

A novel GRASP based on mixed k-opt method for the Traveling Salesman Problem

Zheng Ming

College of information and electronic engineering,
Wuzhou University,
Wuzhou, Guangxi,
e-mail: 370505375@qq.com

Guo Hui

College of information and electronic engineering,
Wuzhou University,
Wuzhou, Guangxi,
e-mail: 13066724@qq.com

He Jie

College of information and electronic engineering,
Wuzhou University,
Wuzhou, Guangxi,
e-mail: 64875130@qq.com

Liu Guixia*

College of computer science and technology,
Jilin University,
Changchun, Jilin, China,
email: liugx@jlu.edu.cn
* Corresponding Author

Abstract—Objective: A novel Greedy Randomized Adaptive Search Procedure was proposed in this paper to resolve the traveling salesman problem, which is proven to be NP-complete in most cases.

Methods: The proposed novel algorithm has two phases. In the first phase the novel algorithm finds an initial solution of the problem with a proposed merge feature greedy randomized method. In the second phase the expanded neighborhood adaptive search procedure was proposed to find the TSP solution.

Results: The proposed algorithm was tested on numerous benchmark problems from TSPLIB. The algorithm is compared with other two algorithms and the results showed that the results of the proposed algorithm are always the best. The results were very satisfactory.

Conclusion: For the majority of the instances the results were equal to the best known solution. The algorithm is suitable for the TSP. This kind of novel algorithm can be used for many aspects of object, especially for logistical problem.

Keywords—Randomized Adaptive Search Procedure; traveling salesman problem; optimal algorithm; Greedy Randomized Adaptive Search Procedure; NP-complete problem

I. INTRODUCTION

The traveling salesman problem (TSP)[1] is one of the well-known and extensively studied NP-hard combinatorial optimization problem. Consider a salesman who has to visit n cities. Given a finite set of cities together with the cost of travel between n cities, find a tour of lowest cost which visits each city just once and returns to the city of origin. Cost can be distance, time, money, energy, etc. We speak of a symmetric TSP[2], if for all pairs i, j the distance d_{ij} is equal to the distance d_{ji} . We speak of Euclidean TSP[3], if the cities can be represented as points in the plain such that d_{ij} is the Euclidean distance between point i and point j . Graph theory defines the problem as finding the Hamiltonian cycle[4] with the least

weight for a given complete weighted graph. In this paper, Euclidean symmetric TSP (EsTSP) with Hamiltonian cycle (HC) is considered.

TSP is applied widely in engineering and business such as logistics. It has been used in designing hardware devices, communications, and etc. In addition, some industrial problems such as machine scheduling and cellular manufacturing can be formulated as a TSP[5].

It is natural to define the EsTSP in terms of a complete undirected weighted graph $UWG = (V, E)$ with $V = \{v_1, v_2, v_3, \dots, v_n\}$ representing nodes collection in UWG, and $E = \{e_{ij}\}_{n \times n}$ representing the edges collection between nodes in UWG. $n = |V|$, represented the number of nodes in UWG. the distance between node i and node j can be defined as w_{ij} . In EsTSP, $w_{ij} = w_{ji}$. EsTSP obeys in particular the triangle inequality $w_{ij} \leq w_{ik} + w_{kj}$ for all i, j, k . The main difficulty in finding the optimal HC (OHC) is that the number of tours increases in proportion to the factorial of the number of nodes. If there are n nodes, the number of tours is $(n-1)!/2$ [6]. It's too huge to calculate OHC when n is big enough.

Algorithms for solving the TSP may be divided into two classes, exact algorithms and heuristic algorithms. The exact algorithms, such as branch and cut algorithms[7], are suitable for less nodes TSP. when more nodes TSP must be calculated, heuristic optimal algorithms, which were the motioned in this paper, were used to find OHC. These algorithms include Genetic Algorithm (GA)[8], Particle Swarm Optimization (PSO)[9], ant colony optimization (ACO)[10], Gravitation Field Algorithm (GFA)[11-13] and etc.

