einstein, Langevin as well as the law of particle thermal motion, therefore, it contains some characteristics of the simulated annealing algorithm, because of which the proposed algorithm is named [1-4]. In this paper we will apply ITO algorithm to the optimization of the TSP problem and we will study the diversity of mechanisms in the search process of ITO algorithm.

2. ITO algorithm for TSP

In this section we introduce the idea of ITO algorithm and the operator for TSP in detail, including the determination of the particle radius, the move operator, the wave operator and the temperature of the annealing environment.

* Yunfei Yi(1981-), male, Ph.D., associate professor, research interests include intelligent computing, machine learning, etc. This work was in part supported by the NSF of China under Grant Nos. 61170305, 60873114 and the State Scholarship Fund (File No, 2012084291) of CSC.



2.1. Process of the algorithm

Its process is proposed mainly inspired by the law of thermal motion of particles, likely, we mimic the swarm intelligence algorithms, such as genetic algorithms, ant colony optimization, etc., to map the Ito process to an algorithm that can be used to solve the optimization problems. Besides, ITO algorithm is also designed to be population-based algorithm. The iterative process is as follows:

- (1) Initialize a population of N particles utilizing some strategies, set relevant parameters of the algorithm;
- (2) Assess the particles in the population;
- (3) If the stop conditions are not met, perform the following loop :
 - Find the best particle and the worst particle in the population;
 - Each particle choose the attractive center (or called the attractor or attractive element, we will mix these concepts in the rest of this article) for move operator;
 - Calculate the annealing temperature, the radius of each particle, the move operator and the wave operator;
 - Each particle moves in accordance with the move operator;
 - Each particle moves to a new location in accordance with the wave operator;
 - Evaluate the solution that each particle represents ;
- (4) Output the optimal solution

2.2. The design of several key factors

In order to achieve desired results, we need to design a few key factors including the particle radius, wave operator, move operator, annealing table and the selection of the maximum intensity and the minimum intensity.

A better approach to design the particle radius is the method based sorting, The radius of each particle is calculated as follows:

$$r(x) = r_{\max} - \frac{(r_{\max} - r_{\min})}{n-1} \times (n_i - 1) \quad (1)$$

The particle radiuses generated by this method distribute evenly in $[r_{\min}, r_{\max}]$, regardless of the objective function value of each particle, this approach makes the algorithm has a strong search capabilities even in the late period, therefore it's more suitable for complex function.

3. Parameter sensitivity configuration and population diversity experiment

There are several parameters in ITO algorithm to be settled when we try to solve TSP problem: the number of particles, the upper and lower bounds of wave, the method to select the attractive center, annealing strategy, and the method to calculate the radius.

3.1. The impact of the lower and upper bounds in waves

The wave operator has a decisive influence on the global convergence, and we can prove that when the wave lower bound is 0, the algorithm is not convergent, but when the upper bound and lower bound take the same value, the wave operator turns into a pure random algorithm. In order to study the impact of wave with the upper and lower bounds, we can take $\gamma_{\max} = 1 - \sqrt[n]{p_{\min}}$, $\gamma_{\min} = 1 - \sqrt[n]{p_{\max}}$ when design the algorithm, so once we know p_{\min} and p_{\max} , we can get γ_{\min} and γ_{\max} . We select 3 values 0.0005, 0.05, 0.5 as the value of p_{\min} , and select 4 values 0.1, 0.4, 0.5, 0.8 as the value of p_{\max} we also remove the combinations of $p_{\min} > p_{\max}$. Table 1 are the results of our experiments.

Table 1. Wave operator in the upper and lower bounds for different combinations of four cases, Data in the table is the averages of 30 independent runs, the percentage in brackets indicate the gap between the average path length and the optimal solution. Bold part is the best result.

Instance	(0.0005, 0.1)	(0.0005, 0.4)	(0.0005, 0.5)	(0.0005, 0.8)

