AN EFFICIENT ANT COLONY SYSTEM FOR SOLVING THE NEW GENERALIZED TRAVELING SALESMAN PROBLEM

Lianming Mou

Key Laboratory of Numerical Simulation of Sichuan Province, Neijiang Normal University, Sichuan, China
mlianming@163.com

Abstract

The Generalized Traveling Salesman Problem (GTSP) is an extension of the classical traveling salesman problem and has many interesting applications. In this paper we present a New Generalized Traveling Salesman Problem (NGTSP), and the current GTSP is only a special case of the NGTSP. To solve effectively the NGTSP, we extend the ant colony system method from TSP to NGTSP. Meanwhile, to improve the quality of solution, a local searching technique is introduced into this method to speed up the convergence, and a novel parameter adaptive technique is also introduced into this method to avoid locking into local minima. Experimental results on numerous TSPlib instances show that the proposed method can deal with the NGTSP problems fairly well, and the developed improvement techniques is significantly effective.

Keywords: GTSP; NGTSP; ACS; parameter adaptive

1 Introduction

The Generalized Traveling Salesman Problem (GTSP) extends the classical Traveling Salesman Problem (TSP). In the literature, there are two versions of the problem: One in which we are interested in finding the shortest closed tour visiting exactly one node from each cluster. This problem is called the generalized traveling salesman problem (GTSP) and has been introduced independently by Srivastava et al. [1], Henry-Labordere [2] and Saskena [3]; the other is the problem of finding the shortest closed tour including at least one node from each cluster. This version of the problem was introduced by Noon and Bean [4] and by Laporte and Nobert [5]. When edge costs satisfy the triangle inequality, even if we allow more than one node per cluster to be connected, an optimal solution of the GTSP always contains only one node from each cluster [5]. When edge costs do not satisfy the triangle inequality, the second case can be transformed into the first one by calculating shortest path between any two nodes and add virtual edge on the graph. Currently most researchers concentrate the research on the problem of choosing exactly one node from each cluster.

The GTSP has many interesting applications, such as location problems, planning, postal routing, logistics, manufacture of microchips, telecommunication problems, railway optimization. More information on the problem and its applications can be found in Laporte and Nobert[5], Fischetti, Salazar and Toth[6], Pop et al. [7], Snyder and Daskin [8], etc.

Since GTSP requirements to visit exactly one node from each cluster, this will limit its application scope. Because in practical applications, a useful tour often not visit exactly one node from each cluster, but many nodes; in order to further expand the application range of the GTSP, we put forward a new and more general GTSP problem, and design an effective ant colony system for solving the new GTSP. Meanwhile, to speed up the convergence, an effectively local searching technique is also introduced into this method. To avoid locking into local minima, a novel parameter adaptive technique is also introduced into this method. Numerical simulation on 30 TSPlib instances shows that the proposed method can deal with the NGTSP problems fairly effectively.

2 Related work

2.1 The generalized traveling salesman problem

The GTSP is known to be an NP-hard problem. A lot of attention was paid by the researchers for solving effectively the GTSP. Existing approaches focus on exploiting the correlations between TSP and GTSP, which can be roughly grouped into three categories, namely transforming approaches, heuristic approaches and hybrid approaches.

The transforming approaches are to transform the GTSP into the classical TSP. They have been studied by Lien et al. [9], Noon and Bean [10], Dimitrijevic and Saric [11], Laporte and Semet [12], and Behzad and Modarres[13]. But many transformations substantially increase the numbers of nodes and edges, are therefore of limited practical value. Furthermore, some transformations even require additional constraints, thus making general

978-1-61284-204-2/11/$26.00 ©2011 IEEE

algorithms for the classical TSP inapplicable. The difficulty of obtaining optimal solutions for the GTSP has led to the development of several heuristic and meta heuristic algorithms: an efficient composite heuristic [14], a random key genetic algorithm [8], variable neighborhood search [15], effective neighborhood structures [16], memetic algorithms [17,18]. In order to improve the solution, some effective hybrid approaches was given, such as an efficient genetic algorithm [19], hybrid metaheuristic algorithm [20].

2.2 Ant Colony System for GTSP

Dorigo et al. developed Ant Colony System (ACS) which is applied to the TSP [21]. Yang et al. [22] and Pintea et al. [23] extended the ant colony algorithms from TSP to GTSP by only simple modifying the tabu list, but not make fully use of the characteristic of GTSP. Detailed introduction is as follows.

