

An Improved Simulated Annealing Algorithm for Travelling Salesman Problem

<http://dx.doi.org/10.3991/ijoe.v9i4.2822>

Y. Wang, D. Tian and Y.H. Li

North China Electric Power University, Beijing, China

Abstract—Traveling salesman problem (TSP) is one well-known NP-Complete problem. The basic simulated annealing algorithm is improved with the four vertices and three lines inequality to search the optimal Hamiltonian circuit (OHC) or approximate OHC. The four vertices and three lines inequality is taken as the heuristic information to convert the local Hamiltonian paths in the Hamiltonian circuit (HC) into the local optimal Hamiltonian paths and the HC become shorter. The HCs are generated with the basic simulated algorithm first. Then the local Hamiltonian paths are changed into the local optimal Hamiltonian paths with the four vertices and three lines inequality. The algorithm of the improved simulated annealing is designed and tested with twenty TSP instances. The experimental results show that the better approximate OHCs are found than those searched with the basic simulated annealing algorithm under the same preconditions.

Index Terms—Traveling salesman problem, Simulated annealing, Four vertices and three lines inequality.

I. INTRODUCTION

The objective of traveling salesman problem (TSP) is to find the optimal Hamiltonian circuit (OHC) in a tourist map. It has been proven to be NP-Complete. The number of the Hamiltonian circuits (HC) increases in proportion to the factorial of the number of the cities in the map [1] (given the cities are connected by routes). The TSP has been widely studied in the fields of combinatorial mathematics, graph theory and computer science due to its theoretical and practical values. The efficient algorithms are conceived and improved to resolve it within a reasonable computation time.

The algorithms for TSP can be categorized into three types which are the exact algorithms, the approximate algorithms and the intelligent optimization algorithms. With the exact algorithms, the OHC is ensured to find whereas they are not suitable to cope with the large scale of TSP. These algorithms include the traditional search graph algorithms [2], linear programming methods [3] and dynamic programming methods [4]. The experiments illustrated that these exact algorithms are feasible for TSP with less than 1,000 cities [5]. If the TSP scale becomes larger, the computation time is too long or the powerful computers must be employed. The experts have found that the performance of the exact algorithms is hard to improve for big TSP. The approximate algorithms cannot guarantee to find the OHC. However, they play an important role for large scale of TSP due to their polynomial but not exponential time complexity. When the local heuristic information is merged into the exact

algorithms, the exact algorithms turn into the approximate algorithms. The approximate algorithms for TSP include the minimum spanning tree algorithm [6], the subset cover algorithms [7] etc in related to graph. These algorithms are efficient whereas the solutions they found have a big gap from the best solutions. The k -opt ($k=2, 3, 4, 5$) algorithms and the LK or LKH algorithms are the competitive approximate algorithms for TSP [5]. It is reported that these algorithms are robust to tackle the large scale of TSP with thousands of cities, even with more than 3,000,000 cities [8]. The researchers also claimed that the tours quality can not be evaluated because the OHCs of these instances are not known.

In the end of last century, the intelligent optimization algorithms become more and more mature. TSP is taken as one of the best platforms to test their performance. Almost all of the intelligent algorithms, such as the artificial neural network [9], the genetic algorithms [10], the simulated annealing (SA) algorithm [11], the ant colony optimization algorithm [12], the particle swarm optimization algorithm [13] and the consultant-guided search algorithm [14] etc, are applied to TSP. The intelligent optimization algorithms evolve to the best solutions based on the evolutionary rules which are different from the local heuristic rules used by the approximate algorithms. The evolutionary rules are executed to generate the best or approximate solutions under the defined statistical formulae. The approximate optimal solutions are usually found with the intelligent algorithms and they are always being improved. The memetic algorithms are the integration of the intelligent algorithms and the local heuristics, which utilize the advantages of the two kinds of algorithms for the large scale of TSP [4]. The experiments show that they will find the better solutions.

