Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090220

Monte Carlo Simplification Model for Traveling Salesman Problem

Dong WANG[1,*] *and Dong-mei LIN*[2]

[1] Department of Computer Science and Technology, Foshan University, Foshan, Guangdong 528000, China
[2] Center of Information and Education Technology, Foshan University, Foshan, Guangdong 528000, China

Received: 22 May 2014, Revised: 21 Aug. 2014, Accepted: 23 Aug. 2014
Published online: 1 Mar. 2015

Abstract: Traveling Salesman Problem (TSP) is one of classical *NP-hard* problems in the field of combinatorial optimization. It is because of the problem complexity that almost all of the accurate computing algorithm could not find the global optimal solution (GOS) in a reasonable time. The complexity is characterized by the large number of edges in the initial edge set of TSP. By analyzing the relationship between GOS and high-quality local optimal solutions, it is found that the edge union set of some local optimal solutions could include most even all edges of GOS, and that their edge intersection could fix partial edges of GOS with high probability. The method reducing the initial edge set of TSP is established based on the probability statistic principle. The search space in which to solve the problem is cut down greatly, and the edge quantity in the new edge set is about twice times of the problem scale. Accureate algorithm can find GOS with high probability for TSPs whose scale are up to 200 nodes based on the simplified initial edge set. The method could be applied to many kinds of algorithms also used for solving TSP.

Keywords: traveling salesman problem, simplification, initial edge set

1 Introduction

Traveling salesman problem (TSP) is a classical *NP-hard* problem in the field of discrete and combinatorial optimization researches, and is one of the most intensively studied problems in operations research and theoretical computer science. Most of the work on TSP are motivated by its use as a platform for the study of general methods that can be applied to a wide range of discrete optimization problems. Since 1930 when the problem was first formulated as a mathematical problem, various methods were proposed so that some larger instances could be solved.

In this paper, a kind of Monte Carlo simplification model to reduce the initial edge set (IES) of symmetric TSP is proposed so that the problem could be simplified. The edge number in IES after simplified would be nearly twice of the problem scale, and IES would include most even all edges included by global optimal solution (GOS) with high probability. All strategies based on finding those edges belonging to GOS could be benefit from our work.

2 Review of the existing work

Many researches have concluded that the problem-solving essence is the process of finding and accumulating those edges belonging to GOS [1]. Therefore, fixing some GOS edges and deleting those edges not included by GOS will decrease the number of edges searched and searched by solving strategies, so it could improve their performance.

The multilevel approach proposed by Walshaw [2,3] introduced first the fixing edge method to coarsen recursively a given graph by matching and merging node pairs to generate smaller graphs at each level, and then uncoarsening each intermediate graph and finally resulting in a valid solution of the original problem. As the coarsening step defines the solution space of a recursion level, its strategy is decisive for the quality of the multilevel algorithm. Thomas etc. [4] used some tour-generating algorithms as fixing heuristics, including minimum spanning tree, nearest neighbor, lighter than median and close pairs. They found that the more edges the resulting tour has in common with the optimal tour. Zhou etc. [5] took full advantage of the conclusions by Thomas to improve the multilevel approach. They utilized Lin-kernighan local search algorithm to generate a pair of

* Corresponding author e-mail: wdong@fosu.edu.cn

high-quality tours, and then construct their edge intersection. The edges in the intersection belong to the optimal solution with high probability, so they proposed an improved multilevel algorithm using the intersection to coarsen. The new algorithm almost get optimal tour every time for instance in reasonable time and thus outperformed the known best ones in the quality of solutions and the running time.

