The Differential Evolution Algorithm for the Travelling Salesman Problem with Time Windows

Timur Keskinturk

Department of Quantitative Methods, School of Business, Istanbul University, Avcilar, Istanbul 34180 Turkey

tkturk@istanbul.edu.tr

Keywords: Differential evolution algorithm, Travelling salesman problem, Optimization, Routing, Meta-heuristic.

Abstract. In this paper we study the differential evolution algorithm meta-heuristic to solve the travelling salesman problem with time windows. To create an initial population of the problem we use time windows. For every costumer we generate a random number in those time intervals. These time variables represent the visiting time of the related costumer. Our results for the benchmark problems show that with this initial population process and differential evolution algorithm that it could be a good alternative for the travelling salesman problem with time windows. In addition to describing the method, the experimental results of the benchmark test problems are presented.

1. Introduction

The Travelling Salesman Problem (TSP) is one of the most studied combinatorial optimization problems. An extended version of the TSP with time windows is the travelling salesman problem with time windows (TSPTW). The problem deals with delivery requests that should be served by a salesman. The vehicle's journey should start and end at a central depot. The vehicle does not have a limited capacity and could also include a capacity constraint. In addition, every request should be served within a predetermined time window interval. If the sales person arrives earlier than the allowed service time, she/he should wait until the beginning of the specified period. An objective of the problem is to minimize the total travel cost that could be distance, time or fuel.

The TSPTW problem is the sub-problem of the vehicle routing problem with time windows (VRPTW) in which the fleet of vehicles should service costumers within a time interval. The problem is NP-Hard [1]. There are exact algorithms, heuristic and meta-heuristic methods in the literature to solve the problem. Exact solution approaches for the TSPTW are the branch-and-bound algorithm [2] and the dynamic programming [3, 4]. Heuristic methods for the problem are constraint programming [5], construction heuristic with local search [6] and the compressed-annealing heuristic [7]. Some meta-heuristics for TSPTW are the ant colony optimization [8] and the artificial bee algorithm [9]. In this paper we present a differential evolution algorithm (DEA) for the TSPTW problems. After the problem formulation we explain the process of DEA and operators for the problem. In the fourth section we show some benchmark problem results and comparisons.

2. Problem Formulation

J.Q. Li's definition of the problem is modified as follows [4]; Let $N = \{1, 2, \ldots, n\}$ set of customers and 0 refers to the depot as the starting point and n+1 is the destination of the tour which is the same depot. The time windows of costumer i is $[a_i, b_i]$ and c_{ij} is the traveling cost from customer i to customer j which could be time, distance or fuel. x_{ij} is a binary variable when it is equal to 1, the salesman visits costumer j after costumer i. s_i is the time when the service starts at customer i. We assume the service time is 0 according to the benchmark problems that we use in this paper. The TSPTW is formulated as follows:

$$min \sum_{i=0}^{n} \sum_{j=1}^{n+1} c_{ij} x_{ij} \tag{1}$$

s.t.

$$\sum_{j=1}^{n+1} x_{ij} = 1, \qquad i = 0, 1, \dots, n \tag{2}$$

$$\sum_{i=0}^{n} x_{ij} = 1, \qquad j = 1, 2, \dots, n+1 \tag{3}$$

$$x_{ij}\left(s_i + t_{ij} - s_j\right) \leq 0, \qquad i = 0, 1, \dots, n, j = 1, 2, \dots, n+1 \tag{4}$$

$$a_i \leq s_i \leq b_i, \qquad \forall i = 1, 2, \dots, n+1 \tag{5}$$

$$s = 0 \tag{6}$$

$$x_{ij} \in \{0, 1\}, \qquad i = 0, 1, \dots, n, j = 1, 2, \dots, n+1.$$

Eq. 1 represents an objective which minimizes the total traveling cost. Constraints (2) and (3) ensure that each customer is served only once. Eq. 4 guarantees that the sales person visits customer j immediately after customer i, the visiting time of j will not be earlier then the arrival time from i. Eq. 5 guaranties that the service starting time is within the given time window. Finally, in Eq. 6 the salesman's tour starts from the depot at time 0. The objective in this paper is to minimize the total travel distance which is the same with the total travel time according to the benchmark problems if the waiting time before service start is ignored.

3. The Differential Evolution Algorithm for the TSPTW Problems

There are many techniques that have developed for solving NP-Hard problems such as the TSPTW. These problems become more difficult relating to the number of variables and constraints. The solution of these problems with deterministic methods may include difficulties in both modeling and solving depending on the problem. Heuristics and meta-heuristics are developed in order to overcome these difficulties. The differential evolutionary algorithm (DEA) related to the genetic algorithm (GA) is an efficient population based meta-heuristic optimization technique especially for the problems that include continuous variables [10-12]. In this paper we use the DEA for the TSPTW problem with using a different type of initial population. The general process and operators are detailed below.

