Makespan Optimization for the Travelling Salesman Problem with Time Windows Using Differential Evolution Algorithm

Nihan Kabadayi, Istabul University School of Business, Turkey
Timur Keskinturk, Istabul University School of Business, Turkey

The Asian Conference on the Social Sciences 2015
Official Conference Proceedings

Travelling salesman problem with time windows (TSPTW) is a well-known NP-hard problem in which the objective is to minimize the total travel cost or makespan when visiting each of a set of customers within a given time period. In the literature, TSPTW with the objective of travel cost minimization is extensively studied and a good many heuristic methods are developed to solve these problems. However, TSPTW with the objective of makespan minimization rarely studied and fewer solution approaches are proposed. In this study, we develop a novel differential evolution algorithm to solve TSPTW with the objective of makespan minimization. The performance of the proposed algorithm is tested on several benchmark problems from the literature. According to the experimental results, the differential evaluation algorithm outperforms the existing approaches for makespan minimization.

Keywords: travelling salesman problem, time windows, differential evolution algorithm, makespan optimization

Introduction

Travelling salesman problem is a well-known NP-hard problem in which a set of nodes are visited only once by a single vehicle with the objective of minimization the tour cost starting and ending at a given depot. Each customer has a time-window defining its earliest and latest service start time and a service time. In a feasible tour, each customer has to be visited before its latest service time. Otherwise the tour is qualified as an infeasible tour. Moreover, if a customer is visited before its earliest service time, the delivery has to wait until the earliest service time is reached. The vehicle spends a predetermined period of time at each customer to conduct its service. Afterwards the service is completed the vehicle immediately leaves the relevant customer to serve the next scheduled customer.

TSPTW problems have two possible minimizing objective functions that are minimizing the total travel time on the path or minimizing the total travel time on the path. The former objective function has been more commonly investigated in the literature. Baker (1983) developed a branch-and-bound approach, Dumas et al. (1995) used an exact dynamic programming algorithm, whereas Langevin et al. (1993) proposed a branch-and-cut algorithm to solve TSPTW with two-commodity flow. Mingozzi, Bianco, & Ricciardelli (1997) suggested a two-phase dynamic programming method and their proposed algorithm outperforms previous algorithms when solving similar test problems. Pesant et al. (1998) developed a novel branch-and-bound optimization algorithm without any restrictive assumption on the time windows. Ascheuer et al. (2001) studied the asymmetric version of the TSPTW, and generated a branch-and-cut algorithm applying techniques for these kinds of problems. In this study, recent techniques such as data preprocessing, primal heuristics, local search, and variable fixing are adapted for the asymmetric TSPTW.

Since TSPTW problem is an NP-hard problem, the exact algorithms are limited to solve some large TSPTW problems. Thus, heuristic and meta-heuristic algorithms are developed to solve large TSPTW problems in a short time. Carlton and Barnes (1996) proposed a robust tabu search algorithm with a static penalty function to promote more diverse neighborhood search. Gendreau et al. (1998) developed insertion heuristic based on GENIUS heuristic which a combination of the insertion procedure and a post-optimization procedure. Calvo (2000) suggested a classical two-phased insertion heuristic with 3-opt local search to solve TSPTW problems. The advantage of their algorithm is that it is able to find an initial solution close enough to a feasible solution of the original problem with solving an assignment problem with an ad hoc objective function. Ohlman and Thomas (2007) presented simulated annealing variant compressed annealing that integrates a penalty method with heuristic search and they showed that compressed annealing method employing a variable penalty function outperforms simulated annealing method with a static penalty function. Ibanez and Blum (2010) proposed a new heuristic named as Beam-ACO that is combining ant colony optimization and beam search to solve TSPTW for minimizing the travel cost. Silva and Urrutia (2010) suggested a two stage VNS based heuristic, which is composed of constructive and optimization stages. They show that, the proposed algorithm is an effective method, which can solve larger problem instances in a short time and improves some best knows results from the literature.

Authors developing heuristic approaches have focused on travel-time optimization and fewer results are published on makespan optimization. The objective of the TSPTW with makespan optimization is to minimize the total tour completion time. In the makespan calculation, the waiting times are included to the tour time. Langevin (1993), Carlton and Barnes (1996), Cheng (2007) and Favaretto (2006) considered TSPTW with makespan optimization objective in their research. Ibanez et al. (2013) adapted two novel algorithms for the TSPTW with travel time minimization objective, compressed annealing (CA) and the Beam-ACO algorithm, to solve makespan optimization objected TSPTW. Karabulut and Tasgetiren (2014) generated a variable iterated greedy algorithm for solving the traveling salesman problem with time windows (TSPTW) to minimize the total travel cost and the total travel completion time so called makespan, separately. They showed that the proposed method outperforms the other methods in terms of performance and speed.