Greedy Randomized Adaptive Search Procedure (GRASP)[14] is two phase optimal algorithm which can be used to combinatorial optimization. Every iteration consists of two phases. One is construction phase, the other is the optimal search phase. In the first phase, greedy random method provides feasible solutions incorporating both greedy and random characteristics. A start node can be selected firstly, The choice of the next node to be added is determined by ordering all nodes in a restricted

candidate list (RCL) with respect to a greedy function. Greedy method will be used to select the next node to form a HC in RCL. The random method in the first phase is also used to choose one of the best candidate in RCL but not necessary the top candidate node. In the second phase, an optimal search algorithm is used to find OHC from HCs. GA, PSO, GFA can be used for this part.

A novel expended GRASP (eGRASP) was proposed to solve EsTSP described above in this paper. eGRASP was divided into two phases. One is the construction part, which was proposed in this paper. This method is a kind of greedy random method with fusion characteristic. This method is based on traditional Kruskal's algorithm, but the restriction for the cycle path part is removed. Some local HCs (LHCs) can be calculated by this method, but these LHCs will be merged into a whole HC with fusion characteristic. The second part of the algorithm is the optimal search part, which was based 2-opt and 3-opt[15] methods. These two method was used one by one to search the optimal results.

The proposed eGRASP is compared with GRASP, 2-opt and 3-opt with TSPLIB database. And the efficiency of eGRASP is proved to solve TSP.

II. METHOD

In the previous section, the traditional GRASP was produced. And the framework of the proposed eGRASP in this paper was presented. In this section, the eGRASP algorithm will be described in detail step by step. The addition and the edition of edges is a very important part in this algorithm. The selection of the data structure in eGRASP will be the special part in this paper and based on the operation of edges. The edge collection which contains two column and n rows is the data structure in the proposed algorithm. The first column is the start point in the corresponding edge, the other column is the end point of the edge. A row presents a weight value of an edge in the graph. n represents the number of nodes in the graph.

The eGRASP can be divided into two parts, construction phase and search phase. The details will be described as below.

A. Construction phase of eGRASP

Construction phase consists of two parts. In stage one, the RCL will be created. And the best candidate edge for the tour inclusion in the current iterator will be selected from the RCL. Initially, all edges which were possible belonged to the tour will be created by ordering these edges from the smallest to the largest. From this list, the first D edges were selected to form the RCL. D is the dimension of the RCL, which means D smallest edges were in the RCL. In the RCL, the next edge will be selected randomly from RCL in t iterator. Then the RCL will be readjusted by replaying the edge included in the RCL by another edge which is not included in the RCL.

The stage two is the kernel part in the construction phase. In this stage, the modified Kruskal's algorithm was used. Kruskal's algorithm, which is a total greedy algorithm, is one of the most important construction algorithm for searching OHC. Different from the traditional Kruskal's algorithm, the nearest merger method was used to search for more accurate results. For traditional algorithm, prevention for HC cycle path in one

iteration must be done when a new edge was added to the tour. For the proposed modified Kruskal's algorithm, this HC cycle prevention examination is not necessary. So many sub HCs (sHCs) will be created which consists of sequence T_1, \dots, T_n such that each T_i is a set of $n - i + 1$ disjoint sub-tours. All sHCs were covered by all the nodes. In this part, one tour is T_1 , the other tour is T_2 . The total procedure is described as below:

1. If both T_1 and T_2 consist of one single node, then these two tours are merged in one tour.

2. If T_1 consists of one single node k , T_2 contains at least two nodes, the merged tour is then the TOUR (T_2, k). For every edge e_{ij} in T_2 , which contains i and j as end nodes, if $d_{ik} + d_{kj} - d_{ij}$ is the minimum in all possible edges, the merged TOUR will be changed by deleting e_{ij} and adding e_{ik} and e_{jk} . If T_2 consists of one single node k' , T_1 contains at least two nodes, then the merged tour is the TOUR(T_1, k'). For every edge e_{ij} in T_1 , if $d_{ik} + d_{kj} - d_{ij}$ is the minimum in all possible edges, the merged TOUR will be changed by deleting e_{ij} and adding $e_{ik'}$ and $e_{jk'}$.

