A Memetic-Clustering-Based Evolution Strategy for Traveling Salesman Problems

Yuping Wang[1] and Jinhua Qin[2]

[1] School of Computer Science and Technology,
Xidian University, Xi'an, China
ywang@xidian.edu.cn
[2] Department of Mathematics Science,
Xidian University, Xi'an, China
abcxyz-999@163.com

Abstract. A new evolution strategy based on clustering and local search scheme is proposed for some kind of large-scale travelling salesman problems in this paper. First, the problem is divided into several subproblems with smaller sizes by clustering, then the optimal or the approximate optimal tour for each subproblem is searched by a local search technique. Moreover, these tours obtained for the subproblems are properly connected to form a feasible tour based on a specifically-designed connection scheme. Furthermore, a new mutation operator is designed and used to improve each connected feasible tour further. The global convergence of the proposed algorithm is proved. At last, the simulations are made for several problems and the results indicate the proposed algorithm is effective.

Keywords: TSP, evolutionary algorithm, clustering, memetic algorithm.

1 Introduction

The traveling salesman problem (TSP) is one of the most famous combinatorial optimization problems. Given n cities and a distance matrix $D = [d_{ij}]$, where d_{ij} is the distance between city i and city j, TSP requires finding a tour (i.e., a permutation of cities) through all of the cities, visiting each exactly once, and returning to the originating city such that the total distance traveled is minimized.

In this paper we consider the two-dimensional (2-D) Euclidean TSP, in which the cities lie in R^2 and the distance between two cities is calculated by Euclidean distance. The 2-D Euclidean TSP is known to be NP-hard ([1],[2],[3]). It was proven that there is no polynomial-time approximation scheme for TSP unless $P = NP$ ([4]). However, the TSP and its variants have a diverse practical applications. More than 1700 related papers have been published during the past five years (see the INSPEC database: http://www.inspec.org). They are one of the most actively studied topics in the evolutionary computation community, too. Many papers have published in this field (e.g., [5]$\sim$ [10]). One of the most successful evolutionary algorithms (EAs) for TSP is the hybrid EAs (or memetic

J.T. Yao et al. (Eds.): RSKT 2007, LNAI 4481, pp. 260–266, 2007.

EAs)that incorporate local search scheme into EAs (e.g., [7],[8],[9]). To the best of our knowledge, Freisleben and Merz's hybrid EA ([8],[9]) with powerful Lin-Kernighhan (LK) local search algorithm ([10]) is in practice one of the best EAs for TSP among the published algorithms. However, LK local search algorithm needs a lot of computation and can not be applied to general large scale problems. For some special kind of TSP problems, it is possible to design effective evolutionary algorithms.

In this paper, we focus our attention on some special TSP problems, i.e., the problems in which the cities can be classified into several groups, and in each group the cities crowd together, and try to design an effective evolutionary algorithm. To do so, we first divided cities into several groups by using clustering technique. Second, we look for the optimal or approximate optimal tour for each group by a local search technique. Third, we connect the tours found for these groups to form a feasible tour based on a specifically-designed connection scheme. Fourth, we design a new mutation operator and use it to improve the feasible tour. Based on these, a novel evolution strategy based on clustering and local search is proposed for this kind of TSP problems. At last, the simulations are made and the results demonstarte the effectiveness of the proposed algorithm.

2 Classification of All Cities into Several Groups by Clustering

For traveling salesman problems in which the cities can be classified into several groups, and in each group the cities crowd together, it seems that the salesman should go through all cities in best way in one group, then move to some other group and also go through all cities in this group in optimal way. Repeat this process until he goes through all groups and returns the starting city. Based on this idea, we have to classify all cities into several groups. In this paper we use the following clustering algorithm: K-mean clustering ([11]).

Algorithm 1

1. Randomly choose k cities as the initial centers of k clusters, where k is a parameter. Let $t = 0$.
2. Calculate the distance between each city and each center of the clusters. Classify the city into a cluster in which the distance between the city and the center of the cluster is the shortest.
3. Re-compute the center for each cluster. Let $t = t + 1$.
4. If the center for every cluster is the same as the center for this cluster in previous iteration $t - 1$, then stop; otherwise, go to step 2.