The SA algorithm is one of the efficient methods for the continuous and discrete optimization problems. It is derived from the simulation of the cooling schedule of metals. The cooling process is controlled by a defined function which is convenient to implement. In addition, the SA algorithm has no much demand for the initial solutions. Although it has many merits, it is always criticized for its bad performance and slow convergence when it is applied to the complex TSP. The basic simulated annealing (BSA) is improved by several researchers for TSP. Liu introduced the SA integrated with the Tabu search to detect the better solutions. The temperature is reduced adaptively with a temperature control function [11]. Based on most of the edges in the best circuit linked by neighbor cities, the probabilistic neighborhood model was introduced by Li [15] and merge into the optimization process of the BSA algorithm. The

SA algorithm is also integrated with the ant colony optimization [16] to utilize their advantages together.

To accelerate the convergence of the BSA and obtain the better approximate OHCs, the four vertices and three lines inequality is merged into the optimization process of the BSA. The four point conditions for symmetrical TSP has been summarized by Vladimir [17] under their assumptions. The four vertices and three lines inequality is the extension of one of the four point conditions. It can be taken as the constraint of the local optimal Hamiltonian paths composed of four vertices and three lines. When the Hamiltonian circuits are generated with the BSA, the four vertices and three lines inequality is applied to the local Hamiltonian paths in the HCs. After the local Hamiltonian paths, the better HCs are obtained. The computation process is executed until the OHC or approximate OHC is found. The improved simulated annealing (ISA) algorithm is tested with the TSP instances and compared with the results computed with the BSA. The experiments show that the better approximate OHCs are found with the ISA and the convergence is also accelerated.

II. MATHEMATICAL MODEL OF TSP

The objective of TSP is to find the OHC in the map. The map is generally represented as a weighted graph (WG). The cities and routes correspond to the vertices and edges in the WG, respectively. Given a complete symmetrical WG with n vertices, the number of HCs is $(n-1)!/2$. The length of the OHC is shortest among all of the HCs. For graph G with n vertices, it is represented as $G=\langle V, E \rangle$. Where $V=\langle v_1, v_2, \dots, v_n \rangle$ are the vertices sets and $E=\langle e_{1 \times 2}, e_{1 \times 3}, \dots, e_{(n-1) \times n} \rangle$ are the edges sets. v_i ($1 \leq i \leq n$) is the vertex and $e_{i \times j}$ ($1 \leq i, j \leq n$) is the edge linking the two vertices v_i and v_j . Graph G is represented as the adjacent matrix $A(G)=[a_{ij}]_{n \times n}$ ($1 \leq i, j \leq n$). Where $a_{ij}=1$ if $(v_i, v_j) \in E(G)$ and v_i and v_j are adjacent in the graph G . Otherwise, $a_{ij}=0$. If the edges are assigned with weights $W=[w_{ij}]_{n \times n}$, the graph G becomes one WG. The weight w_{ij} is often taken as distance, cost etc for various kinds of TSP. For the symmetrical TSP, w_{ij} is equal to w_{ji} .

Given a HC with n vertices, it is represented as $HC^{n+1}=(v_1, v_2, v_3, \dots, v_n, v_1)$. The HC includes all of the vertices in the WG once and exactly once unless it is the head and end vertices. The head and end vertices are identical to form the HC. The other vertices between them are the middle vertices. Given $l_{i \times j}$ is the distance between the two adjacent vertices v_i and v_j in the HC, the mathematical model of the Euclidean TSP is given as formula (1).

$$L_{\min} = \min(L(HC)) = \min \left\{ \sum_{i,j=1}^n l_{i \times j} \right\} \quad (1)$$

s. t. $v_i \neq v_j$ and $e_{i \times j} \in E(HC)$

Where $L(HC)$ is the length of the HC, $e_{i \times j}$ ($1 \leq i, j \leq n$) is the edge linking the two adjacent vertices v_i and v_j in the HC. For a symmetrical WG with n vertices, the number of HCs is $(n-1)!/2$. It is a great challenge to find an OHC among all of the HCs.