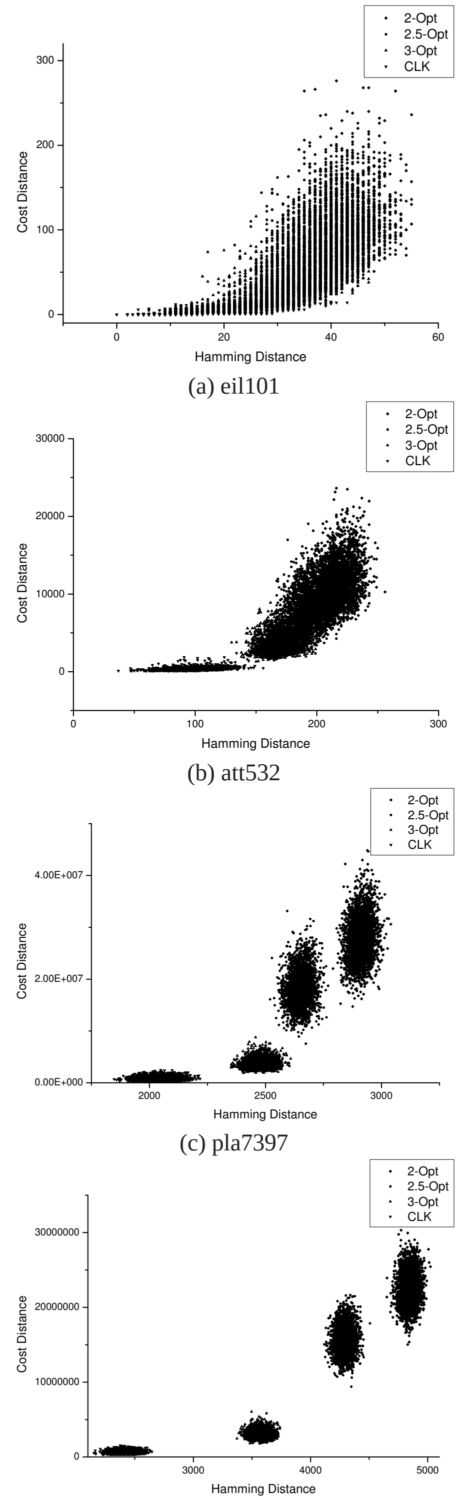
Just as Thomas's work, we found also that it is very restrictive to find and fix effectively some GOS edges from IES of a problem. In other words, the bottleneck to improve solving performance and quality of all edge-based strategies is due to the large quantity of edges in IES. Therefore, we propose a kind of Monte Carlo model selecting and fixing some edges to reduce the scale of IES.

3 Characteristic of local search algorithm

A tour of TSP is a Hamilton loop p which passes through each city once and only once, namely problem feasible solution. The tour cost is remarked as $W(p)$. All edges included in p make up of an edge set, $E(p)$. P is a tour set which includes all feasible solutions of a TSP instance. The goal is to find tours whose costs are equal to $\min_{p\in P} W(p)$.

Assumed that f_{LS}:$P \to P$ is a mapping from a tour to another tour under the function of a local search algorithm, let p'=$f_{LS}(p)$ then $W(p')\leq W(p)$. Let P_{LS}=$\{s|s = f_{LS}(t), t \in P\}$. $\forall t \in P$, $W[f_{LS}(t)]\leq W(t)$, so $P_{LS} \subseteq P$. Especially, $P_{LS} \subset P$ when $P - P_{LS} \neq \emptyset$. Fitness distance correlation analysis [1] reveals the performance difference among local search algorithms. We generated randomly 2,500 tours, and optimized these tours using 2-Opt, 2.5-Opt, 3-Opt and chained Lin-kernighan algorithm (CLK), then generated the figure according to fitness distance correlation analysis method, as Figure 1. The difference becomes more obvious along with the expansion of problem scale. The function f_{LS} cuts problem solution space as shown in Figure 2.

We know from Figure 2 that TSP could be simplified if P_{LS} could be generated, but it is impossible for us to generate P_{LS} for all TSPs. Assumed that P_G is the GOS set of a TSP instance, $\forall(s,t) \in P_G, W(s) = W(t)$. For $\forall t \in P_G$, a local search algorithm could not find a better solution after applying once optimization operation, we have $f_{LS}(t) = t$ and $W[f_{LS}(t)]$=$W(t)$. Furthermore $P_G \subseteq P_{LS} \subseteq P$ and $|P_G| \leq |P_{LS}| \leq |P|$. For the most of TSP instances, we have $P_G \subset P_{LS} \subset P$ and $|P_G| < |P_{LS}| < |P|$. Therefore, searching the solution in P is equivalent to searching the solution in P_{LS}. If $|P_{LS}| < |P|$ or $|P_{LS}| \ll |P|$ comes into existence, the problem solution space of TSP would be greatly reduced.