3 Local Search

For each cluster, a local search algorithm, $2-opt$ algorithm, is used to search high quality tour. To explain the idea of $2 - opt$ algorithm clearly, we first introduce the following definitions.

Definition 1. *For a given tour P, if λ links (edges) of tour P are replaced by other λ links and the resulted graph is also a feasible tour, then the operation of exchanging λ links is called a λ-exchange, and the resulted tour is called a λ-exchange tour.*

Definition 2. *A tour is said to be λ-optimal (or simply $\lambda-opt$) if it is impossible to obtain a shorter tour by replacing any λ of its links (edges) by any other set of λ links.*

The following is a local search algorithm for each cluster of cities.

Algorithm 2 (Local search heuristics based on λ-exchange)

1. Set $L_{opt} = M$, where M is a large enough positive number and $P_{opt} = \emptyset$. Let $t1 = 0$.
2. Arbitrarily take a feasible tour P.
3. Check whether there is a λ-exchange tour $P_{\lambda-ex}$ which is better than current tour P. If there is such a tour, let $P = P_{\lambda-ex}$. Go to step 3; otherwise, go to step 4.
4. Tour P is the λ-optimal tour for the initial tour, let $P_1 = P$. Let $L1$ is the length of tour P_1.
5. If $t1 < 2$, go to step 6; otherwise, P_{opt} is an approximate optimal tour.
6. If $L1 < L_{opt}$, let $L_{opt} = L1$, $P_{opt} = P_1$, go to step 2; otherwise, let $t1 = t1+1$, go to step 2.

In our simulation we take $\lambda = 2$. Note that for a given tour there are total $\frac{n(n-3)}{2}$ λ-exchange tours.

4 Scheme for Connecting Clusters

In the previous two sections, all cities of a special kind of TSP problems are divided into several clusters, and for the cities contained in each cluster an optimal or approximate optimal tour is found. To get a high quality tour for all cities it is necessary to design an effective connection scheme to connect all these tours. Our motivation is that the connection is made in such a way that the connected tour is as short as possible. The detail is as follows.

Algorithm 3 (Connection Scheme)

1. Let k denotes the number of clusters and let Ł $= 2$.
2. Choose two clusters whose centers are the nearest. Their corresponding tours are denoted as $T1$ and $T2$, respectively.
3. Choose an edge $A_1B_1 \in T1$ and an edge $A_2B_2 \in T2$ satisfying

   $$|A_1A_2|+|B_1B_2|-|A_1B_1|-|A_2B_2| = \min\{|C_1C_2|+|D_1D_2|-|C_1D_1|-|C_2D_2|\},$$

   where edge $C_1D_1 \in T1$ and edge $C_2D_2 \in T2$, respectively, and $|A_1A_2|$ denotes the length of edge A_1A_2.

4. Connect A_1 and A_2 (i.e., add edge A_1A_2), and B_1 and B_2. Remove edges A_1B_1 and A_2B_2, respectively. Then a new tour T through all cities of two chosen clusters is formed by connecting $T1$ and $T2$, and a new cluster is formed by the union of these two chosen clusters.
5. If Ł $< k$, then let Ł $=$ Ł $+ 1$, go to step 2; otherwise tour T is a feasible tour through all cities. Stop.

5 Mutation Operator

To further improve the quality of the tours obtained by section 4, we design a mutation operator and use it to each tour obtained. The detail is as follows.

Algorithm 4 (Mutation Operator)

1. For each tour $i_1i_2\cdots i_n$ obtained by section 4, where $i_1i_2\cdots i_n$ is a permutation of $1, 2, \cdots, n$.
2. For $q = 1, 2, \cdots, n$, randomly change i_q into any element, denoted by j_q, in $\{1, 2, \cdots, n\}/\{j_1, j_2, \cdots, j_{q-1}\}$ with equal probability, where

$$\{1, 2, \cdots, n\}/\{j_1, j_2, \cdots, j_{q-1}\}$$

   represents a set whose elements belong to set $\{1, 2, \cdots, n\}$ but do not to set $\{j_1, j_2, \cdots, j_{q-1}\}$.
3. Tour $j_1j_2\cdots j_n$ is the offspring of tour $i_1i_2\cdots i_n$.