The HC is composed of the local Hamiltonian paths (LHP) and the OHC includes the local optimal

Hamiltonian paths (LOHP). The LHP or LOHP including i vertices is represented as LHP^i or $LOHP^i=(v_1, v_2, v_3, \dots, v_i)$. For an arbitrary LOHP in the OHC, its two end vertices are determined in general. Its length is the minimum among those of the LHPs with the same vertices in case that their two end vertices are the same. From above analysis, we know that the LHPs in the OHC and most of the LHPs in an approximate OHC must be LOHPs. The four vertices and three lines inequality is used to convert the LHP's into the LOHP's and a HC will become a better HC. The local heuristic is helpful to accelerate the convergence of BSA as well as find the better approximate OHC.

III. THE IMPROVED SIMULATED ANNEALING (ISA)

The ISA algorithm is the combination of the BSA algorithm and the four vertices and three lines inequality. The framework of the ISA for TSP is given in Table I.

TABLE I. THE PROCEDURE OF THE ISA ALGORITHM

Step	The pseudo codes of the ISA algorithm
1	Generate an initial HC at random and compute its length L .
2	Set the initial and end temperature $T:=T_0$ and $T:=T_E$.
3	Set the maximum iterations of the inner loop $N:=N_c$ and iteration index $t:=1$.
4	While (temperature bigger than T_E)
5	While (iterations less than N_c)
6	Generate the next HC' with the previous HC.
7	Compute the length L' of the HC' and the absolute error $\Delta L:=L-L'$.
8	If ($\Delta L > 0$)
9	Replace the HC with the HC' and $t:=t+1$.
10	Else
11	If ($\exp(\Delta L/T) > \text{rand}()$)
12	Apply the four vertices and three lines inequality to generate shorter HC'.
13	Replace the HC with the HC' and $t:=t+1$.
14	Else
15	Discard the HC' and $t:=t+1$.
16	End
17	Update the temperature $T:=kT$.
18	End

The main steps in Table 1 are described as follows. Step 1 is used to compute the initial HC. The numbers representing the n vertices are chosen randomly to form the initial HC, such as $HC=(2,1,4,3,5,2)$ for TSP with 5 cities (vertices). The initializations of ISA are completed before step 4. There are two computation loops in the ISA algorithm. The outer loop is terminated by the appointed end temperature assigned in step 2. The terminal condition of the inner loop is the maximum computation iterations assigned in step 3.

The next HC' is generated at step 6. Firstly, two different cities in the previous HC are selected at random and their positions in the HC are exchanged. Then the next HC' is generated. For example, two cities 1 and 3 in the $HC=(2,1,4,3,5,2)$ are selected and exchanged and a new $HC'=(2,3,4,1,5,2)$ is generated. If the HC' is longer than the HC, the 2-opt method [18] is used to change the HC into a new a HC'. With the method, a LHP in the HC' is

selected at random and the vertices in the LHP are reversed. For example, a LHP=(3,4,1,5) in the HC' is chosen and reversed and the next new HC'=(2,5,1,4,3,2) is produced. The two operations to generate the HC' are executed once in one computation cycle. The new HC' is maintained if it conforms to the conditions at step 8 or step 11. Otherwise, it will be discarded at step 15.

\exp denotes the base of the natural logarithms and the value is assigned as 2.71828. The four vertices and three lines inequality can be applied to the HC' once it is generated. In the ISA algorithm, it is placed at step 12 to change the HC into a shorter HC'. The principle of four vertices and three lines inequality is illustrated in Fig. 1.