(1) *ACS state transition rule*

In ACS the state transition rule is as follows: an ant positioned on node *i* chooses the city *j* to move to by applying the rule given by Eqs. (1) and (2)

$$j=\begin{cases}\arg\max\limits_{u\notin tabu_k}\{[\tau_{iu}(t)]\cdot[\eta_{iu}(t)]^{\beta}\}, & if\ \ q\le q_0\\ S, & else\end{cases} \tag{1}$$

$$P_{ij}^{\,k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]\cdot[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\notin tabu_k}[\tau_{is}(t)]\cdot[\eta_{is}(t)]^{\beta}}, & if\ \ j\notin tabu_k\\ 0, & else\end{cases} \tag{2}$$

$$tabu_k=\{x\mid x\in V, x\in \mathrm{C}(v), \forall v\in CT\}$$

where τ_{iu} is the pheromone, $\eta_{iu}=1/d_{iu}$ is the inverse of the distance d_{iu}, $\mathrm{C}(v)$ is the cluster containing node v, CT abbreviations for the current tour, and β is a parameter which determines the relative importance of pheromones versus distance ($\beta>0$), q is a random number uniformly distributed in [0,1], q_0 is a parameter ($0\le q_0\le 1$), and S is a random variable selected according to the probability distribution given in Eq. (2). The state transition rule derived from Eqs.(1) and (2) is called *pseudo-random -proportional rule.*

(2) *ACS local updating rule*

While building a solution (i.e., a tour) of the TSP, ants visit edges and change their pheromone level by applying the local updating rule of Eq.(3)

$$\tau_{ij}\leftarrow(1-\xi)\tau_{ij}+\xi\tau_0 \tag{3}$$

where $0<\xi<1$ is a parameter, τ_0 is the initial value of pheromones.

(3) *ACS global updating rule*

In ACS only the globally best ant is allowed to deposit pheromone. This choice, together with the use of the pseudo-random-proportional rule, is intended to make the search more directed. Global updating is performed after all ants have completed their tours. The pheromone level is updated by applying the global updating rule of Eq. (4)

$$\tau_{rs}\leftarrow(1-\rho)\tau_{rs}+\rho\Delta\tau_{rs}$$
$$\Delta\tau_{rs}=\begin{cases}(L_{gb})^{-1}, & if\ (r,s)\in global-best-tour\\ 0, & else\end{cases} \tag{4}$$

where ρ is the pheromone decay parameter, and L_{gb} is the length of the globally best tour from the beginning of the trial.

3 Definition of the New GTSP

Let $G=(V,E,W)$ be n-node undirected weighted complete graph whose node set V, edge set E, and edge cost function $W:E\rightarrow R^{+}$. Node set V is partitioned into k pairwise disjoint clusters $V_1,V_2,\cdots,V_k$, $i,j=1,2,\cdots,k$,

$$\bigcup_{i=1}^{k}V_i=V;\ V_i\bigcap V_j=\phi,\ \ i\ne j.$$

A solution to the *New Generalized TSP* (NGTSP) defined on G is a subgraph $S=(P,T)$, where $P\subseteq V$ connects exactly k_i node from the *i*-th cluster, that is

$$\left|S\bigcap V_i\right|=k_i,\ 1\le k_i\le|V_i|,\ i=1,2,\cdots,k. \tag{5}$$

$T\subseteq E$ is a closed tour, see Fig 1. The costs of such a closed tour are its total edge costs, i.e. $C(T)=\sum_{(u,v)\in T}c(u,v)$, and the objective is to identify a solution with minimum costs.

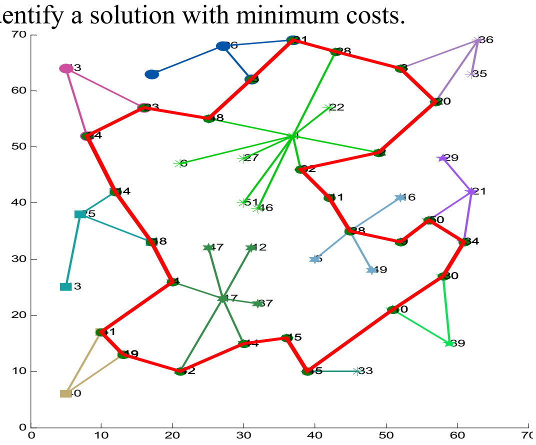

Figure 1. Example for a NGTSP solution

Obviously the NGTSP is a more general extension of the classical traveling salesman problem. The NGTSP contains the classical TSP as a special case when $k_i=|V_i|,\ i=1,2,\cdots,k$, and contains the GTSP as a special case when $k_i=1,\ i=1,2,\cdots,k$.

