TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.6, June 2014, pp. 4413 ~ 4418
DOI: 10.11591/telkomnika.v12i6.5476

An Improved NSGA-II Algorithm for Multi-objective Traveling Salesman Problem

Yabo Luo*[1,2], Min Liu[3], Zhongxiao Hao[4], Dongbo Liu[1]
[1]Department of Computer and Communication, Hunan Institute of Engineering,
Xiangtan 411104, China
[2]College of Transportation and Logistics, Central South University of Forestry and Technology,
Changsha 410004, China
[3]School of Computer Science and Engineering, Hunan University of Science and Technology,
Xiangtan 411201, China
[4]School of Information Science and Electrical Engineering, Hebei University of Engineering,
Handan 056002, China
*Corresponding author, e-mail: luoyabo@foxmail.com

Abstract

Multi-objective traveling salesman problem (MOTSP) is an extended instance of traveling salesman problem (TSP), which is a well-known NP hard problem. In this paper, an improved NSGA-II algorithm (denoted by INSGA-II-MOTSP) is proposed to solve the MOTSP. Specifically, a layer strategy according to need is proposed to avoid generating unnecessary non-dominated fronts. The arena's principle is also adopted to construct non-dominated set, so as to reduce the count of dominance comparison. In addition, an order crossover like operator and an inversion mutation operator are adopted for MOTSP. The experiment results show that the proposed INSGA-II-MOTSP algorithm is able to find better spread of solutions compared to other three algorithms.

Keywords: *multi-objective evolutionary algorithm, layer strategy according to need, arena's principle, multi-objective traveling salesman problem*

1. Introduction

Multi-objective problems (MOPs) arise in a natural fashion in most disciplines and their solution has been a challenge to researchers for a long time [1]. The presence of multiple objectives in a problem, in principle, gives rise to a set of optimal solutions (largely known as Pareto-optimal solutions), instead of a single optimal solution. In the absence of any further information, one of these Pareto-optimal solutions cannot be said to be better than the other. This demands a user to find as many Pareto-optimal solutions as possible. The use of evolutionary algorithms (EAs) to solve MOPs has been motivated mainly because of the population-based nature of EAs which allows the generation of several elements of the Pareto optimal set in a single run. To date there have been many evolutionary multi-objective optimization (EMO) algorithms proposed for multi-objective optimization problems. Among of them, the Non-dominated sorting genetic algorithm II (NSGA-II) [2] is well-known and is being most widely used.

Traveling Salesman Problem (TSP) is a classic NP hard problem in combination optimization domain. Given a collection of cities and the cost of travel between each pair of them, the TSP is to find the cheapest way of visiting all of the cities and returning to starting point. More formally, it can be represented by a complete weight graph, $G = (N,E)$ with N being the set of cities and E the set of edges fully connecting the nodes N. Each edge is assigned a value d_{ij}, which is the length of edge $e_{ij} \in E$. The TSP is problem of finding a minimal length Hamiltonian circuit of the graph, where a Hamiltonian circuit is a closed tour visiting exactly once each of the $n = |N|$ cities of G.

Much of the work on the TSP is 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. The TSP naturally arises as a sub problem in many transportation and logistics applications, for example the problem of arranging school bus routes to pick up the children in a school district.

Received December 22, 2013; Revised February 14, 2014; Accepted February 27, 2014

More recent applications involve the delivery of meals to homebound persons and the routing of trucks for parcel post pickup. Therefore, in recent years, many intelligent optimization algorithms, such as evolutionary algorithm (EA) [3], ant colony algorithm (ACO) [4-5], particle swarm optimization (PSO) algorithm [6] and simulated annealing (SA) algorithm [7] and so on, were proposed to solve the TSP problem.