The DEA is used for problems that have continuous variables. Some modifications for the DEA operators: crossover, mutation and selection, as in genetic algorithm, made the process less simple, and with around 20 lines of codes, the algorithm could work. The operators do not apply one by one as in the GA. The selected chromosome recombined with the other three random selected chromosomes according to the DEA operators, and one of them which has a better fitness value is selected for the next generation. For the TSPTW problems our initial population is created by using time windows. Through the operators they are changed as continuous variables. To evaluate the fitness value we use a sort function and genes which represent the costumers are being arranged according to their values in ascending order. Fig. 1 shows a process of the DEA for the TSPTW.

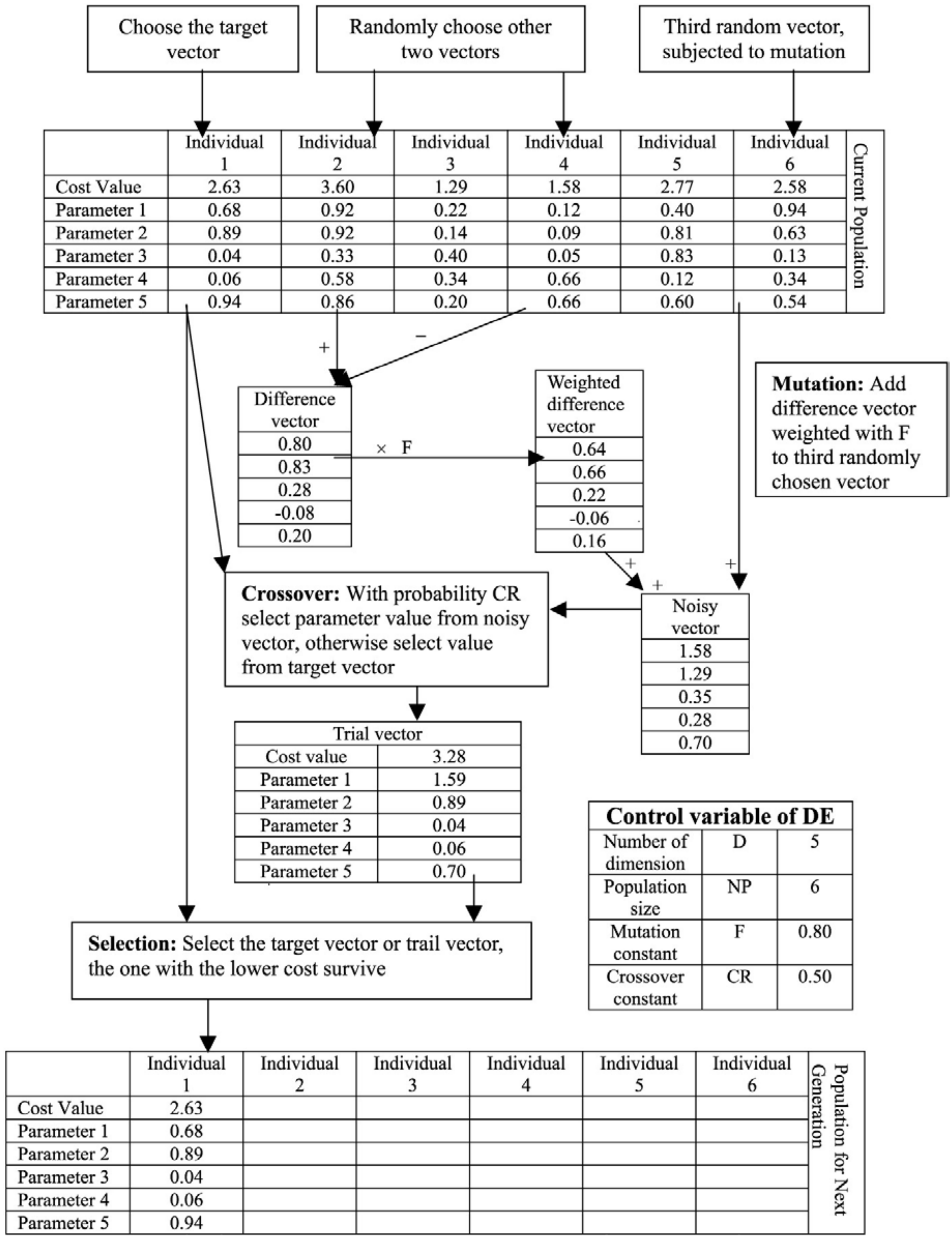

Fig. 1. DEA Flow chart [13].

4. Computational Results

The computational experimentation is performed on a Pentium Dual Core Machine with 4 GB of memory and 120 GB of hard drive using MATLAB R2009a. The parameters for the algorithm are selected as: F=0.5, number of iterations=1000, population size=50 and crossover rate=0.9. First the proposed DEA algorithm was tested on the asymmetric problems from the selected instances for the asymmetric TSPTW [14].