Given at least two HC exists in the WG, the two different HCs with n vertices are shown in Fig. 1 (a) and (b). The HC in Fig. 1 (a) is the OHC. One of the LOHP⁴s in the Fig. 1 (a) is LOHP⁴=($v_{i-1}, v_i, v_j, v_{j+1}$) ($2 \leq i \leq n, 1 \leq j \leq n-1$). The LHP⁴=($v_{i-1}, v_j, v_i, v_{j+1}$) in Fig. 1 (b) includes the same four vertices $v_{i-1}, v_j, v_i, v_{j+1}$ and the positions of the two middle vertices are different from those vertices in the Fig. 1 (a). Given the two dashed paths LHPⁿ⁻⁴s are identical except the LOHP⁴ and LHP⁴, the length of the two LHPⁿ⁻⁴s are equal and they are noted as L_{rest} in the Fig. 1 (a) and (b). For the LOHP⁴ and LHP⁴, $l_{i \times j}$ is the length of the edge $e_{i \times j}$. Two pairs of three edges $e_{(i-1) \times i}, e_{i \times j}, e_{j \times (j+1)}$ and $e_{(i-1) \times j}, e_{i \times i}, e_{i \times (j+1)}$ link the four vertices $v_{i-1}, v_i, v_j, v_{j+1}$ in the LOHP⁴ and LHP⁴. The length of the LOHP⁴ is computed as $l_{(i-1) \times i} + l_{i \times j} + l_{j \times (j+1)}$ and the length of the LHP⁴ is computed as $l_{(i-1) \times j} + l_{i \times i} + l_{i \times (j+1)}$. Because the length of the OHC is shorter than or equal to the length of the HC, the four vertices and three lines inequality holds as formula (2). An arbitrary LOHP⁴ in the OHC conforms to the principle of the four vertices and three lines inequality. It is the heuristic information to change the LHP⁴ into the LOHP⁴, and the shorter HCs will be obtained.

$$l_{(i-1) \times i} + l_{i \times j} + l_{j \times (j+1)} \leq l_{(i-1) \times j} + l_{i \times i} + l_{i \times (j+1)} \quad (2)$$

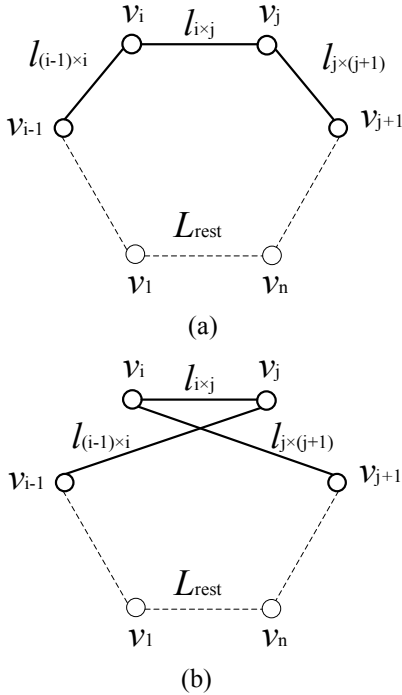


Figure 1. The four vertices and three lines inequality

Because the OHC is composed of the LOHPs, the OHC is also taken as the combinations of the n LOHP⁴s. The four vertices and three lines inequality is used to transform the LHP⁴s into the LOHP⁴s to generate the shorter approximate OHCs. It notes that the function of four vertices and three lines inequality is equal to the 2-opt method [18] when a LOHP is taken as the edge $e_{i \times j}$. Based on the four vertices and three lines inequality, we can understand the intrinsic efficiency of 2-opt method. It always changes a LHP into a LOHP to generate the OHC or approximate OHC. As we know, the number of LOHPs is much less than that of general LHPs. It is unfortunate that the LHP⁴s still exist after 2-shift of vertices and edges. The four vertices and three lines inequality is applied to the LHP⁴s and change them into the LOHP⁴s.

When the inner loop is completed at step 16, the temperature is updated at step 17. k is the factor of temperature decrement, and it is less than 1 to ensure the temperature become smaller and smaller. The assignment of parameter k is important to determine the convergent rate of SA algorithm. The ISA converges to the OHC or approximate OHC once the temperature T is equal to or less than the end temperature T_E .

IV. ILLUSTRATIVE EXAMPLES

The TSP instances are downloaded from the website: www2.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/. These TSP instances are used to illustrate the performance of the ISA. The ISA algorithm is coded with C++ language and runs on the Lenovo computer with processor 2.3GHz and inner memory 2G. The BSA algorithm is also coded and executed on the same computer for comparisons. The parameters of SA algorithm are set as those in Table II. The initial HCs are generated at random. For different TSP instances, the initial temperature and computation cycle of inner loop are set as different values.