3. If both at least two nodes are belonged to the unmerged tour T_1 and T_2 , i and j are nodes in T_1 , k and l are nodes in T_2 . For every nodes pair e_{ij} and e_{kl} , If $d_{ik} + d_{jl} - d_{ij} - d_{kl}$ is minimized in all possible cases, the merged tour is then changed by deleting e_{ij} and e_{kl} and replacing them by e_{ik} and e_{jl} .

The three kinds of merged procedure were showed as Fig. 1. In Fig. 1, (a) represents the case of merging T_1 and T_2 which both contained one single node described above. (b) represents the case of merging T_1 , which contains at least two nodes with a sub tour, and T_2 , which contains one single node. The right tour is T_2 , which contains only node k , and the left tour is T_1 . (c) represents the case of merging T_1 and T_2 which both contained non-single node sub tour. The left tour is T_1 , and the right node is T_2 in this figure.

After every sHC was added to the merged total HC, the construction phase ended. One initial OHC which is not the final best results can be obtained in this part.

B. Search phase of eGRASP

When the initial OHC is obtained, the search phase is used for more accurate result. Two different local search algorithms, namely 2-opt and 3-opt method which are belonged to k-opt, were used to search the final result. Through the computation of two kinds of k-opt algorithms, k edges were deleted and k other edges were added. Given a feasible initial OHC, the proposed search phase of eGRASP examines every possible solution near to the constructed OHC and finds out another lower cost neighboring solution. Using the mixed k-opt method, lower cost solutions can be calculated by deleting k edges from the current tour and reconnected the corresponding unconnected nodes using k new edges. In this paper, a mixed neighborhood searching method, which includes 2-opt and 3-opt, are used.

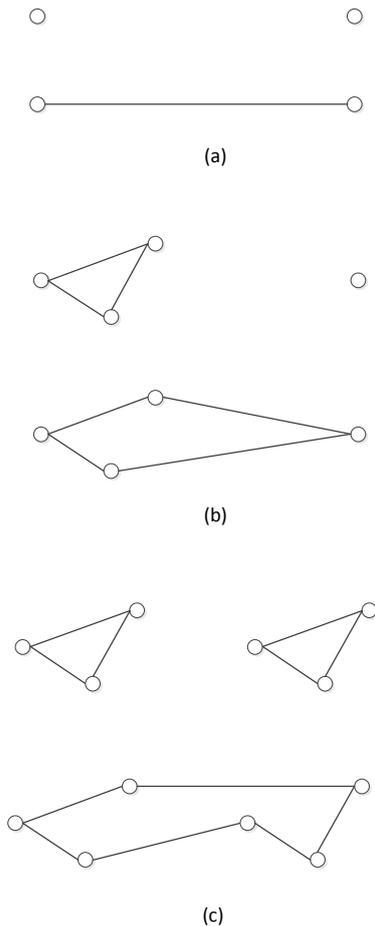


Figure 1. Three situations of merging two sub-HCs to a larger HC.

Firstly, 2-opt procedure was used. In this method, 2 edges were deleted and 2 new edges were reconnected between these 4 nodes. Note that there is only one way to reconnect the paths. In this paper, traditional 2-opt are not used. The restricted algorithm, which is more complex, is used for EsTSP. The deleted two edge are not selected randomly like traditional 2-opt, more conditions are used in the algorithm.

Initially, all edges in the initial OHC are ordered from largest to smallest costs, using a list structure. And the edge on the top list is selected as e_{ij} . The nodes i and j are the end nodes. Let nodes k and l are the end nodes for any other candidate edge e_{kl} in the tour T . The two candidate edges for deletion in the tour are expected to obey the equation:

$$T' = T \setminus \{e_{ij}, e_{kl}\} \cup e_{il} \cup e_{jk} \quad (1)$$

In this equation, T is the initial tour solution. T' is the searched OHC solution. ' \setminus ' represents the deletion operation, and ' \cup ' represents the inclusion operation. The edge e_{ij} represents the largest edge in the ordered edges. e_{kl} represents the other candidate edge in the tour. The proposed algorithm will examine every edge in the tour to find out the suitable e_{kl} which can make length of T' minimum in the equation. The edges e_{il} and e_{jk} represent two new edges for inclusion. Compare T with T' in one iteration, if the total distance of T' is longer, the operation of the equation is canceled. Otherwise, T' replace T in this iteration. The 2-opt method terminates when a certain number of edges is deleted. For most case of the algorithm,

the number of iteration is not large, which is equal to n (the number of nodes) at most. One instance of 2-opt is shown as below:

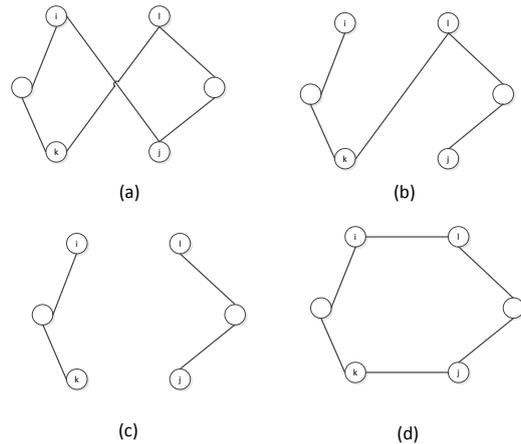


Figure 2. One instance of one improved 2-opt

In Fig .2. (a) represents the initial feasible OHC solution of the algorithm. (b) represents the tour in which e_{ij} is deleted. (c) represents the tour in which e_{kl} is deleted. Compare T and T' , the total distance of T' is shorter than T in this iteration. (d) represents the tour in which e_{il} and e_{kj} are reconnected.

After the improved 2-opt method in our work was used as above, a similar improved 3-opt was used. Not like randomly selected method in traditional 3-opt method, the edges for selection are more carious. When 3 edges are deleted from the current tour T , T will break into three parts instead of two parts like 2-opt. Because of the larger neighborhood than 2-opt, 3-opt method have more results which are more reasonable and complex. There are $C(n,3)$ methods to remove the three edges. Even when the three edges are fixed, there are also eight ways to reconnect the corresponding nodes to reform the tour T . If local optimization of 2-opt is achieved, the 3-opt algorithm has now the possibility to escape from it and find out a better OHC.

Firstly, all the edges are ordered from largest to smallest. The largest length of edge e_{ij} will be selected. Then the algorithm will examine all possible candidate edges in the tour T to and find out other two candidate edges e_{kl} and e_{mn} which make the value of the total new tour's length be minimum in Eq. (2) as below.

$$T' = T \setminus \{e_{ij}, e_{kl}, e_{mn}\} \cup e_{im} \cup e_{ln} \cup e_{jk}$$

In Eq. (2), e_{ij} is the edge which has the longest length in the tour T . The edges e_{kl} and e_{mn} are the candidate edges. The edges e_{im} , e_{ln} and e_{jk} are the new edges, which reconnect these six nodes i , m , l , n , k , and j , in the tour. ' \setminus ' represents the deletion operation. ' \cup ' represents the inclusion operation. Compare T with T' in the lengths of the two tours. If T' is shorter, the operation will be remained. Otherwise, this iteration of 3-opt will be canceled. The number of iteration in 3-opt will be not large enough, which is n at most. One instance of 3-opt is shown as below:

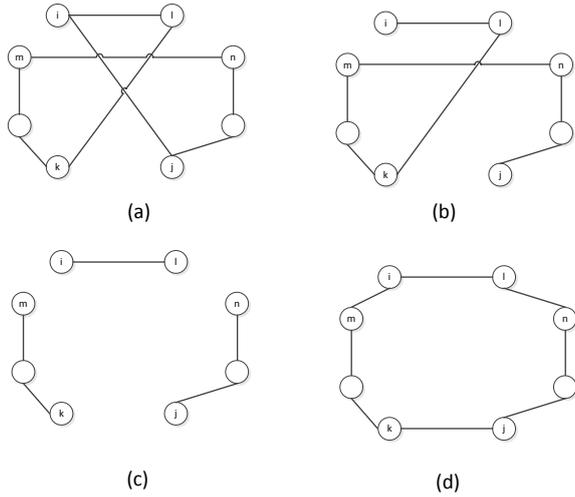


Figure 3. Instance of one improved 3-opt

C. The procedure of the eGRASP

The procedure of the eGRASP can be described as below:

1. An initial solution of eGRASP can be established according to the greedy random algorithm. Every edge must be ordered by their lengths to form the RCL. Then many local HCs will be constructed from the RCL. Finally, initial OHC can be merged by these HCs.
2. find out a more accurate OHC according to 2-opt method.
3. the final result of the OHC will be found out according to 3-opt.
4. If the end condition of the proposed algorithm reached, the eGRASP ended. Otherwise, the algorithm goes to 1.

III. EXPERIMENT

To test the efficiency of the proposed algorithm eGRASP, 20 different two-dimensions EsTSP instances from TSPLIB database were tested. The number of nodes in these instances is from 105 to 783. Every instance name in the experiments can be divided into two parts, the first part is the instance name, the other part is the number of nodes. For example, Eil in Eil51 is the instance name. 51 in Eil51 is the number of nodes in the TSP instance. The proposed algorithm is compared with the other three kinds of algorithms to test the algorithm's efficiency. These three kinds of algorithms are traditional GRASP, 2-opt and 3-opt. 20 different runs for the four algorithms were used to test these instances data. The optimal values of all instances were calculated for their comparison. The search phase of traditional GRASP were 2-opt method. The length of RCL for both traditional GRASP and eGRASP is 50. The edge selection method of both 2-opt and 3-opt is random.

The results of the four algorithms for 20 instances are shown as Table 1 and Table 2 as below:

TABLE I THE BEST LENGTH OF FOUR ALGORITHMS WITH 20 TSPLIB INSTANCES

	Best length			
	eGRASP	GRASP	2-opt	3-opt
Lin105	14379	15764	135450	14379
Pr107	44303	49836	209996	135124
Pr124	59030	65369	211327	126915
Pr136	96772	102003	1133200	690952
Pr144	58537	61162	280392	81366
Pr152	73682	84183	289570	209994
Rat195	2331	2546	17330	9315
D198	15788	16383	76533	112354
Pr226	80414	94360	1138829	218604
Gil262	2385	2616	27632	9678
Pr264	49135	55517	264838	345910
Pr299	48235	54251	552269	258786
Rd400	15385	16618	91839	67542
Pr439	107401	123147	806272	494270
Pcb442	50941	56921	494578	203620
D493	35253	38196	362971	151209
Rat575	6863	7500	60483	36981
P654	34707	38908	550477	56122
D657	49531	55180	419176	221082
Rat783	8897	9890	88676	45967

TABLE II THE RE VALUE OF FOUR ALGORITHMS WITH 20 TSPLIB INSTANCES

	RE value			
	eGRASP	GRASP	2-opt	3-opt
Lin105	0.00	9.63	8.42	0.00
Pr107	0.00	12.49	3.74	2.05
Pr124	0.00	10.74	2.58	1.15
Pr136	0.00	5.41	10.71	6.14
Pr144	0.00	4.48	3.79	0.39
Pr152	0.00	14.25	2.93	1.85
Rat195	0.34	9.60	6.46	3.01
D198	0.05	3.82	3.85	6.12
Pr226	0.05	17.41	13.17	1.72
Gil262	0.29	10.01	10.62	3.07
Pr264	0.00	12.99	4.39	6.04
Pr299	0.09	12.57	10.46	4.37
Rd400	0.68	8.75	5.01	3.42
Pr439	0.17	14.86	6.52	3.61
Pcb442	0.33	12.10	8.74	3.01
D493	0.71	9.13	9.37	3.32
Rat575	1.32	10.73	7.93	4.46
P654	0.18	12.31	14.89	0.62
D657	1.26	12.81	7.57	3.52
Rat783	0.00	9.63	8.42	0.00