(a) eil101

(b) att532

(c) pla7397

(d) usa13509

Fig. 1: FDC of different local search algorithms over 2,500 stochastic solutions to eil101, att532, pla7397 and usa13509 from TSPLIB95

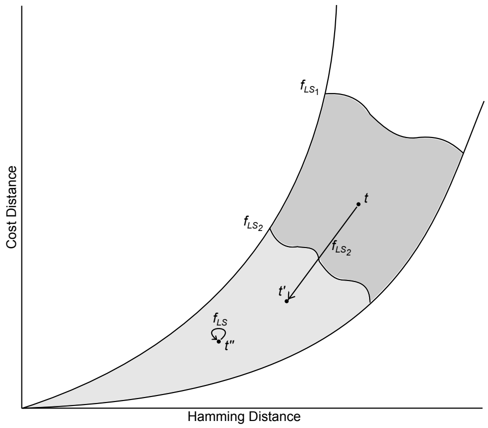


Fig. 2: Cutting function of a local search algorithm to the solution space of TSP

4 Reducing IES by the tour union set method

Lemma 1 $\bigcup_{g\in P_G} E(g) \subseteq \bigcup_{s\in P_{LS}} E(s)$.

[Proof] $P_G \subseteq P_{LS}$, for $\forall g \in P_G$, there exists $g \in P_{LS}$ and $E(g) \subseteq \bigcup_{s\in P_{LS}} E(s)$, so for $\forall e \in E(g)$, there is $e \in \bigcup_{s\in P_{LS}} E(s)$. To take a step further, $\bigcup_{g\in P_G} E(g) \subseteq \bigcup_{s\in P_{LS}} E(s)$.

[Done]

According to Lemma 1**,** it is a evident that GOS could be found in P_{LS}. For $\forall e \in \bigcup_{g\in P_G} E(g)$, $Prob.\left[e \in \bigcup_{s\in P_{LS}} E(s)\right] = 1$ is a certain event. If the distribution of e belonging to p stochastically selected from P_{LS} is symmetrical, the probability that e is contained in p is:

$$Prob.\left[e \in E(p)\right] = \frac{\left|\{s | e \in E(s), \forall s \in P_{LS}\}\right|}{|P_{LS}|} \quad (1)$$

$0 < Prob.\left[e \in E(p)\right] \leq 1$ in formula (1) is probability event. Given $P_c = \min_{e\in \bigcup_{g\in P_G} E(g), p\in P_{LS}} Prob.\left[e \in E(p)\right]$. Taking out stochastically K tours $p_1, p_2, \ldots, p_K$ from P_{LS}, it is an independent event because each tour could be computed independently. According to the countable additivity of probability:

$$\begin{aligned} &Prob.\left[e \in \bigcup_{k=1}^{K} E(p_k)\right] \\ &= Prob.\left[e \in E(p_1)\right] + \cdots + Prob.\left[e \in E(p_K)\right] \\ &\geq K \times P_c \end{aligned} \quad (2)$$

Hereby, a least positive integer $K = k$ which makes $k \times P_c \geq 1$ must exist, furthermore $Prob.\left[e \in \bigcup_{k=1}^{K} E(p_k)\right] \geq 1$ turns to a certain event. From formula (2) we get

$$K = \left\lceil 1/P_c \right\rceil \quad (3)$$

The conclusion come into existence for $\forall e \in \bigcup_{g\in P_G} E(g)$. P_c is a priori probability which could be established by some statistic experiments.

Conclusion 1 The edge union set, formed by those tours selected stochastically from P_{LS}, includes all of the edges of a GOS when the number of tours reaches a certain number.