In the literature TSPTW with the objective of travel cost minimization is extensively studied and a good many heuristic methods are developed to solve these problems. However TSPTW with the objective of makespan minimization rarely studied and fewer solution approaches are proposed. Therefore, in this study we developed a novel differential evolution algorithm to solve TSPTW with makespan minimization objective. The performance of the proposed algorithm is tested on several benchmark problems from the literature. According to the experimental results, the differential evaluation algorithm outperforms the existing approaches for makespan minimization. The paper is organized as follows. Section 2 provides the problem formulation. In Section 3, the proposed method differential evolution algorithm is considered. Computational experiments and results are presented in Section 4. Finally, conclusions are reported in Section 5.

Problem Formulation

Graph *G = (V, A)* is given, where *V = {0, 1, 2,…, n}* is a set of nodes representing the depot (node 0) and n customers and the arc set is A $=\{(i,j): i,j \in V,\ i \neq j\}$. For every edge $(i,j) \in A$ between two nodes i and j, there is an associated cost c_{ij} that denotes the travelling cost from customer i to customer j which includes both the service time of a customer i and the time needed to travel from i to j. Additionally to this, each customer i has an associated time window $[e_i, l_i]$ where e_i and l_i represent the ready time and the due date, respectively. Therefore, the TSPTW can be considered as a problem of finding a Hamiltonian tour that starts and ends at the depot, satisfying all time windows constraints and minimizing the total distance traveled. The TSPTW with the objective of the total travel time is formulated by Karabulut and Tasgetiren (2014) as follows:

$$\min f(x) = \sum_{i=0}^{n} c(x_i\,, x_{i+1}) + \sum\nolimits_{i=0}^{n+1} W_{x_i} \qquad (1)$$

$$st\colon p(x) = \sum_{i=0}^{n+1} \omega(x_i) = 0 \qquad (2)$$

where

$$\omega(x_i) = \begin{Bmatrix} 1 & if \quad A_{x_i} > I_{x_i} \\ 0 & otherwise \end{Bmatrix} \qquad (3)$$

Equation 1 represents the objective function, which minimizes the total travelling cost, which includes the waiting times. In the above definition, W_{x_i} denotes the waiting time at customer i and $p(x)$ represents the total number of time windows, which are violated by tour x that must be zero for feasible solutions. Equation 2 guarantees that all the customers are served within the relative time windows and there is no time windows violation.

Differential Evolution Algorithm for the TSPTW Problems

Differential evolution algorithm (DEA) is one of the relatively new and effective population based meta-heuristic method. It is firstly developed by Storn and Price (1997). This evolutionary algorithm is related to genetic algorithm (GA) and it is easy to implement and use, effective and efficient when compared to genetic algorithm. It uses same operators with genetic algorithm such as crossover and mutation and selection. However the operators are implemented simultaneously in DEA. In addition to this, DEA relays on mutation while GA rely on crossover.