The best lengths and RE values of all instances for the four algorithms were shown in Table 1 and Table 2 respectively. In Table 1 and Table 2, eGRASP represents the proposed algorithm in this paper, GRASP represents the traditional GRASP algorithm. The RE value represents the relative error value which is shown as Eq. (3) as below:

$$RE = (L_{best} - L_{opt}) / L_{opt} \times 100\% \quad (3)$$

In Eq. (3), L_{best} represents the best length obtained by the four algorithms which is the value in Table 1. The L_{opt} represents the optimal value obtained by the TSPLIB database. The corresponding L_{opt} can be downloaded from the database website as below: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>. The best length is the minimum result in the 20 different runs. The efficiency of the four different algorithms can be seen

from Table 2. The RE values of eGRASP were the minimum in the four different algorithms results in most cases. For the instances with all kinds of number of nodes, the RE value is smaller than 1.5%. The corresponding RE value were smaller than the results from GRASP, 2-opt or 3-opt. 3-opt is better than the other two algorithms. The worst algorithm is traditional GRASP.

The accuracy of the proposed eGRASP were tested above. Then the relative time consumption of every phase in the proposed algorithms were tested and compared. 20 different runs were used to test the eGRASP time consumption. And the best direct value obtained by the eGRASP can be seen from Table 4. The two results were shown as Table 3 and Table 4 as below.

TABLE III THE TIME CONSUMPTION PERCENT OF THE THREE ALGORITHMS WITH 20 TSPLIB INSTANCES

	time consumption percent(%)		
	Kruskal	2-opt	3-opt
Lin105	4.8	1.8	93.4
Pr107	4.9	1.6	93.5
Pr124	4.4	1.5	94.1
Pr136	3.6	1.4	95
Pr144	3.2	1.4	95.4
Pr152	3.7	1.2	95.1
Rat195	2.5	1.2	96.3
D198	3.6	1.1	95.3
Pr226	2.5	1.1	96.4
Gil262	1.8	0.9	97.3
Pr264	2.5	0.9	96.6
Pr299	1.8	1	97.2
Rd400	2.1	0.6	97.3
Pr439	1.2	0.6	98.2
Pcb442	1.4	0.5	98.1
D493	1.2	0.4	98.4
Rat575	1	0.4	98.6
P654	0.7	0.3	99
D657	1.26	0.4	99.3
Rat783	0.00	0.3	98.9

TABLE IV THE DIRECT VALUES OF THE THREE ALGORITHMS WITH 20 TSPLIB INSTANCES

	direct values		
	Kruskal	2-opt	3-opt
Lin105	21112	15764	14379
Pr107	56587	49836	44303
Pr124	84078	65369	59030
Pr136	143527	102003	96772
Pr144	87281	61162	58537
Pr152	97398	84183	73682
Rat195	3498	2546	2331
D198	19700	16383	15788
Pr226	112729	94360	80414
Gil262	3308	2616	2385
Pr264	71794	55517	49135
Pr299	68071	54251	48235
Rd400	21378	16618	15385
Pr439	153667	123147	107401
Pcb442	74886	56921	50941
D493	46089	38196	35253
Rat575	9587	7500	6863
P654	45809	38908	34707
D657	67962	55180	49531
Rat783	12623	9890	8897

The time consumption can be seen from Table 3. The time consumption percent of the 3-opt procedure were greater than 95% except the first 3 instances. If the 3-opt method were not used in the proposed algorithm eGRASP, the results will be not accuracy enough, but running time of the algorithm will be shorter than the algorithm 2-opt used only. So the mixed method was used in the search phase in this paper.

IV. CONCLUSION

A novel algorithm named eGRASP was proposed in this paper for the solution of the EsTSP. The new algorithm is one modification of the well know Greedy Random Adaptive Search Procedure. The computational results obtained using eGRASP were in most cases equal to the optimal value downloaded from the Heidelberg University TSPLIB database. As an example, the result of Lin105 obtained from eGRASP is equal to the value in TSPLIB. The improvement of the proposed algorithm were in the quality, but the running time is longer.

The improvement is both in the construction phase and the search phase in eGRASP. In the construction phase, the Kruskal's algorithm was modified. And many small local cycles can be obtained and used to reform the whole cycle TSP path. In the search phase, two traditional search algorithm 2-opt and 3-opt were mixed to find out the more accuracy results for EsTSP.