TABLE II. THE PARAMETERS OF THE IMPROVED SA ALGORITHM

Initial temperature T_0	End temperature T_E	Maximum cycle of inner loop N_c	Factor of temperature decrement k
Citynum $\times 15$	0.0001	Citynum $\times 500$	0.99

The experiments are tried for the selected twenty TSP instances and the computation results with the two algorithms are shown in Table III and Table IV. The length of the OHC or approximate OHC is computed with real numbers of city coordinates. The integer parts of the length are maintained and the decimal parts are neglected. The sequences of OHC of some TSP instances are given [5] and their length is also computed with the real numbers of the city coordinates. They will be larger than those of the length computed with the integers of the city coordinates. For the TSP with unknown sequence of OHC, the given length of OHC computed with integer coordinates are shown in Table III and Table IV. To show the distinctions of performance between the ISA and BSA, the error of the approximate OHCs to the OHC is defined as $Error = (L(HC) - L(OHC)) / L(OHC) \times 100\%$. Where $L(HC)$ represents the length of the searched approximate OHC and $L(OHC)$ is the length of the given OHC. For the selected TSP instances, the errors of the results are computed and shown in Table III and Table IV.

TABLE III. LENGTH OF THE APPROXIMATE OHCs DETECTED WITH THE ISA ALGORITHM

TSP instances	ISA algorithm Approximate OHC	Error /%	OHC
Berlin52	7544	0	7544
St70	677	0	677
Eil76	544	0	544
Pr76	108159	0	108159
Rat99	1220	0.74	1211
KorA100	21285	0	21285
kroB100	22139	0	22141
KroC100	20750	0	20750
KroD100	21309	0.07	21294
Pr107	44324	0.05	44301
Pr124	59030	0	59030
Ch130	6110	0	6110
KroA150	26851	1.23	26524
KroB150	26263	0.51	26130
Rat195	2359	1.55	2323
KroA200	29548	0.61	29368
KroB200	29461	0.08	29437
TSP225	3933	0.33	3920
Pr264	49135	0	49135
A280	2627	1.58	2586

TABLE IV. LENGTH OF THE APPROXIMATE OHCs DETECTED WITH THE BSA ALGORITHM

TSP instances	BSA algorithm Approximate OHC	Error /%	OHC
Berlin52	7544	0	7544
St70	682	0.74	677
Eil76	544	0	544
Pr76	109187	0.95	108159
Rat99	1221	0.83	1211
KroA100	21344	0.28	21285
KroB100	22509	1.66	22141
KroC100	20852	0.49	20750
KroD100	21442	0.70	21294
Pr107	44575	0.62	44301
Pr124	59181	0.26	59030
Ch130	6184	1.21	6110
KroA150	26920	1.49	26524
KroB150	26513	1.47	26130
Rat195	2376	2.28	2323
KroA200	29754	1.31	29368
KroB200	29693	0.87	29437
TSP225	3950	0.77	3920
Pr264	50028	1.82	49135
A280	2637	1.97	2586

In view of the computation results, it is found that the better approximate OHCs are found with the ISA algorithm. With the BSA algorithm, most of the LHPs in

the approximate OHCs conform to the four vertices and three lines inequality whereas a smaller number of LHPs do not. When these LHPs are adjusted with the four vertices and three lines inequality, the shorter approximate OHCs are obtained. On the other hand, the performance of the ISA is determined by the BSA. The four vertices and three lines inequality is helpful when the LHP⁴s in the HCs violate it. In addition, the executive times of the four vertices and three lines inequality affect the computational results. Once the executive times of four vertices and three lines inequality is changed, the computational results are different. It is also found that the parameters of the SA algorithm play an important role to affect the quality of the approximate OHCs. When the parameters are changed slightly, the different approximate OHCs are generated. The two contents will be researched in the future.

For the Eil76 and A280, the change of the HCs computed with the ISA and BSA algorithms are illustrated in Fig. 2 and Fig.3. The x-coordinate is the computation cycles and the y-coordinate is the length of the approximate HCs. Before the two algorithms converge to the same approximate OHCs, the ISA algorithm always finds the better approximate HCs. It is verified that the convergence of ISA is faster than that of the BSA in the same preconditions. For Eil76, before the 1500th computation cycle, the two algorithms converge to the OHCs. For the A280, the ISA finds a better approximate OHC than the BSA does. The computation results illustrate that the performance of the ISA is better than that of the BSA. The optimization processes of the other TSP instances have the similar change.