Multi-objective traveling salesman problem (MOTSP) is an extended instance of TSP. Given a complete, weighted graph $G = (N,E,c)$ with being the set of N nodes, E being the set of edges fully connecting the nodes, and c being a function that assigns to each edge $(i, j) \in E$ a vector $\left(c_{ij}^{1},\ldots,c_{ij}^{k}\right)$, where each element c_{ij}^{k} corresponds to a certain measure like distance, cost, etc. between nodes i and j. For the following we assume that $c_{ij}^{k} = c_{ji}^{k}$ for all pairs of nodes i, j and objectives k, that is, we consider only symmetric problems. The MOTSP is also the problem of finding "minimal" Hamiltonian circuits of the graph, that is, a set of closed tours visiting each of the $n = |N|$ nodes of G exactly once; here "minimal" refers to the notion of Pareto optimality.

A number of studies were conducted in the multi-objective optimization literature. In [8], a dynamic multi-objective particle swarm optimizer (PSO), maximinPSOD, which is self-adaptive and capable of handling dynamic multi-objective optimization problems, was proposed. In [9], Goh and Tan suggested a new dynamic multi-objective coevolutionary algorithm that hybridizes competitive and cooperative mechanisms.

Recently, some researchers have designed EMO algorithms to deal with MOTSP in [10-15]. In this paper, an improved NSGA-II algorithm (denoted by INSGA-II-MOTSP) is proposed to solve the MOTSP. To improve the run-time efficiency of NSGA-II, a layering strategy according to need is proposed to avoid generating unnecessary non-dominated fronts, and the arena's principle is also adopted to construct non-dominated set. In addition, an order crossover like operator and an inversion mutation operator are designed for the MOTSP. The experiment results show that the proposed INSGA-II-MOTSP algorithm is able to find better spread of solutions compared to other three algorithms.

2. Related Work

First, assume the size of set P is n, each individual in P has r attributes, and $f_k(\ )$ is an evaluation function (k=1,2,…,r).

Definition 1 (Pareto Dominance): $\forall x$, $y \in P$, if $f_k(x) \le f_k(y)$, (k=1,2,…,r), and $\exists l \in \{1,2,\cdots,r\}$ such that $f_l(x) < f_l(y)$, then x dominates y (denoted $x \succ y$). Here '$\succ$' represents dominance relationship, and y is a dominated individual.

Definition 2 (Non-dominated set): $\forall x \in P$, if $\neg\exists y \in P$, $x \succ y$, then x is a non-dominated individual of P. The set consisting of all the non-dominated individuals of P is called the non-dominated set of P.

One popular way of designing EMO algorithm is fist to construct non-dominated set, and then make selection, crossover, and mutation on the population set P to generate the next generation. So constructing the non-dominated set is an very important step for EMO algorithms to find the final Pareto optional solutions. In the NSGA-II algorithm, a non-dominated sorting is used to sort a population into different non-dominated levels, and this sorting procedure's time complexity is $O(rN^2)$. In next section, an improved NSGA-II algorithm is proposed to solve the MOTSP problem. To reduce the run-time of the non-dominated sorting of the NSGA-II, an improved non-dominated sorting is proposed by using a layer strategy according to need and an arena's principle.

3. INSGA-II-MOTSP Algorithm

The main frame of the INSGA-II-MOTSP algorithm is as follows:

Algorithm INSGA-II-MOTSP

(1) Read the input data (the demand amount, location of the city, the distance and cost between cities and so on);

(2) Initialize the population;

(3) Assign 1 to gen

a. Evaluate each individual's fitness according to the fitness function;

b. Generate a children population by genetic operators of NSGA-II (i.e., selection, crossover, mutation);

c. Combine the parent population and the children population to form a combined population;

d. An improved non-dominated sorting (a layer strategy according to need and the arena's principle) is performed on the combined population to sort the combined population into different non-dominated fronts; If the total number of individuals in the fronts has exceeded the size of a single population, then the non-dominated sorting process will stop.

e. Execute a truncation procedure on the non-dominated fronts to make new population of the next generation;

f. gen++, IF $gen \le MAXGEN$, GOTO a;

3.1. Population Initialization and Individual Evaluation

This paper adopts nature number coding for the expression of chromosome in the population. Every gene of the chromosome represent a city and the sequence between the genes reflects the trip, for example the code is 0 1 2 3 4 5 0, which express that the trip go through the number 0 city to number 1 then 2, ... at last return the original city.