According to the order of nodes in same group which appears in the closed tour, the NGTSP has two cases: One in which the order of nodes in same group can be required to appear continuously in the closed tour; the other has not the requirement. In the paper we confine ourselves to the second case.

4 An Efficient Ant Colony System for the New Generalized Traveling Salesman

ACS is often applied to the TSP and GTSP because it allows integrating other constraints into the entire search process. We extend the ant colony system method from TSP to NGTSP according to the characteristic of the NGTSP.

4.1 ACS state transition rule for the NGTSP

The modifications of tabu list in NGTSP is different from these in TSP and GTSP, it contains two parts.

The first part is $tabu_k(1)$ when the number of nodes in cluster c which appear in the current tour (CT) equal to k_c, that is

$$tabu_k(1) = \{x \mid x \in V \wedge x \in C(v), \forall v \in CT \wedge N(v) = k_c\}.$$

The second part is $tabu_k(2)$ when the number of nodes in cluster c which appear in the current tour is less than k_c, that is

$$tabu_k(2) = \{x \mid x \in V, \forall x \in CT \ \ but \ \ N(x) < k_c\}.$$

So the tabu list is

$$tabu_k^* = tabu_k(1) \cup tabu_k(2). \tag{6}$$

After modifying the tabu list in ACS, the state transition rule will be rewritten as follows:

$$j = \begin{cases} \arg\max\limits_{u \notin tabu_k^*} \{[\tau_{iu}(t)] \cdot [\eta_{iu}(t)]^{\beta}\}, & if \ \ q \le q_0 \\ S, & else \end{cases} \tag{7}$$

$$P_{ij}^{\,k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)] \cdot [\eta_{ij}(t)]^{\beta}}{\sum\limits_{s \notin tabu_k^*} [\tau_{is}(t)] \cdot [\eta_{is}(t)]^{\beta}}, & if \ \ j \notin tabu_k^* \\ 0, & else \end{cases} \tag{8}$$

4.2 ACS global updating rule with adaptive parameter for the NGTSP

Global pheromone updating is mainly used to speed up the convergence. The pheromone decay parameter ρ has a great influence on the evolution of the entire ant colony system state; and its size is directly related to the global search ability and convergence speed in ant colony algorithm. When ρ is bigger, because the positive feedback of pheromone play a dominant role, convergence is fast, but it is easy to fall into local optimum and stagnation; the other hand, when ρ is smaller, because the positive feedback of pheromone is relatively weak, which increases the randomness of searching , ACS will has a slow convergence speed and a poor solution quality. At present, the pheromone decay parameter ρ generally takes a constant value, not be adjusted according to the current whole searching state and the pheromone distribution. In order to speed up the convergence and to prevent premature ahead, in this paper ρ will be adaptively adjusted according to the whole pheromone distribution. Since the number of *Average Node Branching* (*ANB*) [24] can reflect the diversity of the internal solution and the whole state of pheromone distribution, we let ρ adaptively change with the *ANB*. Generally, when the number of *ANB* is half of nodes, the global searching ability is strongest. So the global updating rule will be rewritten as follows:

$$\begin{aligned} &\tau_{rs} \leftarrow (1-\rho(t))\tau_{rs} + \rho(t)\Delta\tau_{rs} \\ &\rho(t) = \rho_0\left(\frac{2ANB(t-1)-m-1}{m-3}\right)^2 + 0.05 \\ &\Delta\tau_{rs} = \begin{cases} (L_{gb})^{-1}, & if \ (r,s) \in global-best-tour \\ 0, & else \end{cases} \end{aligned} \tag{9}$$

where $\rho(t)$ is the pheromone decay parameter in t-th round, $ANB(t)$ is the number of ANB in t-th round. When the number of ANB is two, this means that all the ants choose the same path, which is stagnation.

4.3 ACS local searching for the NGTSP

To speed up the convergence, a local searching technique called "*2-OPT Local search*" [25] can be improve a tour, which can also be applied to the NGTSP to delete the crossover of traveling lines.