The priori probability P_c can be established through a large number of statistic experiments, consequently K is established. The experimental procedure is as follows:

1) Generating 2,500 stochastic solutions for each instance;

2) Utilizing CLK to optimize all of these solutions, and obtaining 2,500 optimized solutions;

3) Counting the occurrence times of all edges included by a given GOS, and recording the minimum occurrence times among them.

In these experiments, the primary parameters of CLK algorithm are set up as follows. The repeated counter is 1, the kicks type is a random type, and the reference optimization edge set is generated by Quadrant 3-nearest neighbor algorithm. All GOSs are computed by CONCORDE[1] or downloaded from TSPLIB95. The statistic results are shown in Table 1.

According to the above analysis, P_c=101/2500, and K=2500/101≈50. The following stochastic algorithm is established, which uses the conclusion drawn in this paper.

Algorithm 1: TSP edge set cutting algorithm
Input: TSP dataset
Output: Reduced TSP initial edge set
Begin
  1) Initialization
    (1) $CE \leftarrow \emptyset$;
    (2) *RepeatTimes*←1;
    (3) *KickType*←Random kicks type;
    (4) Establishing Quadrand 3-nearest neighbor edge set N_e;
  2) Repeat for K=50 times, do
    (1) Initializing a tour p stochastically;
    (2) p'←CLK(*RepeatTimes*, *KickType*, N_e, p);
    (3) CE←CE∪$E(p')$;
  3) return *CE*;
End.

Fig. 3: TSP initial edge set cutting algorithm

[1] http://www.tsp.gatech.edu/concorde.html

Table 1: Statistic data for minimum edge occurrence times of the GOS

Dataset Name	Occurrence Times	Dataset Name	Occurrence Times
att48	2301	rat195	293
berlin52	2500	rat99	750
bier127	1727	rd100	2500
burma14	2500	st70	1201
ch130	960	u159	1019
ch150	655	wi29	2455
d198	547	a280	432
eil51	1552	ali535	518
eil76	859	att532	331
eil101	785	d493	184
kroa100	2500	d657	365
kroa150	892	fl417	173
kroa200	2460	gil262	599
krob100	2109	lin318	355
krob150	892	p654	464
krob200	2454	pcb442	259
kroc100	2500	pr226	267
krod100	2488	pr264	2305
kroe100	1409	pr299	613
lin105	2468	rat575	143
pr107	464	rat783	459
pr124	2263	**rd400**	**101**
pr136	885	ts225	352
pr144	2500	tsp225	1958
pr152	2204	u574	522
pr76	443	u724	157

5 Fixed partial edges belonging to GOS

In general, if $W(s) < W(t)$, $\exists s,t \in P$, s is the solution with higher quality than t. The similarity between two tours is adopted to describe solution quality because the objective studied in this paper is to fix those edges belonging to GOS. There are many ways to define the similarity between the tours, and Hamming distance is used in this paper.

Definition 1 The ratio of the same edge number in both s and t to the edge number in the tour length is called the similarity between the two tours, denoted by $S(s,t)$.

$$S(s,t) = \frac{|E(s) \cap E(t)|}{N} \tag{4}$$

From Definition 1, we know that $S(s,t)$ is a real number belonging to a closed interval [0,1].

Definition 2 The minimum similarity value among all local optimal solutions to GOS is called the solution quality of a local optimal solution set, indicated by D_{LS}.

$$D_{LS} = \min_{\forall s \in P_{LS}, \forall g \in P_{GOS}} S(s,g) \tag{5}$$

The larger D_{LS} is the closer local optimal solutions are to GOS, and the higher solution quality of P_{LS} is.

Property 1 $\forall e \in \bigcup_{\forall s \in P_{LS}} E(s)$, the probability of that the edge set intersection of K_1 tours selected stochastically from P_{LS} includes e is inversely proportional to K_1.