After initialize population, every chromosome represents a random arranged of city's natural number. Suppose a coding of chromosome is $a_0a_1...a_{n-1}a_0$ for the bi-objective TSP, the two evaluative fitness functions are defined as follows:

The length of the total tour: Length = dist(a_0,a_1)+dist(a_1,a_2)+…+dist(a_{n-1},a_0). Here, dist(a_i,a_j) means the distance between city $a_{i,}$ and city $a_{j.}$

The total cost: Costs = cost(a_0,a_1)+cost(a_1,a_2)+…+cost(a_{n-1},a_0). Here, cost(a_i, a_j) means the cost between city $a_{i,}$ and city $a_{j.}$

3.2. Genetic Operators

The binary championships selection [1] is utilized as the selection operator.

The order crossover (OX) and inversion mutation [3] is used to generate the children individual by choosing a part of the traveling route and changing over the corresponding genetic sequence.

3.3. Arena's Principle

This paper adopts the arena's principle to construct the non-dominated set of the population. The arena's principle is given as follows: Assume *P* is a evolutional population, *Q* is a construction set, Set *Q*=*P*. Assume *Nds* is a non-dominated set and let *Nds* be a empty set. Fist, select an individual *x* from the *Q* to be champion, then compare it with all the others individuals in *Q* one by one. If *x* dominates *y*, then eliminate *y* from *Q*, else eliminate *x* and set *y* to be *x* (the champion) and go on. After the first round comparison, move *x* into *Nds*. Next, repeat the above procedure until *Q* is empty.

When compare the individual of *Q* with the champion, if it is dominated by the champion, it will be eliminated and will not go to the next round. More number of individual in the *P*, Less comparison count of the arena's principle. In [16], the time complexity of the arena's principle is analyzed to be $O(rmN)$, here *r* is the number of objects, *m* is the amount of non-dominated individual, *N* is the size of population. In usual, there exists $m<N$, Therefore, the arena's principle has lower time complexity than that (i.e., $O(rN^2)$) of NSGA-II.

3.4. A Layer Strategy According to Need

The NSGA-II algorithm divides the combination population R_t (combined with parent population P_t and children population Q_t) into multi-layer non-dominated fronts *F* ($F = F_1 \bigcup F_2 \bigcup \ldots \bigcup F_e$, *e* is the number of non-dominated layers). Firstly, use the same way like constructing the non-dominated set to get the non-dominated set R_t, which called the first non-dominated front F_1. Then, eliminate the individuals of F_1 from population R_t, i.e., set $R_t = R_t \backslash F_1$. Repeating the above procedure, and obtain $F_{2,}$ F_3, $F_{4\ ,}$ and so on, until R_t is empty. The procedure of the layering of NSGA-II is described as Figure 1. After the process of layering, the next process truncates the double size of non-dominated set *F* into a half, to obtain the new population of next generation. It eliminates the individuals of the second part of *F* and the first part is remained to be the new population P_{t+1}.

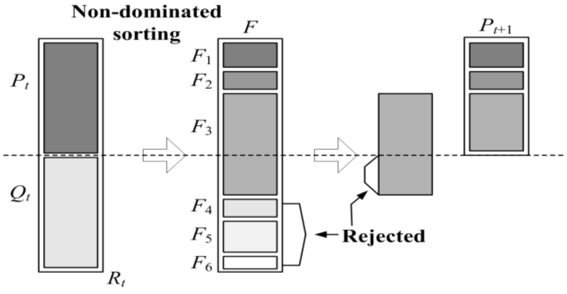

Figure 1. Procedure of NSGA-II

However, the way of layering above is not good enough, just like Figure 1, after layering of F_3, the total individuals of F_1, F_2, F_3 have exceeded the size of a single population. So the layering of F_4, F_5, F_6 is unnecessary. Therefore, in this paper, a layering strategy according to need is proposed, and it can be described as the Figure 2. When the number of total individuals is larger than the size of population, the layering procedure will stop. Hence, it is better than the layer method of NSGA-II, for its less number of layering fronts and at the same time still having the same population P_{t+1} [17].