We denote a tour as $a_1, a_2, \cdots, a_i, \cdots, a_n, a_1$, $a_i \in V_i$, where a_i is the position number of node i, $L_c = \sum_{l=1}^{m-1} d(a_l, a_{l+1}) + d(a_m, a_1)$ is the sum of all edge cost on a tour c. The pseudo-code of the local searching process with the characteristic of NGTSP is shown as Fig. 2.

```
For (i = 1; i<m-3; i++) {
    For (j= i+2; j<m; j++) {
        L_T = d(a_i, a_{i+1}) + d(a_j, a_{j+1});
        L_N = d(a_i, a_j) + d(a_{i+1}, a_{j+1});
        If ( L_T > L_N ) {
            L_c = L_c - (L_T - L_N);
            For (k=0; k < (j-i)/2; k++)
                { swap(a_{j-k}, a_{i+k+1}) ;} }
} }
```

Figure 2. Pseudo-code of the local searching

Time complexity of the local searching algorithm is $O(m^2)$.

4.4 The efficient ACS for the NGTSP

After modifying the tabu list and introducing local searching and adaptive parameter method, an efficient ant colony system (EACS) for NGTSP is as follows.

Input: W, β, ξ, ρ, q_0, l, m, K, $TIME\_MAX$

%$K = (k_1, k_2, \cdots, k_m)$

Output: the best closed tour

1. Initialize:

(1) Set ITime=0; %ITime is a counter of iteration times

(2) For every edge(i,j) {set an initial $\tau_{ij}(t) = c$ for

trail density and τ_0 ;}

2. Set s=0; %s is a counter of travel steps
For (k=1; k<= l; k++) {
Place ant k on a node v randomly;
If $K(v)$=1 then {the group which contains the node v is placed into tabu$_k$;}
Else {the node v is placed into tabu$_k$;}}

3. Repeat until $s \le m$
Set s=s + 1;
For (k=1; k<= l;k++) {
(1) Choose the next node j to be visited according to Eqs. (7) and (8); % state transition rule
(2) Move the ant k to the selected node j;
(3) Update the tabu list according to Eqs. (6);
(4) Update the local pheromone by applying the local updating rule of Eqs. (3);
% local *updating rule*}

4. For (k=1; k<= l; k++) {
(1) Move the ant k from visited$_k(m)$ to visited$_k(1)$;
(2) apply the *local searching* technique *for* the tour k;
(3) Compute the tour length L_k traveled by ant k; }

5. (1) Update the shortest tour found;
(2) Update the global pheromone by applying the global updating rule of Eqs.(9);
(3) ITime = ITime + 1;

6. If (ITime<*TIME_MAX*)
{Empty all visited$_k$; Go to Step 2;}
Else { Print the shortest tour;   Stop.}

5 Computational results

To verify the validity of our proposed methods, we calculate 30 Instances on a PC with 2.1 GHz processor and 2G memory. These instances can be obtained from TSPLIB library [26]. To test GTSP algorithms, Fischetti et al. [6] provided a partition algorithm to convert the instances used in TSP to those that could be used in GTSP. This partition algorithm sets the number of clusters $s = [n/5]$, identifies the s farthest nodes from each other and assigns each remaining node to its nearest center. To test the NGTSP and the performance of the proposed EACS algorithm for solving it, we set exactly $k_i = \lceil |V_i|/5 \rceil$ nodes to be visited from i-th cluster in the shortest closed tour, where $|V_i|$ represents the set cardinality of i-th cluster. Assignment of basic parameter values is $\beta = 2$, $\xi = 0.1$, $\rho_0 = 0.2$, $q_0 = 0.95$, l = 2m, *TIME_MAX* = 100.

5.1 Comparison of Experiment One

In the experiments, the simple extended ACS, and the efficient ACS with local searching and parameter adaptive (EACS) were all performed 60 times on kroc100 in TSPLIB. The results of comparison are given in Fig. 3~5. Fig. 3 is the best tour on kroc100 with ACS, and Fig. 4 is the best tour on kroc100 with EACS. Fig. 5 is the results of comparison on the kroc100 in two methods after performing 60 times. Obviously, the quality and stability of solutions in EACS is better than in ACS.