[Proof] $\forall e \in \bigcup_{\forall s \in P_{LS}} E(s)$, the probability of that a tour selected stochastically from P_{LS} includes e is just like the formula (1). Taking stochastically out K_1 tours $p_1, p_2, \cdots, p_{K_1}$ from P_{LS} is an independent event because each tour can be computed independently. While constructing the intersection of these tours, we have

$$\begin{aligned}
&\text{Prob.}\left[e \in \bigcap_{k=1}^{K_1} E(p_k)\right] \\
&= \text{Prob.}\left[e \in E(p_1)\right] \times \text{Prob.}\left[e \in E(p_2)\right] \times \\
&\quad \cdots \times \text{Prob.}\left[e \in E(p_{K_1})\right] \\
&\leq P_c^{K_1}
\end{aligned} \tag{6}$$

According to formula (6), the larger the value of K_1 is the smaller the value of $P_c^{K_1}$ is. While the value of $P_c^{K_1}$ approximates to 0, the event becomes a lower probability event.

[Done]

It is difficult for us to establish P_c in Property 1 because it is impossible for us to enumerate all elements in P_{LS}. However, its value is closely connected to D_{LS}. The larger the value of D_{LS} is the more similar all elements in P_{LS} are to the GOS. There are fewer elements in P_{LS}, so $|P_{LS}|$ is inversely proportional toP_c.

Property 2 While $D_{LS} \to 1$, if the edge intersection of K_1 tours selected stochastically from P_{LS} has not been empty, the probability of that the edges in the intersection belong to GOS is proportional to K_1.

[Proof] According to Definition 2 and $D_{LS} \to 1$, $\forall s \in P_{LS}, \forall g \in P_{GOS}$, $E(s) \bigcap E(g) \neq \emptyset$. Let $L = |E(s) \cap E(g)|$, and L is an integer greater than 0. Let $L_{\min} = \min_{s \in P_{LS}, g \in P_{GOS}} |E(s) \bigcap E(g)| \geq \lfloor N \times D_{LS} \rfloor$. $\forall e \in E(s)$, the probability which e belongs to the GOS is $Prob.[e \in E(g)] \geq L_{\min}$.

Because $\bigcap_{k=1}^{K} E(p_k) \subseteq \bigcup_{k=1}^{K} E(p_k)$, the probability property of the elements in both edge sets is satisfied to formula (6). If we select stochastically K_2 tours from P_{LS}, we have

$$Prob.\left[e \in \bigcup_{g \in P_G} E(g)\right] \geq K_2 \times \frac{L_{\min}}{N} = K_2 \times D_{LS} \tag{7}$$

By the same token, let $L_{\max} = \max_{s \in P_{LS}, g \in P_{GOS}} |E(s) - E(g)| \leq N \times (1 - D_{LS})$. $\forall e \in E(s) \wedge e \notin \bigcup_{g \in P_G} E(g)$, having

$$Prob.\left[e \notin E(g)\right] \leq \frac{L_{\max}}{N} \leq 1 - D_{LS} \tag{8}$$

According to formula (6) and (8), we have

$$Prob.\left[e \notin \bigcup_{g\in P_G} E\left(g\right)\right] \leq \left(1-D_{LS}\right)^{K_2} \qquad (9)$$

While $D_{LS} \rightarrow 1$, the probability value in formula (7) will reach 1, and the event becomes a certain event. The probability value in formula (8) will approach zero, and the event becomes an improbable event.

[Done]

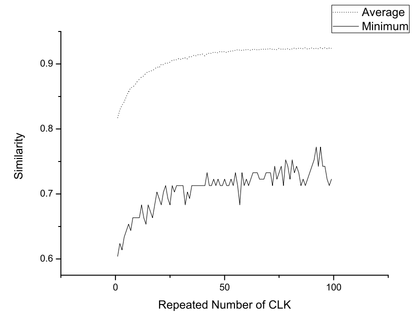
We know that formula (6) and (8) are directly related to D_{LS} from Property 2, and could conclude as follows.