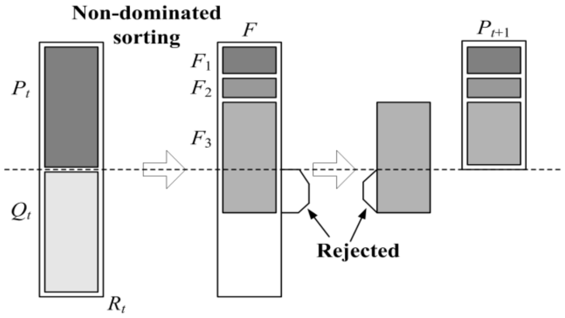

Figure 2. Improved Procedure of NSGA-II

4. Experiment Results and Analysis

In this paper, we have adopted the case in [9] as the test problem of the MOTSP. As Figure 3 shows, D1 is the length between each city, and D2 is the cost between each city. In the experiment, the computer is the PC of P4-1.7G CPU, 1024M internal storage, Windows XP, and programming VC++ 6.0 programming platform. The parameter setting is set as follows: size of population *N*=50, the max of generation Maxgen=5, the crossover ratio *Pc*=0.9 and mutation ratio *Pm*=0.08.

$$D1 = \begin{pmatrix} 0 & 81 & 72 & 55 & 81 & 3 \\ 81 & 0 & 3 & 44 & 9 & 40 \\ 72 & 3 & 0 & 87 & 77 & 21 \\ 55 & 44 & 87 & 0 & 67 & 25 \\ 81 & 9 & 77 & 67 & 0 & 93 \\ 3 & 40 & 21 & 25 & 93 & 0 \end{pmatrix} \quad D2 = \begin{pmatrix} 0 & 82 & 14 & 14 & 43 & 47 \\ 82 & 0 & 61 & 76 & 29 & 47 \\ 14 & 61 & 0 & 29 & 31 & 51 \\ 14 & 76 & 29 & 0 & 78 & 67 \\ 43 & 29 & 31 & 78 & 0 & 28 \\ 47 & 47 & 51 & 67 & 28 & 0 \end{pmatrix}$$

Figure 3. The Length and Cost Matrix between the Cities

In this test problem, both the NSGA-II-MOTSP and the improved NSGA-II algorithm (INSGA-II-MOTSP) are used to solve MOTSP. After repeat 5 times experiments, the average of data is adopted. The results of the two algorithms are expressed in Table 1, and the deviation is economizing data ratio of INSGA-II-MOTSP than the NSGA-II.

Table 1. Results of NSGA-II-MOTSP and INSGA-II-MOTSP

	NSGA-II-MOTSP	INSGA-II-MOTSP	deviation
Average count of dominance comparison	2569.1	1955.7	-23.9%
Average running time (s)	0.051	0.035	-31.3%

From Table 1, INSGA-II-MOTSP is better than NSGA-II-MOTSP both on the average count of dominance comparison and average running time. The main reason is that arena's principle is adopted. The time complexity of arena's principle is $O(rmN)$ which is better than $O(rN^2)$ of NSGA-II. In addition, the layering strategy as need reduces the number of layering. In short, the experiment result shows that there are better efficiency and solutions in the new algorithm.

Table 2 is the solutions found by INSGA-II-MOTSP and other three intelligence algorithms [9]. From this table, it can be seen that each solution has two entities: (Leng, Cost). For example, the solution (158, 280), its corresponding tour line is 0-5-2-1-4-3-0, here, 158 present the total length of the tour and 280 is the total cost. From the results of Table 2, the INSGA-II-MOTSP is not only able to obtain the same solution of other algorithm but also can find much more Pareto solutions like (271, 197), (194, 265). Therefore, the INSGA-II-MOTSP is able to find much better spread of solutions and better convergence near the true Pareto-optimal front compared to other three algorithms.