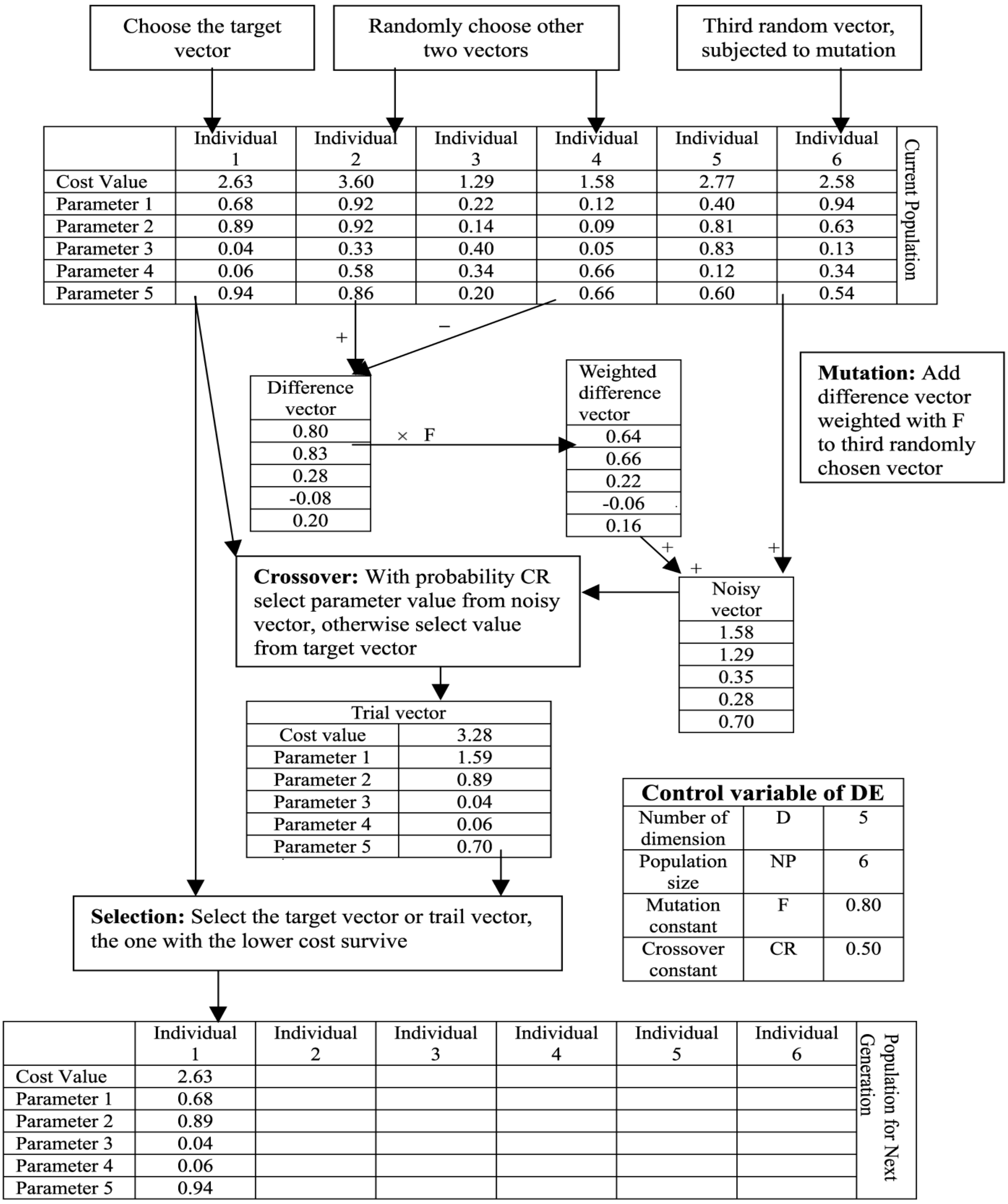

Figure 1 - DEA Flow Chart (Schmidt and Thierauf, 2005)

The Figure 1 determines the flow chart of DEA. 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.

Computational Experiments

In this section, we present our results from our computational experiments. We aim to apply our proposed DEA method to solve TSPTW problems and evaluate the performance of our method. And then we compare our computational results with the state-of-art methods for the previously described test instances. 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.

Table 1- Results for Potvin&Bengio Test Instances

		VIG_VNS		ACO		DEA	
Problem	**N**	**Avg**	**Best**	**Avg**	**Best**	**Avg**	**Best**
rc201.1	20	592,06	592,06	592,06	592,06	592,06	592,06
rc201.2	26	869,9	869,9	877,49	877,49	**860,175**	**860,175**
rc201.3	32	854,12	854,12	867,61	853,71	**853,71**	**853,708**
rc201.4	26	889,18	889,18	900,52	900,38	889,18	889,18
rc202.1	33	850,48	850,48	880,74	871,11	850,71	850,48
rc202.2	14	342,20	342,20	338,52	338,52	345,17	338,52
rc202.3	29	904,48	904,48	892,18	**847,31**	894,10	894,10
rc202.4	28	854,12	854,12	∞	∞	855,06	**853,71**
rc203.1	19	488,42	488,42	673,07	663,66	490,45	488,42
rc203.2	33	853,71	853,71	926,75	897,88	863,69	853,71
rc203.3	37	956,92	956,92	∞	∞	1108,38	**921,44**
rc203.4	15	350,83	350,83	493,85	493,85	**338,52**	**338,52**
rc204.1	46	950,36	950,36	949,68	949,68	**925,877**	**925,234**
rc204.2	33	**701,62**	**701,62**	863,65	821,63	774,59	715,04
rc204.3	24	**455,03**	**455,03**	642,06	635,36	516,74	**455,03**
rc204.4	14	426,13	426,13	428,39	425,2	**417,81**	**417,81**
rc205.1	14	455,94	455,94	422,24	**417,81**	**417,81**	**417,81**
rc205.2	27	820,19	820,19	820,19	820,19	820,19	820,19
rc205.3	35	950,05	950,05	950,05	950,05	950,05	950,05
rc205.4	28	867,13	867,13	870,43	850,99	**837,71**	**837,71**
rc206.1	4	117,85	117,85	117,85	117,85	117,85	117,85
rc206.2	37	917,26	917,26	914,99	909,30	882,6824	**870,488**
rc206.3	25	661,07	661,07	650,59	650,59	650,59	650,59
rc206.4	38	930,1	930,1	943,31	943,31	912,28	**911,98**
rc207.1	34	865,07	865,07	860,98	851,06	**833,40**	**821,78**
rc207.2	31	735,56	735,56	∞	∞	**734,25**	**713,90**
rc207.3	33	800,39	800,39	955,7	944,52	**795,984**	**751,494**
rc207.4	6	133,14	133,14	133,14	133,14	133,14	133,14
rc208.1	38	841,28	841,06	934,8	925,36	822,12	**810,701**
rc208.2	29	644,13	644,13	722,24	712,96	**601,29**	**580,205**
rc208.3	36	747,15	747,15	795,03	774,72	750,70	**698,69**