Conclusion 2 The elements in the intersection $E*$, which are established by K_1 tours selected stochastically from P_{LS} satisfy to the formula (6) and (8). While the value K_1 is closer to a certain number, the algorithm established based on Property 2 satisfies to the property of Monte Carlo algorithm, so new algorithm used to fix partial edges belonging to GOS could be established as well.

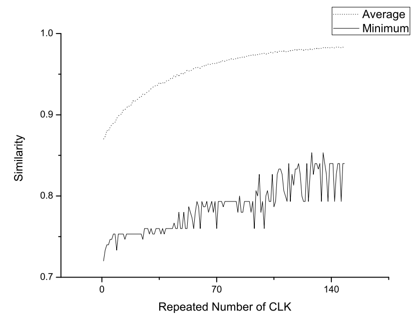
The priori probability, D_{LS}, must be established before we utilize the new algorithm to solve problems. The primary parameters of CLK algorithm are the same as above-mentioned experiments but the repeated counter is the experimental object. The experimental procedure is as follows: initializing stochastically 2,500 tours, setting the repeated number to 1, the others parameters are the same, utilizing CLK to optimize all tours and getting a group of local optimal solutions, computing the similarities of each local optimal solutions to GOS, recording the maximum value, the minimum value and the average value; and then the repeated number increases 1, repeating the procedure as above-mentioned procedure until the repeated number is equal to the problem scale N. The experimental results of three datasets from TSPLIB95, eil101, ch150 and a280, are given in Figure 4.

We concluded that the similarity is greater than 0.7 while the repeated number is to the scale of the problems. It can be said also that a certain local optimal solution includes at least 70% edges belonging to GOS, D_{LS}=0.7.

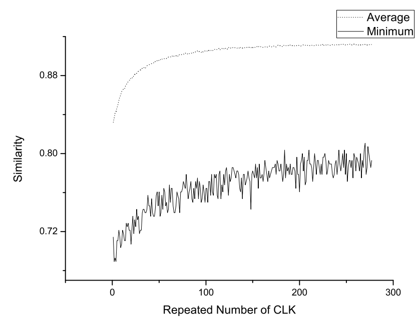
The threshold M is set toguarantee that formula (8) comes into existence. When $Prob.\left[e \in \bigcup_{g\in P_G} E\left(g\right)\right] \leq \left(1-D_{LS}\right)^{K_2} < M$, the edge e couldn't be fixed. The threshold M is set to 1.0×10^{-12}, and then K_2=24, according to Table 2. Therefore, the edges in the intersection established by 24 tours selected P_{LS} belong to the GOS with higher probability. On the contrary, the probability that the edges don't belong to the GOS is too lower. It can be considered an impossible event according to the threshold M.



(a) eil101



(b) ch150



(c) a280

Fig. 4: Similarity experiments of eil101, ch150 and a280 from TSPLIB95

6 Integrated simplification algorithm for IES of TSP

According to above-mentioned analysis and experimental conclusion, integrated simplification algorithm for IES can be established as Algorithm 2.

While combining Algorithm 2 and Algorithm 1, we could get a more efficacious simplification algorithm for IES. Table 3 illustrates the experimental results of the algorithm.

The experimental environment in this paper is Intel T2300E 1.66GHz microprocessor, 1GB RAM, Microsoft Windows XP operating system.