Eil51	432.9(1.62%)	433.2(1.69%)	440.6(3.43%)	437.6(2.72%)
KroA100	21846.9(2.65%)	21732.3(2.12%)	21926.9(3.03%)	21880.1(2.81%)
D198	16944.5(7.38%)	16872.5(6.92%)	16630.5(5.39%)	16934.5(7.32%)
Lin318	43007.4(2.33%)	43239.7(2.88%)	43843.5(4.32%)	42994.9(2.30%)
Instance	(0.005, 0.1)	(0.005, 0.4)	(0.005, 0.5)	(0.005, 0.8)
Eil51	438.5(2.93%)	441.7(3.69%)	429.3(0.77%)	426.9(0.21%)
KroA100	21997.5(3.36%)	21799.5(2.43%)	21803.6(2.45%)	21808.9(2.48%)
D198	16795.5(6.43%)	16666.5(5.62%)	16120.5(2.16%)	15839.5(0.0%)
Lin318	42901.5(2.08%)	42798.4(1.83%)	42321.2(0.70%)	42029.0(0.0%)
Instance	(0.05, 0.1)	(0.05, 0.4)	(0.05, 0.5)	(0.05, 0.8)
Eil51	429.6(0.85%)	438.1(2.84%)	439.2(3.10%)	441.4(3.62%)
KroA100	21285.0(0.0%)	21983.3(3.30%)	21983.3(3.30%)	21996.7(3.36%)
D198	16064.5(1.80%)	17091.5(8.31%)	17091.5(8.31%)	17326.5(9.80%)
Lin318	43108.9(2.57%)	44089.4(4.90%)	44089.4(4.90%)	44331.8(5.48%)

It can be seen from Table 1 that three of the four instances obtained the best computing performance in the combination of (0.005, 0.8), and the instance KroA100 obtained best computing performance in the combination of (0.05, 0.1). On the whole, regardless of the combination type, the performance of the algorithm are quite good, the gap between the worst results of the calculation and the optimal solution is relatively small. Compared to MMAS, the solution quality of the algorithm is not ideal, but to the general AS, GA and PSO algorithm, the solution quality has been greatly improved and showed certain stability.

3.2. The local move strategy and global move strategy

Local move strategy utilizes the optimal solution in each particle iteration trajectory as the attractive center and it is initially set to the optimal solution in the population. ITO algorithm using this strategy is abbreviated as ITO^{les}. Global move strategy employ the optimal solution in the population as the attractive center, ITO algorithm using this strategy is abbreviated as ITO^{ges}. In a certain sense, the attractor of each particle with the local move strategy may not be the same; hence the diversity of the algorithm can be guaranteed better than the global move strategy.

3.3. The measurement and analysis of the diversity of the population

We mainly study the diversity of the ITO algorithm retention mechanism, that is, we will test that whether the algorithm keeps diverse in the whole iterative process or not. Here we use two diversity measures: one is based on connection matrix, another is based on information entropy.

As can be seen from Table 2, the population diversity of each algorithm is larger when the algorithm is in the initialization after 100 generations or when the algorithm terminates, regardless of using connection matrix or information entropy, besides, the common trend is as the iteration depth increasing, the diversity of the population became lower, but reduced marginally. At the same time, we can observe the four computing instance using ITO^{les} algorithm possess higher diversity than that using ITO^{ges} algorithm.

Table 2. The statistical results of diversity difference

Instance	Start(ITO ^{les})	100th(ITO ^{les})	terminate(ITO ^{les})	(ITO ^{ges})	100th(ITO ^{ges})	Terminate(ITO ^{ges})
Criterion	D(P)	H(P)	D(P)	D(P)	H(P)	D(P)
Eil51	0.559	0.354	0.531	0.351	0.515	0.308
KroA100	0.620	0.210	0.601	0.202	0.593	0.197
D198	0.691	0.299	0.645	0.276	0.630	0.266
Lin318	0.663	0.301	0.639	0.299	0.622	0.298

4. The experimental results and analysis

This section we study the performance of ITO algorithms for TSP, mainly from two aspects to design the experiment, one is comparative experiment using different algorithms, and the other is to increase the performance of the algorithm using local search technique.

4.1. The comparative experiment using different algorithms

As a meta-heuristic algorithm, the heuristic algorithms we compare for TSP problem are as follows: MMAS, ACS, AS, DPX-GA and PSO algorithm [5-15]. Among these algorithms, the experimental data of MMAS ACO, AS comes from [16], the experimental data of DPX-GA come from [17], the experimental data of PSO algorithm and the algorithm proposed in this paper is obtained by our programming and the parameter settings of the population size and the instance size should match: local learning rate $c1 = 0.8$, global learning rate $c2 = 0.6$, use log function to convert the continuous variables between $[0, 1]$, then reverse the gene locus in accordance with the probability; Shutdown condition is for 20 consecutive generation algorithm does not improve. The parameter settings of ITO algorithm: the number of particles matches the size of the instance and shutdown condition is that for 20 consecutive generation algorithm does not improve, the method for calculating the particle radius is the linear transformation and $P_{min} = 0.005$, $P_{max} = 0.8$.