It can be seen from this mutation operator that, for any feasible tours $i_1i_2\cdots i_n$ and $j_1j_2\cdots j_n$, the probability of generating $j_1j_2\cdots j_n$ via $i_1i_2\cdots i_n$ by mutation operator is $\frac{1}{n} > 0$.

6 The Proposed Algorithm

Based on algorithms in the previous four sections, the proposed evolution strategy can be described as follows:

Algorithm 5 (A Memetic-Clustering-Based Evolution Strategy)

1. (Initialization) Given population size N, maximum generations g_{max}, and N positive integers $k_1, k_2, \cdots, k_N$. For each k_i for $i = 1, 2, \cdots, N$, do the following:
   - Generate k_i clusters by algorithm 1, and generate a tour for each cluster by using Algorithm 2.
   - Connect all these tours for clusters into one feasible tour for all cities by using Algorithm 3.

   All these N feasible tours constitute the initial population $P(0)$, Let $t = 0$.

2. (Mutation)For each individual

$$T_r = i_1 i_2 \cdots i_n \in P(t),$$

generate an offspring

$$O_r = j_1 j_2 \cdots j_n,$$

$r = 1, 2, \cdots, N$. The set of all offspring is denoted as $O(t)$.
3. (Selection) Select best N individuals from $P(t) \cup O(t)$ as the next generation population $P(t+1)$.
4. (Stop Criterions) If $t > g_{max}$ and the best tour obtained can not improved in successive 10 generations, then stop; otherwise, let $t = t+1$, go to step 2.

7 Global Convergence

First, we introduce the concept of the global convergence as follows.

Definition 3. *Let $a^* \in A$ denote the chromosome which corresponds to an optimal tour. If*

$$prob\{\lim_{t\to\infty} a^* \in P(t)\} = 1,$$

then the proposed genetic algorithm is called to converge to the global optimal solution with probability one, where $prob\{\}$ represents the probability of random event $\{\}$.

For any feasible tours $i_1 i_2 \cdots i_n$ and $j_1 j_2 \cdots j_n$, note that

$$prob\{M(i_1 i_2 \cdots i_n) = j_1 j_2 \cdots j_n\} = \frac{1}{n} > 0,$$

and the best tour found so far will be kept by the selection process, where $M(i_1 i_2 \cdots i_n)$ represents the offspring of $i_1 i_2 \cdots i_n$ by mutation. It can be proved by making use of the conclusions in [12] that the proposed algorithm has the following property of the global convergence.

Theorem 1. *The proposed evolution strategy (Algorithm 5) converges to the global optimal solution with probability one.*

8 Simulations

8.1 Test Problems

In the simulations, We execute the proposed algorithm to solve six standard benchmark problems: $nrw1379$, a 1379-city problem, $rL1889$, a 1889-city problem $u2319$, a 2319-city problem, $pr2392$, a 2392-city problem, $pcb3038$, a 3038-city problem, and $rL5915$, a 5915-city problem. These problems are available from TSPLIB at http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html

8.2 Parameter Values

We adopt the following parameter values: $N = 20$, $k_1 = \cdots = k_N = s$, and $s = 5, 8, 10$ respectively for problems with fewer than 3000 cities, and $s = 5, 8, 10$ and 15 respectively for problems with more than 3000 cities. $g_{max} = 300$ for problems with fewer than 3000 cities and 500 for problems with more than 3000 cities.

8.3 Results

The simulations are carried out on a AthlonXP1800+512M PC and we program the proposed algorithm in Matlab language. For each benchmark problem, we perform 20 independent executions. We record the following data:

- The best, the average and the worst lengths over 20 runs, denote them by Best, Mean and Worst, respectively.
- Average CPU time over 20 runs, denoted by CPU.
- Percentage of the amount the best tour found surpasses the optimal tour over the optimal tour, denoted by %.