20 instances downloaded from the TSPLIB database were used to prove the proposed algorithm's efficiency of the eGRASP. Most of the best results obtained were from eGRASP. The relative time consumptions are summarized in Table 3. And the most time are consumed in the 3-opt method for more accuracy results.

ACKNOWLEDGMENT

The National Natural Science Foundation of China (No. 61502343, No. 61373051, No. 61175023 and No. 61402423), the Guangxi Natural Science Foundation (No.2015GXNSFBA139262 and 2015GXNSFAA139295), the Science Research Funds for the Guangxi Universities (No. KY2015ZD122), Science and Technology Development Program of Jilin Province (No. 20140204004GX), the Science Research Funds for the Wuzhou University (2014A002) and Guangxi Colleges and Universities Key Laboratory of Professional Software Technology, Wuzhou University.

REFERENCES

- [1] Y.Q. Zhou, Q.F. Luo, H. Chen, A.P. He and J.Z. Wu, "A discrete invasive weed optimization algorithm for solving traveling salesman problem," *Neurocomputing*, vol. 151, 2015, pp. 1227-1236.
- [2] B. Escoffier and J. Monnot, "A better differential approximation ratio for symmetric TSP," *Theoretical Computer Science*, vol. 396, 2008, pp. 63-70.
- [3] S. Hougardy, "On the integrality ratio of the subtour LP for Euclidean TSP," *Operations Research Letters*, vol. 42, 2014, pp. 495-499.
- [4] Y.Q. Xiong, "Limit cycle bifurcations by perturbing piecewise smooth Hamiltonian systems with multiple parameters," *Journal of Mathematical Analysis and Applications*, vol. 421, 2015, pp. 260-275.
- [5] L. Koch, "A new window into the genetics of complex diseases," *Nature Reviews Genetics*, vol. 15, 2014, pp.644-645.

- [6] J. Abrahamson and A. Shokoufandeh, "Euclidean TSP on two polygons," *Theoretical Computer Science*, vol. 411, 2010, pp. 1104-1114.
- [7] I. Rodriguez-Martin, J.J. Salazar-Gonzalez and H. Yaman, "A branch-and-cut algorithm for the hub location and routing problem," *Computers & Operations Research*, vol. 50, 2014, pp. 161-174.
- M. Li, Z. Cao, X. Li and H.W. Chen, "A Novel Informative SNPs Selection Method Based on Genetic Algorithm," *Journal of Computational and Theoretical Nanoscience*, vol. 11, 2014, pp. 2109-2114.
- W.H. Lim and N.A.M. Isa, "Bidirectional teaching and peer-learning particle swarm optimization," *Information Sciences*, vol. 280, 2014, pp. 111-134.
- F. Valdez and I. Chaparro, "Ant Colony Optimization for solving the TSP symmetric with parallel processing," *Proceedings of the 2013 Joint Ifsa World Congress and Nafips Annual Meeting (Ifsa/Nafips)*, 2013, pp. 1192-1196.
- M. Zheng, G.X. Liu, C.G. Zhou, Y.C. Liang and Y. Wang, "Gravitation field algorithm and its application in gene cluster," *Algorithms for Molecular Biology* 5, 2010.
- M. Zheng, Y. Sun, G.X. Liu, Zhou, Y. Zhou and C.G. Zhou, "Improved Gravitation Field Algorithm and Its Application in Hierarchical Clustering," *Plos One*, vol. 7, 2012.
- M. Zheng, J.N. Wu, Y.X. Huang, G.X. Liu and Y. Zhou and C.G. Zhou, "Inferring Gene Regulatory Networks by Singular Value Decomposition and Gravitation Field Algorithm," *Plos One*, vol. 7, 2012.
- R. Marti, V. Campos, M.G.C. Resende and A. Duarte, "Multiobjective GRASP with Path Relinking," *European Journal of Operational Research*, 2015, vol. 240, pp.54-71.
- L. Muyltermans, P. Beullens, D. Cattrysse, and D. Van Oudheusden, "Exploring variants of 2-opt and 3-opt for the general routing problem," *Operations Research*, vol. 53, 2005, pp.982-995.