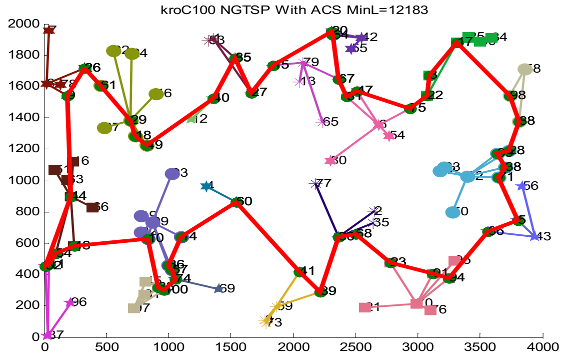

Figure 3. The best tour on kroc100 with ACS

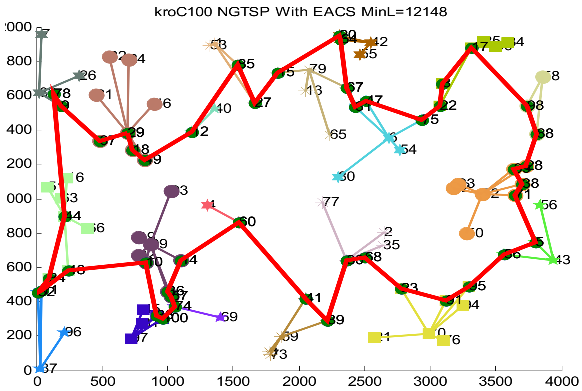

Figure 4. The best tour on kroc100 with EACS

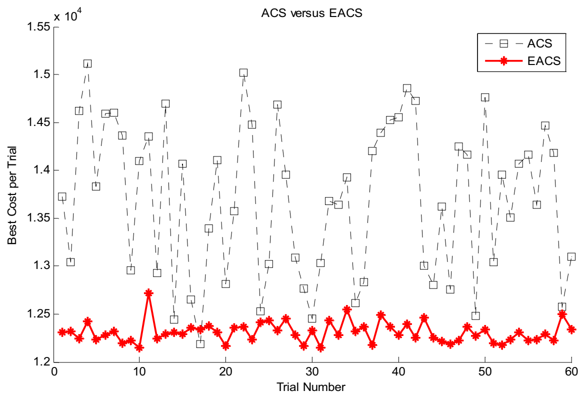

Figure 5. Results of comparison on the kroc100 in two methods

5.2 Comparison of Experiment Two

In the experiments, the simple extended ACS, and the efficient ACS with local searching and parameter adaptive (EACS) were all performed 30 times on 30 instances in TSPLIB. The results of comparison are given in Table 1. The first and the seventh column represent the names of the test instances, the number of clusters (m) and the number of nodes in the closed tour (n). the second and the eighth column represents two kinds of comparison algorithms, and the third to the sixth columns and the ninth to the twelfth columns represent the minimum, maximum, average lengths and standard deviation obtained by the above two methods, respectively.