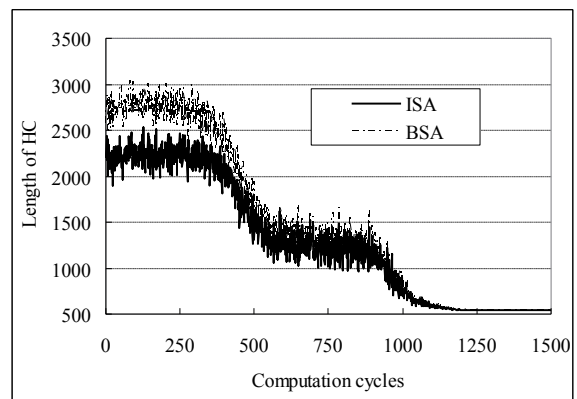


Figure 2. The changes of the HCs generated with the ISA and BSA algorithms for Eil76

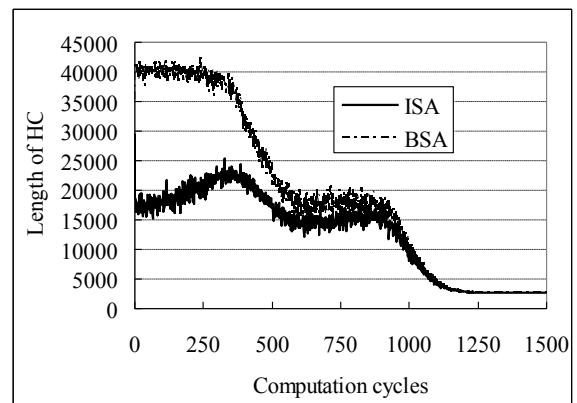


Figure 3. The changes of the HCs generated with the ISA and BSA algorithms for A280

V. CONCLUSIONS

The ISA method is the integration of the BSA algorithm and the four vertices and three lines inequality. It shows better performance than the BSA for TSP. The four vertices and three lines inequality is convenient to implement to change the LHP⁴s into the LOHP⁴s and make the HC become shorter. The performance of the ISA is better than that of the BSA and it nearly always finds the shorter approximate OHCs (or OHC) than the BSA does under the same preconditions. From the experimental results, the ISA algorithm detects the shorter approximate OHCs than those searched with the BSA algorithm in most cases. Of course, the assignment of the parameters is appropriate. The convergence of the ISA algorithm is faster than that of the BSA algorithm.

The authors find that the performance of the ISA is affected by the position and executive times of the four vertices and three lines equality in the algorithm. It will be focused on in the future to improve the performance of ISA. In addition, the assignment of the SA parameters also plays an important role to affect the performance of the ISA. The selection of an appropriate set of parameters will be researched through experiments. In addition, the ISA will be tested with large scale of TSP.

ACKNOWLEDGMENT

The authors acknowledge the funds supported by the Fundamental Research Funds for the Central Universities (Grant No. 12MS48) and NSFC (Grant No.51205129). The work benefits from the facilities of National Key Laboratory of New Energy Power System and the Beijing Key Laboratory of New and Renewable Energy, North China Electric Power University, Beijing, China.