Table 1. Results obtained by proposed algorithm, where $Opt - tour$ represents the length of the optimal tour, and K is the number of clusters used

TSP	Opt-tour	K	Best	Mean	Worst	CPU	%
nrw1379	56638	5	59770	60236	61069	65.3	5.5
		8	59922	60177	60353	57.6	5.8
		10	59981	60389	60873	65.6	5.9
rL1889	316536	5	342447	347407	353226	147.7	8.2
		8	344107	350777	358637	135.4	8.7
		10	344513	349777	355378	130.5	8.8
u2319	234256	5	243379	243997	244465	227.7	3.9
		8	243167	243843	245016	216.1	3.8
		10	242677	243243	243829	209.8	3.6
pr2392	378032	5	389288	394309	396554	246.8	2.9
		8	394863	398895	404705	232.1	4.4
		10	398937	402243	405692	225.6	5.5
pcb3038	137694	5	148031	149461	150881	433.0	7.5
		8	149209	150126	151309	410.4	8.3
		10	148867	149681	150905	408.7	8.1
		15	148326	149853	151286	398.1	7.7
rL5915	565530	5	625686	629688	638488	2229.8	10.6
		8	613469	625109	635527	2040.0	8.5
		10	617823	627439	637816	1828.7	9.2
		15	624123	633052	655479	1965.5	10.3

The results are given in Table 1. It can be seen from Table 1 that the percentage of the amount the best tour found surpasses the optimal tour over the optimal tour is relatively small although the proposed algorithm has not found the optimal tours for these problems. This indicates the proposed algorithm is effective.

9 Conclusions

In this paper we deal with some special TSP problems, i.e., the problems in which the cities can be classified into several groups, and within each group the cities crowd together. We designed an effective and globally convergent evolutionary algorithm for this kind of problems. The proposed algorithm has the ability of finding close-to-optimal solutions with high speed and a relatively small amount of computation. The simulation results demonstrate the effectiveness of the proposed algorithm.

Acknowledgements. The authors wold like to thank the anonymous reviewers for their insightful comments and helpful suggestions that have improved the paper greatly. This work was supported by National Natural Science Foundation of China (No. 60374063).

References

1. S.Jung, B.R.Moon.: Toward minimal restriction of renetic coding and crossovers for the 2-D eunclidean TSP. *IEEE Trans. on Evolutionary Compuation* 6(6) (2002) 557-565.
2. R.Baraglia, J.I.Hidalgo, R.Perego.: A hybrid Hheuristic for the traveling salesman problem. *IEEE Trans.on Evolutionary Compuation* 5(6) (2001) 613-622.
3. M.R.Garey, R.L.Graham, D.S.Johnson.: Some NP-complete geometric problems. In *Proc. 8th Annu. ACM Symp. Theory of Computing*, New York, NY, USA, ACM Press (1976) 10-22
4. S.Arora, C.Lund, R.Motwani, et al.: Proof verification and intractability of approximation problems. In *Proc. 33th IEEE Symp. Foundations of Computer Science*, Pittsburgh, Pennsylvania, USA, IEEE Press (1992) 3-22.
5. K.Katayama, H.Narihisa.: Iterated local search approach using genetic trasformation to the traveling salesman problem. In *Proc. Genetic and Evolutionary computation Conf.* New York, USA, Morgan Kaufmann (1999) 321-328.
6. Y.Nagata, S.Kobayashi.: Edge assembly crossover: A high-power genetic algorithm for the travrling salesman problem. In *Proc. 7th Int. Conf. Genetic Algorithm*, San Mateo, CA: USA, Morgan Kaufmann (1997) 450-457.
7. D.S.Johnson.: Local optimization and the traveling salesman problem. In *Proc. 17th Automata, Languages, and Programming*, Warwick University, England, Springer-Verlag (1990) 446-461.
8. B.Freisleben, P.Merz.: New genetic local search operators for the traveling salesman problem. In *Proc. 4th Conf. Parallel Problem solving fron Nature*, Berlin, Germany, Springer-Verlag (1996) 616-621.
9. P.Merz, B.Freisleben.: Genetic local search for the TSP:New results. In *Proc. of the 1997 International Conferenece on Evolutionary Computation*, Piscataway,NJ:IEEE Press (1997) 159-163.
10. S.Lin, B.W.Kernighan.: An effective heuristic algorithm for the traveling salesman problem. *Operations Research* 21(2) (1973) 495-516
11. A.K. Jain, M.N. Murty, P.J. Flyn.: Data clustering: A review. *ACM Computing Survey* 31(3) (1999) 264-323.
12. T. Bäck.: *Evolutionay algorithms in theory and practice.* New York: Oxford Univ. Press (1996).