Table 1 Comparison result of ACS and EACS for 30 benchmark test problems

Problem (*m*/*n*)	Algorithms	Min	Max	Average	Std	Problem	Algorithms	Min	Max	Average	Std
eil76 (16/37)	ACS	313	389	346.17	21.49	U159 (32/76)	ACS	32717	38191	36018.30	1229.74
	EACS	**295**	**322**	**309.13**	**5.27**		**EACS**	**27207**	**30917**	**28120.93**	**858.80**
PR76 (16/38)	ACS	80323	96407	86857.23	6061.46	D198 (40/97)	ACS	13390	15546	15034.10	408.92
	EACS	**77869**	**82213**	**79425.20**	**1161.93**		**EACS**	**12162**	**13116**	**12526.93**	**251.64**
KROA100 (20/46)	ACS	13015	15868	14371.97	729.62	kroa200 (40/97)	ACS	20958	23733	22375.77	617.26
	EACS	**12409**	**13682**	**12781.33**	**266.54**		**EACS**	**17658**	**20718**	**19015.73**	**559.32**
KROD100 (20/48)	ACS	12919	16344	14708.63	1028.53	krob200 (40/96)	ACS	20898	22659	21949.00	525.69
	EACS	**12290**	**13240**	**12721.93**	**230.54**		**EACS**	**16880**	**19827**	**18095.10**	**510.27**
KROE100 (20/49)	ACS	13474	17177	15694.03	954.89	gr202 (41/100)	ACS	389	496	457.47	25.86
	EACS	**13249**	**14143**	**13614.53**	**197.81**		**EACS**	**353**	**379**	**368.43**	**6.46**
RAT99 (20/50)	ACS	746	952	847.97	53.00	ts225 (45/114)	ACS	98759	109823	103503.33	3208.50
	EACS	**713**	**791**	**741.13**	**18.72**		**EACS**	**82834**	**94744**	**89850.13**	**3869.04**
RD100 (20/48)	ACS	4584	5912	5216.53	335.58	tsp225 (45/100)	ACS	2757	3143	2966.57	95.95
	EACS	**4453**	**4770**	**4604.97**	**78.54**		**EACS**	**2282**	**2685**	**2556.70**	**84.50**
eil101 (21/50)	ACS	365	498	451.60	31.76	pr226 (46/113)	ACS	75384	85859	78392.17	2229.28
	EACS	**342**	**389**	**359.07**	**10.17**		**EACS**	**67474**	**69868**	**67941.30**	**677.30**
LIN105 (21/51)	ACS	9909	11536	10806.17	442.82	gr229 (46/113)	ACS	1268	1512	1372.00	67.38
	EACS	**9592**	**9898**	**9700.53**	**72.77**		**EACS**	**1203**	**1266**	**1226.70**	**17.37**
PR107 (22/56)	ACS	32454	35753	33512.00	899.21	gil262 (53/127)	ACS	1646	1923	1796.70	52.15
	EACS	**32397**	**32719**	**32522.97**	**67.61**		**EACS**	**1353**	**1620**	**1557.70**	**47.94**
PR124 (25/59)	ACS	43126	51554	47483.37	2819.33	pr264 (53/129)	ACS	42569	45833	44486.37	867.35
	EACS	**42553**	**43591**	**42883.87**	**200.40**		**EACS**	**35339**	**39629**	**38055.60**	**761.77**
PR144 (29/65)	ACS	53966	57152	55096.37	659.38	a280 (56/131)	ACS	1902	2196	2066.80	72.52
	EACS	**50094**	**52717**	**50757.10**	**787.38**		**EACS**	**1536**	**1847**	**1769.97**	**64.25**
KROA150 (30/74)	ACS	18423	20542	19471.17	554.26	pr299 (60/144)	ACS	37037	42210	39667.17	1317.60
	EACS	**15400**	**16972**	**15886.53**	**403.36**		**EACS**	**30824**	**35397**	**34292.33**	**872.44**
KROB150 (30/72)	ACS	17882	20963	19713.03	709.75	lin318 (64/153)	ACS	32390	36533	34125.07	976.07
	EACS	**15475**	**17314**	**16188.43**	**551.92**		**EACS**	**27583**	**30925**	**30029.70**	**784.60**
PR152 (31/73)	ACS	61287	69750	65057.50	2076.83	rd400 (80/197)	ACS	11331	12605	12027.03	284.44
	EACS	**58339**	**60857**	**59257.17**	**595.64**		**EACS**	**10279**	**10754**	**10547.20**	**93.26**

From Table 1 it can be noticed that in all the 30 instances, all the indicators in EACS are significantly better than in ACS, specifically in minimum value and average. So the developed local searching technique and the parameter adaptive are considerably effective in promoting the stability of solutions and improving the quality of solutions.

6 Conclusions

In order to further expand the application range of the GTSP, we put forward a more general NGTSP problem, the TSP and the GTSP is a special case of the NGTSP respectively. To solve effectively the NGTSP problem, we design an effective ant colony system by extending the classical ACS method. Meanwhile, to speed up the convergence and avoid locking into local minima, a local searching technique and parameter adaptive technique is also introduced into this method. Numerical simulation on 30 TSPlib instances shows that the proposed method can deal with the NGTSP problems fairly effectively. In future we will further improve the quality of solution and try to solve large-scale NGTSP problems in the proposed EACS or other methods.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 10872085), Applied Basic Research Foundation in Science and Technology Department of Sichuan Province (No. 07JY029-125) and Significant Cultivation Project Foundation in Education Department of Sichuan Province (No. 07ZZ016).

References:

[1] Srivastava, S.S., Kumar, S., Garg, R.C., Sen, P.: Generalized traveling salesman problem through n sets of nodes. CORS Journal 7, pp. 97-101 (1969).

[2] Henry-Labordere: A dynamic programming solution of a generalized traveling salesman problem. RAIRO Operations Research B2, pp. 43-49 (1969).