Table 3 is result of different algorithms. We can see from the table, ITO^{ges} algorithm outperforms the others as for the calculation results, but the performance of these algorithms is not significantly different in all calculations instance, PSO algorithm has the worst performance, followed by the ACS and AS; the algorithm ITO^{ges} has the best performance in these comparison algorithms. As a generic algorithm, the computational performance of the ITO^{ges} algorithm is competitive and the algorithm improvements are worthy of further study.

Table 3. The The average results for different algorithms, each run 30 times independently, The number in bold face shows the best results.

Instance	MMAS	ACS	AS	DPX-GA	PSO	ITO ^{ges}
Eil51	427.6	428.1	437.3	428.5	498.2	426.9
kroA100	21320.3	21420.0	22471.4	21285	21993.5	21285
D198	15972.5	16054.0	16702.1	15839.5	16049.7	15939.5
Lin318	42029.0	0	0	42605.3	49385.1	42029.0
Eil51	427.6	428.1	437.3	428.5	498.2	426.9

5. Conclusion

In this paper we studied how ITO algorithm is used in solving TSP problem and discussed the design and parameter settings of ITO algorithm, including the move operator design, the wave operator design, the calculation of the particle radius, the method to select the attractive center and so on. In this paper, we studied the performance of the algorithm and the diversity of the population information in the case of different parameter values, and preliminary experimental results show that the ITO algorithm for solving the symmetric TSP problem with a certain degree of competitiveness. In this paper, we can see that ITO algorithm performs better than several other of the set when the wave operator mapping range is set to (0.005, 0.8).

References

- [1] Dong W Y. Simulation Optimization Based on the Hypothesis Testing and ITO Process. In: Third International Conference on Natural Computation. Haikou, China: IEEE, 2007. 1210-1221
- [2] Dong W Y. Time Series Modeling Based on ITO Algorithm. In: Third International Conference on Natural Computation. Haikou, China: IEEE, 2007. 398-402
- [3] Dong W Y. The Simulation Optimization Algorithm Based on the Ito Process. In: The 2nd International Conference on Intelligent Computing. Qingdao, China: IEEE, 2007. 563-573
- [4] Dong W Y. The Multi-Objective ITO Algorithms. In: The 2nd International Symposium on Intelligence Computation and Applications (ISICA 2007). Qingdao, China: IEEE, 2007. 21-

23

- [5] Reinelt G. TSPLIB-A traveling salesman problem library. *ORSA Journal on Computing*, 1991, 3 (4), 376-384.
- [6] Shi X H, Liang Y C, Lee H P et al. Particle swarm optimization-based algorithms for TSP and generalized TSP, *Information Processing Letters*, 2007, 103 (1), 169-176.
- [7] Huang L, Zhou C G and Wang K P. Hybrid ant colony algorithm for traveling salesman problem, *Progress in Natural Science*, 2003, 4(13), 295-299
- [8] Ben-nian, Wang et al. RMGA: A reinforcement learning based genetic algorithm. *Acta Electronica Sinica*, 2006, 34(5), 856-860, 866.
- [9] Dorigo M and Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1):53-66
- [10] Yagiura M and Ibaraki T. On Metaheuristic Algorithms for combinatorial Optimization Problems, *System and Computers in Japan*, 2001, 32(3), 33-55
- [11] Mak K T and Moton A J. A modified Lin-Kernighan traveling salesman heuristic. *Operations Res Lett*, 1993, 12, 127-132
- [12] Chang P C, Huang W H and Ting C J. Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems, *Expert systems with Applications*, 2010, 37(2) 1863-1878
- [13] Hendtlass T. Preserving Diversity in Particle Swarm Optimization, in: *Lecture Notes in Computer Science*, vol. 2718, Springer, 2003, 4104-4108
- [14] Chang P C, Chen S S, and Fan C Y. Mining gene structures to inject artificial chromosomes for genetic algorithm in single machine scheduling problems. *Applied Soft Computing Journal*, 2008, 8(1), 767-777
- [15] Munakata T and Nakamura Y. Temperature control for simulated annealing, *Physical Review E*, 2001, 64(2) 046-127
- [16] Stützle T, Hoos H H. MAX-MIN Ant System, *Future generation computer systems*. 2000, Vol.16, No.8, 231-240
- [17] White C M and Yen G G. A hybrid evolutionary algorithm for traveling salesman problem, In: *Congress on Evolutionary Computation (CEC2004)*, Chichogo, American. 2004, 176-180.