Table 2: Statistic data for average similarity of 2 500 tours to the GOS

Dataset	Similarity (%)		Dataset	Similarity (%)	
Name	Minimum	Average	Name	Minimum	Average
att48	0.9988	0.8750	rat195	0.9394	1.0000
berlin52	1.0000	1.0000	rat99	0.7897	0.9422
bier127	0.8425	0.9931	rd100	1.0000	1.0000
burma14	1.0000	1.0000	st70	0.8857	0.9512
ch130	0.7923	0.9748	u159	0.9748	0.9837
ch150	0.8400	0.9832	wi29	1.0000	1.0000
d198	0.8384	0.9084	a280	0.7857	0.9122
eil51	0.7059	0.9853	ali535	0.8411	0.9064
eil76	0.7763	0.9385	att532	0.8252	0.9049
eil101	0.7327	0.9247	d493	0.7343	0.8440
kroa100	1.0000	1.0000	d657	0.8189	0.9117
kroa150	0.9333	0.9845	fl417	0.8417	0.8983
kroa200	0.9350	1.0000	gil262	0.7939	0.9372
krob100	0.9400	0.9994	lin318	0.7799	0.9418
krob150	0.8600	0.9188	p654	0.7034	0.7760
krob200	0.9650	0.9999	pcb442	0.7330	0.8657
kroc100	1.0000	1.0000	pr226	0.9558	0.9842
krod100	1.0000	1.0000	pr264	0.9470	0.9994
kroe100	0.7700	0.9750	pr299	0.8328	0.9567
lin105	1.0000	1.0000	rat575	0.7722	0.8820
pr76	0.8684	0.9998	rat783	0.8404	0.9463
pr107	0.7383	0.9409	rd400	0.8275	0.8925
pr124	0.9435	1.0000	ts225	0.7511	0.8464
pr136	0.8309	0.9305	tsp225	0.8133	0.9993
pr144	1.0000	1.0000	u574	0.8084	0.9276
pr152	0.9211	0.9998	u724	0.8177	0.8917

Table 3: Simplified results of the combinational algorithm

Dataset	Edge Number		Dataset	Edge Number	
Name	Fixed	Simplified	Name	Fixed	Simplified
att48	48	48	rat195	100	327
berlin52	52	52	rat99	99	99
bier127	117	138	rd100	100	100
burma14	14	14	st70	58	79
ch130	90	185	u159	125	205
ch150	123	183	wi29	29	29
d198	137	268	a280	128	446
eil51	48	53	ali535	371	729
eil76	52	103	att532	301	867
eil101	77	123	d493	189	913
kroa100	100	100	d657	388	1082
kroa150	139	165	fl417	304	556
kroa200	200	200	gil262	183	381
krob100	100	100	lin318	200	528
krob150	122	187	p654	250	1133
krob200	200	200	pcb442	190	723
kroc100	100	100	pr226	214	235
krod100	100	100	pr264	264	264
kroe100	69	136	pr299	189	444
lin105	105	105	rat575	280	1008
pr76	76	76	rat783	423	1319
pr107	79	129	rd400	217	667
pr124	116	135	ts225	144	368
pr136	114	176	tsp225	225	225
pr144	144	144	u574	311	958
pr152	152	152	u724	401	1183

Algorithm 2: Integrated simplification algorithm for IES
Input: dataset instance of TSP
Output: Simplified IES
Begin
1) Initialization
(1) Generating original initial edge set *IE* utilizing normal methods;
(2) $RE \leftarrow \emptyset$; //Initial edge set after cutting
2) Repeating for K=24 times, do
(1) Generating stochastically a tour p;
(2) $p' \leftarrow$CLK(N, *Random KickType*, *IE*, p);
(3) if p' is the first one then
$RE \leftarrow RE \cup E(p')$;
Otherwise
$RE \leftarrow RE \cap E(p')$;
3) Computing the degrees of all nodes;
4) Deleting all edges in *IE* that include such node whose degree is equal to 2;
5) $IE \leftarrow IE \cup RE$;
6) Returning *IE*;
End

Fig. 5: Integrated simplification algorithm for IES

7 Benefit from our work

Main strategy of branch and cut (BAC) algorithm is searching solution space tree using breadth first search or least cost first manner. Held-karp (HK) algorithm as BAC uses the cutting edge set by Algorithm 2 as the referring optimal edge set. While the better fitness tour is found, it will be optimized using CLK algorithm, and update the upbound of HK algorithm. The experimental result for those TSPLIB datasets whose city quantity is less than or equal to 1,000 is showed in Table 4. Each experiment instance is repeated 30 times and the average values are computed.

Improved HK algorithm is able to find the GOS of those instances whose number of cities is less than and equal to 200, and some other instances. Comparing to original HK algorithm, computation time of new algorithm is shortened sharply, and the solvable problem scale is increased by a wide margin. The experimental results of 1,000 TSPs produced stochastically and whose number of cities is less or equal to 200 indicates that new algorithm could find the GOSs of 999 instances.