Table 2. Solutions Found by INSGA-II-MOTSP and other Three Algorithms

ACO	SA	2-opt algorithm	INSGA-II-MOTSP
(158, 280)	(158, 280)	(158, 280)	**(271, 197)**
(250, 208)	(250, 208)		(250, 208)
	(209, 248)		(209, 248)
			(194, 265)
			(158, 280)

5. Conclusion

In our really life, there are many problems which are composed with many conflicting multiple optimization objects. MOTSP is an extended instance of traveling salesman problem, which naturally arises as a subproblem in many transportation and logistics applications. In this paper, an improved NSGA-II algorithm (INSGA-II-MOTSP) is proposed to solve the MOTSP. To improve its run-time efficiency, a layering strategy according to need is proposed and arena's principle is also adopted to construct non-dominated set. In addition, an order crossover like operator and an inversion mutation operator are adopted for MOTSP. The experiment results show that the proposed INSGA-II-MOTSP algorithm is able to find better spread of solutions compared to other three algorithms. The better results and low computation time obtained by this algorithm can be explained by the layer strategy according to need. We intend to extend this strategy to other multi-objective optimization problems.

In addition, we would like to use other measures which allows for a sound statistical analysis of our results. To this goal we will adopt the attainment functions methodology [18] for experimental analysis.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under Grant No. 51177040. The authors would like to thank Dr. Xilong Qu for the discussions about the measure code. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] CAC Coello, GB Lamont, DA Van Veldhuizen. Evolutionary Algorithms for Solving Multi-Objective Problems. 2nd ed. New York: Springer-Verlag. 2007.

[2] K Deb, A Pratap, S Agarwal, T Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 2002; 6(2): 182-197.

[3] DB Fogel. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Third Edition ed. New Jersey, USA: John Wiley & Sons, Inc., 2006.

[4] M Dorigo, LM Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation.* 1997; 1(1): 53-66.

[5] C García-Martínez, O Cordón, F Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*. 2007; 180(1): 116-148.

[6] Y del Valle, GK Venayagamoorthy, S Mohagheghi, JC Hernandez, RG Harley. Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation.* 2008; 12(2): 171-195.

[7] KI Smith, RM Everson, JE Fieldsend, C Murphy, R Misra. Dominance-based multiobjective simulated annealing. *IEEE Transactions on Evolutionary Computation*. 2008; 12(3): 323-342.

[8] XD Li, J Branke, M Kirley. *On performance metrics and particle swarm methods for dynamic multiobjective optimization problems*. Proceeding of 2007 IEEE Congress on Evolutionary Computation, Singapore. 2007; 576-583.

[9] CK Goh, KC Tan. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation.* 2009; 13(1): 103-127.

[10] Min Liu, Wenhua Zeng. *A fast Evolutionary Algorithm for Dynamic Bi-objective Optimization Problems*. The 7th International Conference on Computer Science & Education (ICCSE 2012). 2012; 130-134.

[11] D Angus. *Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem.* Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007), 2007; 333-340.

[12] L Paquete, T Stutzle. *A two-phase local search for the biobjective traveling salesman problem*. EMO 2003, 2003; 479–493.

[13] L Ma, F Jiang. Solving multi-criteria traveling salesman problem by ant algorithm. *Systems Engineering Theory Methodology Applications.* 1999; 8(4): 23-27.

[14] Sandip Chanda, Abhinandan De. Congestion Relief of Contingent Power Network with Evolutionary Optimization Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(1): 1-8.

[15] Xuesong Yan, Qinghua Wu, Hammin Liu. An Improved Robot Path Planning Algorithm Based on Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(8): 1948-1955.

[16] JH Zheng, H Jiang, D Kuang, ZZ Shi. An approach of constructing multi-objective pareto optimal solutions using arena’s principle. *Journal of Software*. 2007; 18(6): 1287-1297.

[17] Min Liu, WH Zeng, JF Zhao. A fast bi-objective non-dominated sorting algorithm. *Pattern Recognition and Artificial Intelligence.* 2011; 24(4): 538-547.

[18] VG da Fonseca, C Fonseca, A Hall. *Inferential performance assessment of stochastic optimisers and the attainment function.* Evolutionary Multi-Criterion Optimization (EMO’ 2001), LNCS 1993. 2001; 213-225.