Firstly, the proposed DEA algorithm was tested on the symmetric TSPTW problems derived by Potvin&Bengio (1996). The computational results of the DEA for Potvin& Bengio (1996)'s instances are given in Table 1. The results show that the DEA can generate better solutions than the other methods for some test problems. Secondly, we tested our algorithm on the asymmetric test instances derived by Ascheuer. The Ascheuer benchmark set consists of 50 asymmetric TSPTW instances with up to 233 nodes. Travel times are integer and satisfy the triangle inequality. The test results show that our algorithm also performs well on the asymmetric test instances. The test results are shown in Table 2.

Table 2- Results for Ascheuer Instances

Problem	N	Best	DEA Avg.	DEA Best	Difference	GAP (%)
rbg010a	12	-	3840	3840	-	-
rbg016a	18	-	2596	2596	-	-
rbg016b	18	2094	2094	2094	0	0,0%
rbg017.2	17	2351	2351	2351	0	0,0%
rbg017	17	-	2351	2351	-	-
rbg017a	19	4296	4296	4296	0	0,0%
rbg019a	21	2694	2694	2694	0	0,0%
rbg019b	21	-	3840	3840	-	-
rbg019c	21	-	4536	4536	-	-
rbg019d	21	3479	3479	3479	0	0,0%
rbg020a	22	4689	4689	4689	0	0,0%
rbg021.2	21	4528	4528	4528	0	0,0%
rbg021.3	21	4528	4528,0	4528	0	0,0%
rbg021.4	21	-	4535	4530	-	-
rbg021.5	21	4516	4516,5	4516	0,5	0,0%
rbg021.6	21	4492	4496,2	4486	4,2	0,1%
rbg021.7	21	4481	4487	4481	6,3	0,1%
rbg021.8	21	4481	4485	4481	4,0	0,1%
rbg021.9	21	4481	4485	4481	3,7	0,1%
rbg021	21	-	4536	4536	-	-
rbg027a	29	5093	5093	5093	0,0	0,0%
rbg031a	33	3498	3498	3498	0	0,0%
rbg033a	35	3757	3757	3757	0	0,0%
rbg034a	36	3314	3314	3314	0	0,0%
rbg035a.2	37	3325	3325	3325	0	0,0%
rbg035a	37	3388	3388	3388	0	0,0%
rbg038a	40	5699	5699	5699	0	0,0%
rbg040a	42	5679	5679	5679	0	0,0%
rbg041a	43	3793	3793	3793	0	0,0%
rbg042a	44	3260	3260,4	3260	0,43	0,0%
rbg048a	50	9799	9799	9799	0,0	0,0%
rbg049a	51	13257	13257	13257	0	0,0%
rbg050a	52	-	12050	12050	-	-
rbg050b	52	11957	11958	11957	0,5	0,0%
rbg050c	52	10985	10985	10985	0,0	0,0%
rbg055a	57	6929	6929	6929	0	0,0%
rbg067a	69	10331	10331	10331	0	0,0%
rbg086a	88	16899	16899	16899	0	0,0%
rbg092a	94	-	12501	12501	-	-
rbg125a	127	14214	14214	14214	0	0,0%
rbg132.2	132	18524	18524	18524	0	0,0%

rbg132	132	18524	18524	18524	0	0,0%
rbg152.3	152	-	17455	17455	-	-
rbg152	152	17455	17455	17455	0	0,0%
rbg172a	174	17783	17784	17783	1	0,0%
rbg193.2	193	21401	21401	21401	0	0,0%
rbg193	193	21401	21401	21401	0	0,0%
rbg201a	203	21380	21380	21380	0	0,0%
rbg233.2	233	26143	26143	26143	0	0,0%
rbg233	233	26143	26143	26143	0	0,0%