Table 4: Simplified results of the combinational algorithm

Dataset	HK Runtime(s)		Dataset	HK Runtime(s)	
Name	Basic	Hybrid	Name	Basic	Hybrid
att48	10.672	1.859	rat195	-[1]	10.391
berlin52	41.359	0.672	rat99	-	1.328
bier127	-	4.469	rd100	-	1.547
burma14	0.031	0.250	st70	-	1.485
ch130	-	4.156	u159	-	2.484
ch150	-	3.765	wi29	0.047	0.422
d198	-	10.969	a280	-	-
eil51	23.063	0.891	ali535	-	-
eil76	186.485	1.578	att532	-	-
eil101	-	2.438	d493	-	-
kroa100	-	1.485	d657	-	-
kroa150	-	3.172	fl417	-	-
kroa200	-	4.422	gil262	-	10.234
krob100	-	2.406	lin318	-	17.469
krob150	-	4.937	p654	-	-
krob200	-	6.422	pcb442	-	-
kroc100	-	1.641	pr226	-	5.125
krod100	-	1.750	pr264	-	5.844
kroe100	-	2.313	pr299	-	-
lin105	-	1.203	rat575	-	-
pr76	-	1.860	rat783	-	-
pr107	-	2.672	rd400	-	-
pr124	-	2.954	ts225	-	-
pr136	-	4.469	tsp225	-	7.719
pr144	-	4.485	u574	-	-
pr152	-	7.391	u724	-	-

[1] The symbol - represents no the optimal solution found of the problem after 1,000,000 tries.

8 Conclusion

The probability characteristic relationship between optimal solutions is established through probability analysis of the relationship between high-quality local optimal solutions and GOS of TSP in this paper. Union set of much high-quality local optimal solutions edge set includes edges belong to GOS edge set in a high probability. New method proposed in this paper decreases sharply the problem initial edge set scale, curtails sharply the searching space for searching optimal solutions, and makes different algorithms faster than the existing algorithms to find GOS. The new method can be applied to many kinds of algorithms to solve TSP, and establishes the simplification basement of analogous TSP. It is because of complexity of TSP that how to establish effective solving algorithm for more large scale TSP is further studied.

Acknowledgement

This paper is supported by Natural Science Foundation of Guangdong Province (10152800001000029) and Guangdong Province scientific and technological project (2011B010200031).
The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] K. Boese. Cost versus distance in the traveling salesman problem. TR-950018, Computer Science Department, UCLA, 1995.
[2] C. Walshaw. A Multilevel Approach to the Travelling Salesman Problem. Operations Research, **50**, 862-877 (2002).
[3] C. Walshaw. Multilevel Refinement for Combinatorial Optimisation Problems. Annals of Operations Research, **131**, 325-372 (2004).
[4] T. Fischer, P. Merz. Reducing the size of traveling salesman problem instances by fixing edges. Proceedings of 7^{th} European Conference on Evolutionary Computation in Combinatorial Optimisation, **4446**, 72-83 (2007).
[5] P. Zou, Z. Zhou, G. L. Chen, and et al.. A Multilevel Reduction Algorithm to TSP. Journal of Software, **14**, 35-42 (2003).

Dong WANG is Associate Professor of Computer Science and Technology at Foshan University. He was the Vice director of Electronic and Information Engineering School. He received the PhD degree in Land and Resources Information Engineering at Central South University (China). He is referee and Editor of several Chinese core journals in the frame of intelligent computation, combinatorial optimization. His main research interests are: intelligent computation, optimization theory, high performance computing, image processing and recognition.

Dong-mei LIN is Professor of Computer Science and Technology at Foshan University. She was the director of the Center of Information and Education Technology, visiting scholar at Royal Melbourne Institute of Technology University, director of the Natural Science Fund Project of Guangdong Province . Her main research interests are: intelligent computation, combinatorial optimization, virtual reality.