[3] Saskena, J.P.: Mathematical model of

scheduling clients through welfare agencies. Journal of the Canadian Operational Research Society 8, pp. 185-200 (1970).
[4] Noon, C.E., Bean, J.C.: A Lagrangian based approach for the asymmetric generalized traveling salesman problem. Operations Research 39, pp. 623-632 (1991).
[5] Laporte, G., Nobert, Y.: Generalized Traveling Salesman through n sets of nodes: an integer programming approach. INFOR 21(1), pp. 61-75 (1983).
[6] Fischetti, M., Salazar, J.J., and Toth, P.: A branch-and-cut algorithm for the symmetric generalized traveling salesman problem, Operations Research, Vol. 45, pp. 378-394 (1997).
[7] Pop, P.C., Pop, C.S, Zelina, I. and Tascu, I.: Exact algorithms for generalized combinatorial optimization problems, in Proc. COCOA Conference, Xi'an, China, Lecture Notes in Computer Science, Vol. 4616, pp. 154-162 (2007).
[8] Snyder, L.V. and Daskin, M.S. :A random-key genetic algorithm for the generalized traveling salesman problem, European Journal of Operations Research, Vol. 174, pp. 38-53 (2006).
[9] Lien, Y.N., Ma,E., Wah B.W.S.: Transformation of the generalized traveling salesman problem into the standard traveling salesman problem. Information Sciences 74(1-2), pp. 177-189 (1993).
[10] Noon,C., Bean,J.C.: An efficient transformation of the generalized traveling salesman problem. INFOR 31(1), pp. 39-44 (1993).
[11] Dimitrijevic, V., Saric, Z.: An efficient transformation of the generalized traveling salesman problem into the traveling salesman problem on digraphs. Information Science 102 (1-4), pp. 105-110 (1997).
[12] Laporte, G., Semet, F.: Computational evaluation of a transformation procedure for the symmetric generalized traveling salesman problem. INFOR 37(2), pp. 114-120 (1999).
[13] Behzad, A., Modarres, M.: A new efficient transformation of the generalized traveling salesman problem into traveling salesman problem. In: Proceedings of the 15th International Conference of Systems Engineering, pp. 6-8 (2002).
[14] Renaud, J., Boctor, F.F.: An efficient composite heuristic for the Symmetric Generalized Traveling Salesman Problem. European Journal of Operational Research 108(3), pp. 571-584 (1998).
[15] Hu, B., Raidl, G.: Effective neighborhood structures for the generalized traveling salesman problem. In: van Hemert, J., Cotta, C. (eds.) EvoCOP 2008. LNCS, vol. 4972, pp. 36-47. Springer, Heidelberg (2008).
[16] Hu Bin: Effective Neighborhood Structures for the Generalized Traveling Salesman Problem. EvoCOP 2008, LNCS 4972, pp. 36-47, 2008. Springer-Verlag Berlin Heidelberg (2008).
[17] Bontoux, B., Artigues, C., Feillet, D.: A Memetic Algorithm with a Large Neighborhood Crossover Operator for the Generalized Traveling Salesman Problem. Computers & Operations Research (2009).
[18] Gutin, G., Karapetyan, D.: A memetic algorithm for the generalized traveling salesman problem. Natural Computing 9, pp. 47-60 (2010).
[19] Matei O.， Pop P.: An Efficient Genetic Algorithm for Solving the Generalized Traveling Salesman Problem. IEEE, pp. 87-92(2010).
[20] Pop, P.C., Matei, O., and Sabo, C.: A New Approach for Solving the Generalized Traveling Salesman Problem. HM 2010, LNCS 6373, pp. 62-72, 2010.Springer-Verlag Berlin Heidelberg (2010).
[21] Dorigo M, Ganbardella L.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput; 1(1), pp. 53-66(1997).
[22] Yang, J., Shi, X., Marchese, M., Liang, Y.: An ant colony optimization method for generalized TSP problem. Progress in Natural Science 18(11), pp. 1417-1422 (2008).
[23] Pintea, C., Pop, P.C., Chira, C.: Reinforcing Ant Colony System for the Generalized Traveling Salesman Problem. In: Proc. of International Conference Bio-Inspired Computing-Theory and Applications (BIC-TA), Wuhan, China. Evolutionary Computing Section, pp. 245-252 (2006).
[24] Dortgo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation, 1997, 1(l):53-66.
[25] Voudouris, C., Tsang, E.: Guided local search and its application to the traveling salesman problem. Eur J Oper Res; 113, pp. 469-99 (1999).
[26] Reinelt, G.: TSPLIB – a traveling salesman problem library. ORSA J Comput 3, pp. 376-84 (1991).