Conclusion

Travelling salesman problem is a well-known NP-hard problem in which a set of nodes are visited only once by a single vehicle with the objective of minimization the tour cost starting and ending at a given depot. In the literature TSPTW with the objective of travel cost minimization is extensively studied and a good many heuristic methods are developed to solve these problems. However TSPTW with the objective of makespan minimization rarely studied and fewer solution approaches are proposed. Therefore, in this study we developed a novel differential evolution algorithm to solve TSPTW with makespan minimization objective. We conducted computational experiments to compare the results of the existing heuristics for TSPTW with the results generated by DEA. The experimental results show that our algorithm is comparable with all known heuristic approaches to the problem in terms of the solution quality. Therefore, the proposed DEA algorithm is a good alternative solution methodology to solve TSPTW problems. Additionally, the proposed DEA algorithm can be extended to other variants of TSP problem, algorithm codes could generate with a different programming languages to improve the CPU times or a new meta-heuristics can also be applied to solve the TSPTW in the future research.

References

Ascheuer N. Hamiltonian path problems in the on-line optimization of flexible manufacturing systems. PhD thesis, Technische University at Berlin, Berlin, Germany; 1995.

Ascheuer, N., Fischetti, M., & Grötschel, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, *90*(3), 475-506.

Baker, E. K. (1983). An exact algorithm for the time-constrained travelling salesman problem, *Operations Research*, 31(5), 938-945.

Calvo, R. W. (2000). A new heuristic for the traveling salesman problem with time windows. *Transportation Science*, *34*(1), 113-124.

Carlton, W. B., & Barnes, J. W. (1996). Solving the traveling-salesman problem with time windows using tabu search. *IIE transactions*, *28*(8), 617-629.

Cheng, C. B., & Mao, C. P. (2007). A modified ant colony system for solving the travelling salesman problem with time windows. *Mathematical and Computer Modelling*, *46*(9), 1225-1235.

Da Silva, R. F., & Urrutia, S. (2010). A General VNS heuristic for the traveling salesman problem with time windows. *Discrete Optimization*, *7*(4), 203-211.

Favaretto, D., Moretti, E., & Pellegrini, P. (2006). An ant colony system approach for variants of the traveling salesman problem with time windows. *Journal of information and optimization sciences*, *27*(1), 35-54.

Gendreau, M., Hertz, A., Laporte, G., & Stan, M. (1998). A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research*, *46*(3), 330-335.

Karabulut , K., Tasgetiren M.F. (2014). A variable iterated greedy algorithm for the travelling salesman problem with time windows. *Information Sciences, 279*, 383-395.

Langevin, A., Desrochers, M., Desrosiers, J., Gélinas, S., & Soumis, F. (1993). A two-commodity flow formulation for the traveling salesman and the makespan problems with time windows. *Networks*, *23*(7), 631-640.

López-Ibáñez, M., & Blum, C. (2010). Beam-ACO for the travelling salesman problem with time windows. *Computers & operations research*, *37*(9), 1570-1583.

López-Ibáñez, M., Blum, C., Ohlmann, J. W., & Thomas, B. W. (2013). The travelling salesman problem with time windows: Adapting algorithms from travel-time to makespan optimization. *Applied Soft Computing*, *13*(9), 3806-3815.

Mingozzi, A., Bianco, L., & Ricciardelli, S. (1997). Dynamic programming strategies for the traveling salesman problem with time window and precedence constraints. *Operations Research*, *45*(3), 365-377.

Ohlmann, J. W., & Thomas, B. W. (2007). A compressed-annealing heuristic for the traveling salesman problem with time windows. *INFORMS Journal on Computing*, *19*(1), 80-90.

Pesant, G., Gendreau, M., Potvin, J. Y., & Rousseau, J. M. (1998). An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science*, *32*(1), 12-29.

Potvin, J. Y., & Bengio, S. (1996). The vehicle routing problem with time windows part II: genetic search. *INFORMS journal on Computing*, *8*(2), 165-172.

Schmidt H., & Thierauf, A. (2005). A Combined Heuristic Optimization Technique. *Advances in Engineering Software*,36, 11-19.

Storn, R., & Price, K. (1997). Differential Evolution-A Simple Evolution Strategy for Fast Optimization. *Dr. Dobby's Journal*. 78, 18-24.