REFERENCES

- [1] C. Seife, "What Are the Limits of Conventional Computing," *Science*, vol.309, pp.96, July 2005. <http://dx.doi.org/10.1126/science.309.5731.96>
- [2] B. W. Douglas, Introduction to graph theory, Section Edition, Beijing: Pearson Education Asia Limited and China Machine Press, 2006, pp.74–81.
- [3] L. Gouveia and S. Voß, "A classification of formulations for the (time-dependent) traveling salesman problem," *European Journal of Operational Research*, vol.83, no.1, pp.69–82, May 1995. [http://dx.doi.org/10.1016/0377-2217\(93\)E0238-S](http://dx.doi.org/10.1016/0377-2217(93)E0238-S)
- [4] B. Bontoux, C. Artigues and D. Feillet, "A memetic algorithm with a large neighborhood crossover operator for the generalized traveling salesman problem," *Computers & Operations Research*, vol.37, no.11, pp.1844–1852, November 2010. <http://dx.doi.org/10.1016/j.cor.2009.05.004>
- [5] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," (www2.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/), December 2012.
- [6] G. Borradaile, P. Klein and C. Mathieu, "An $O(n \log n)$ approximation scheme for Steiner tree in planar graphs," *ACM Transactions on Algorithms*, vol.5, no.3, pp.1–33, July 2009. <http://dx.doi.org/10.1145/1541885.1541892>
- [7] L. J. Jia, G.L. Lin, G. Noubir, et al, "Universal approximations for TSP, Steiner tree, and set cover," *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp.386–395, May 22–24, Baltimore, MD, 2005.
- [8] D. S. Johnson and L. A. McGeoch, *The traveling salesman problem and its variations (Combinatorial Optimization)*, London: Springer press, 2004, pp. 445–487.
- [9] H. Ghaziri and I. H. Osman, "A neural network algorithm for the traveling salesman problem with backhauls," *Computers & Industrial Engineering*, vol.44, no.2, pp.267–281, February 2003. [http://dx.doi.org/10.1016/S0360-8352\(02\)00179-1](http://dx.doi.org/10.1016/S0360-8352(02)00179-1)
- [10] Y. H. Liu, "Different initial solution generators in genetic algorithms for solving the probabilistic traveling salesman problem," *Applied Mathematics and Computation*, vol.216, no.1, pp.125–137, March 2010. <http://dx.doi.org/10.1016/j.amc.2010.01.021>
- [11] Y. Liu, S. W. Xiong and H. B. Liu, "Hybrid simulated annealing algorithm based on adaptive cooling schedule for TSP," *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pp.895–898, Shanghai, June 12–14, 2009.
- [12] M. Dorigo, M. Birattari and T. Stützle, "Ant colony optimization. *IEEE Computational Intelligence Magazine*," vol.1, no.4, pp.28–39, November 2006.
- [13] W. N. Chen, J. Zhang, et al, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE T. Evolu. Comput.*, vol.14, no.2, pp.278–300, April 2010. <http://dx.doi.org/10.1109/TEVC.2009.2030331>
- [14] S. Iordache, "Consultant-guided search-a new metaheuristic for combinatorial optimization problems," *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pp.225–232, July 7–11, Portland, 2010.
- [15] Y. Li, A. M. Zhou and G. X. Zhang, "Simulated annealing with probabilistic neighborhood for traveling salesman problems," *Proceedings of the Seventh International Conference on Natural Computation*, pp.1565–1569, Shanghai, 26–28 July, 2011. <http://dx.doi.org/10.1109/ICNC.2011.6022345>
- [16] B. Liu and P. S. Meng, "Simulated annealing-based ant colony algorithm for traveling salesman problems," *Journal of Huazhong University of Science and Technology(Nature Science Edition)*, vol.37, no.11, pp.26–30 (in Chinese), November 2009.
- [17] V. Deineko, B. Klinz and G. Woeginger, "Four point conditions and exponential neighborhoods for symmetric TSP," *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp.544–553, Miami, January 22–26, 2006.
- [18] G.A.Croes, "A method for solving traveling salesman problems," *Operations Research*, vol.6, no.6, pp.791–8126, December 1958. <http://dx.doi.org/10.1287/opre.6.6.791>

AUTHORS

Y. Wang is with the School of Renewable Energy, North China Electric Power University, Changping, Bingjing 102206 China (e-mail: wangyyong100@163.com).

D. Tian is with the School of Renewable Energy, North China Electric Power University, Changping, Bingjing 102206 China (e-mail: tiande@ncepu.edu.cn).

Y.H. Li is with the School of Renewable Energy, North China Electric Power University, Changping, Bingjing 102206 China (e-mail: liyuhua0128@ncepu.edu.cn).

This article is an extended and modified version of a paper presented at the 2013 Chinese Intelligent Automation Conference (CIAC2013), held in Yangzhou, Jiangsu Province, China, in August 2013. Submitted 22 May 2013. Published as re-submitted by the authors 23 July 2013.