Table 1. The comparison of Peško's [15] DEA and the proposed DEA using instances from Ascheuer [14]

Instances	*N*	Optimum	Peško's DEA (Second)	Proposed DEA (CPU)
rbg10a	11	671	107	0
rbg017	18	893	203	0
rbg021	22	4536	245	4
rbg27	28	5091	561	13
rbg031a	32	1863	628	21

Secondly we have selected the test problems from Dumas et al [3]. The problems have 20 costumers and the time window width as 20, 40, 80 and 100. The problem difficulty increases with the time windows' width. The results of our algorithm compared with the optimum solutions are showed in Table 1.

Table 2. The results for DEA

Problem	***N***	**Width**	**Optimum**	**Proposed DEA**	**Iteration**	**CPU**
n20w20.001	20	20	378	378	1	0,06
n20w20.002	20	20	286	286	1	0,08
n20w20.003	20	20	394	394	1	0,06
n20w20.004	20	20	396	396	1	0,06
n20w20.005	20	20	352	352	1	0,04
n20w40.001	20	40	254	254	51	1,95
n20w40.002	20	40	333	333	5	0,20
n20w40.003	20	40	317	317	13	0,51
n20w40.004	20	40	388	388	14	0,56
n20w40.005	20	40	288	288	395	15,19
n20w60.001	20	60	335	335	140	5,31
n20w60.002	20	60	244	244	60	2,27
n20w60.003	20	60	352	352	26	1,02
n20w60.004	20	60	280	280	51	2,09
n20w60.005	20	60	338	338	708	27,00
n20w80.001	20	80	329	329	523	19,85
n20w80.002	20	80	338	338	20	0,78
n20w80.003	20	80	320	320	74	2,83
n20w80.004	20	80	304	304	363	14,07
n20w80.005	20	80	264	264	149	5,69

5. Conclusion

From the results it is clear that the differential evolution algorithm with the initial population which is created by using time windows could be a good alternative method to solve the travelling salesman problem with time windows. The algorithm codes could generate with different programming languages to improve CPU times. Also with a different local search, results could improve. For future research we plan to test our algorithm with different benchmark problems with more costumers and vehicles.

References

[1] M. W. P. Local Savelsbergh, search in routing problems with time windows, *Annals of Operations Research*, vol. 4, pp. 285-305, 1985.

[2] E. K. Baker, An exact algorithm for the time-constrained traveling salesman problem, *Operations Research*, vol. 31, pp. 938-945, 1983.

[3] Y. Dumas, J. Desrosiers, E. Gelinas, M. M. Solomon, An optimal algorithm for the traveling salesman problem with time windows, *Operations Research*, vol. 43, pp. 367-371, 1995.

[4] J.Q. Li, A Bi-directional Resource-bounded Dynamic Programming Approach for the Traveling Salesman Problem with Time Windows, 2008.

[5] G. Pesant, M. Gendreau, J. Y. Potvin, J. M. Rousseau, An exact constraint logic programming algorithm for the traveling salesman problem with time windows, *Transportation Science*, vol. 32, pp. 12-29, 1998.

[6] R. W. Calvo, A new heuristic for the traveling salesman problem with time windows, *Transportation Science*, vol. 34, pp. 113-124, 2000.

[7] J. W. Ohlmann, B. W. Thomas, A compressed-annealing heuristic for the traveling salesman problem with time windows, *INFORMS Journal on Computing*, vol. 19, pp. 80-90, 2007.

[8] C. B. Cheng, C. P. Mao, A modified ant colony system for solving the travelling salesman problem with time windows, *Mathematical and Computer Modelling*, vol. 46, pp. 1225-1235, 2007.

[9] K. Karabulut, M. F. Tasgetiren, A discrete artificial bee colony algorithm for the traveling salesman problem with time windows, *Evolutionary Computation (CEC)*, 2012 IEEE Congress on. IEEE, 2012.

[10] R. Storn, K. Price, Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, *Technical Report TR-95-012*, International Computer Science Institute, Berkeley, 1995.

[11] R. Storn, http://www.icsi.berkeley.edu/~storn/code.html, 2001.

[12] D.G. Mayer, B.P. Kinghorn, A.A. Archer, Differential Evolution – An Easy and Efficient Evolutionary Algorithm for Model Optimisation, *Agricultural Systems*, vol. 83, pp. 315-328, 2005.

[13] H. Schmidt, G. Thierauf, A Combined Heuristic Optimization Technique, *Advances in Engineering Software*, vol. 36, pp. 11-19, 2005.

[14] N. Ascheuer, Instances of the Asymmetric Traveling Salesman Problem with Time Windows (ATSPTW), available on http://elib.zib.de/pub/mptestdata/tsp/atsptw/text.en.html.

[15] Š. Peško, Differential Evolution for Small TSPs with Constraints, *Proceedings of the fourth International Scientific Conference: Challenges in Transport and Communications*, pp. 14-15, 2006.