Creative Construction Conference 2016

A Mixed Integer Model for Optimization of Discrete Time Cost Tradeoff Problem

Ali Can Tatar[1]*, Mert Bilir[2], Rifat Sonmez[3], Tankut Atan[4]

[1] *Research Assistant, Department of Civil Engineering, Middle East Technical University, Ankara, 06800, Turkey*
[2] *Research Assistant, Department of Civil Engineering, Middle East Technical University, Ankara, 06800, Turkey*
[3] *Associate Professor, Department of Civil Engineering, Middle East Technical University, Ankara, 06800, Turkey*
[4] *Assistant Professor, Department of Industrial Engineering, Isik University, Istanbul, 34980, Turkey*

Abstract

In construction projects, activity durations can be expedited by allocating additional resources. Decreasing activity durations by means of crashing, usually leads to increase in the direct expenses. This trade-off between time and cost is called as the time-cost trade-off problem. Since in practice many resources are available in discrete units, numerous research has focussed on the discrete version of this problem called the discrete time-cost trade-off problem (DTCTP). Achieving the project schedule that satisfies the project requirements at an optimum cost is crucial for effective scheduling and management of construction projects. Despite the importance of DTCTP, very few research focused on generating and solving of large scale instances. The objective of this proceeding is to generate large scale instances that reflect the size of real-life construction projects and to solve these instances using mixed integer programming method (MIP) to enable a benchmark set with optimal solutions. Within this context, large scale instances that reflect the size of real-life-size construction projects are generated. A MIP model is developed and the majority of the instances is solved to optimal using GUROBI optimizer.

Keywords: cost optimization, discrete time-cost trade-off problem, exact methods, mixed integer programming, project management

1. Introduction

Construction projects have a certain scope, budget and schedule. Especially, throughout a construction project, budget and schedule have essential effects on each other. Schedule of a project may be shortened by means of expediting activities that requires working overtime or increasing crew size. Accordingly, crashing a project schedule leads to additional cost. This trade-off between activity durations and costs is defined as time cost trade-off problem (TCTP). In the construction industry, the majority of resources are available in discrete units; hence, numerous researches have been conducted on the discrete version of the problem called as discrete time cost trade-off problem (DTCTP).

In the literature, DTCTP has been examined in terms of three categories as deadline problem, budget problem and time-cost curve (Pareto front curve) problem. Deadline problem aims to minimize the total cost with respect to a given project deadline. Budget problem, on the other hand, aims to minimize the project duration within a given budget. The Pareto front problem aims to construct complete and efficient time-cost profile over the set of feasible durations [1].

Methods proposed to solve DTCTP can be classified as exact methods, heuristics and meta-heuristics. De et al. (1997) [2] define DTCTP as strongly non-polynomial hard (NP-hard). Hence, exact methods may require significant amount of computational time [3]. On the other hand, exact methods guarantee optimality. Hence, they provide a benchmark to evaluate the results of heuristic and meta-heuristics. As a pioneer study within the context

* Corresponding author. Tel.: +90-312-2105422; fax: +90-312-2105401.
E-mail address: tatar@metu.edu.tr

of heuristics, Siemens (1971) [4] presented the Siemens approximation method (SAM) for the cost optimization problem with strict deadlines, and implemented it on a problem with eight activities. Goyal (1975) [5] proposed a modified version of SAM and used the same example with 8 activities. Moselhi (1993) [6] developed a heuristic based on schedule compression. Apart from these, numerous meta-heuristics have been presented for the DTCTP. Hegazy (1999) [7] developed a genetic algorithm (GA) for the cost optimization problem. Sonmez and Bettemir (2012) [8] developed a hybrid strategy based on GAs, simulated annealing (SA), and quantum simulated annealing techniques (QSA) for the cost optimization problem. Zheng (2015) [9] presented a GA for the discrete time-cost-environment trade-off problem. Zhang et al. (2015) [10] proposed a GA for the DTCTP in repetitive projects. Elbeltagi et al. (2007) [11] presented a shuffled frog-leaping algorithm for the cost optimization problem. Vanhoucke and Debels. (2007) [1] developed a meta-heuristic involving tabu-search and truncated dynamic programming for the cost optimization problem with strict deadlines. Abdel-Raheem and Khalafallah (2011) [12], proposed an evolutionary algorithm which simulates the behavior of electrons moving through electric circuit branches for the cost optimization problem. Tavana et al. (2014) [13] presented two multi-objective procedures based on Ɛ-constraint method and dynamic self-adaptive evolutionary algorithm for the discrete time-cost-quality trade-off problem. Elbeltagi et al. (2005) [14] and Bettemir (2009) [15] adopted the particle swarm optimization method (PSO) for the cost optimization problem.

In terms of exact methods, mixed integer programming (MIP), dynamic programming (DP) and branch and bound algorithm are the most widely known examples in the literature. Meyer and Shaffer (1963) [16], Crowston and Thompson (1967) [17], Crowston (1970) [18], Harvey and Patterson (1979) [19] are the pioneer studies solving TCT with mixed MIP algorithms., Liu et al. (1995) [20] solved TCT problem for a network with seven activities in Microsoft Excel environment linear and integer programming., Moussourakis and Haksever (2004) [3] presented a flexible MIP model for TCT problems. The term flexible represents that the model has the capability to solve different TCT problems by applying minor modifications. Deadline problem was studied with a problem including 7 activities. Vanhoucke (2005) [21] proposed a branch and bound algorithm for the cost optimization problem with strict deadlines and time-switch constraints. The algorithm is capable of solving instances with up to 30 activities. Akkan et al. (2005) [22] provided lower and upper bounds for the cost optimization problem with strict deadlines. Hazir et al. (2010) [23] presented an exact method based on Benders Decomposition for the duration optimization problem, and was able to solve instances including up to 136 activities with 10 modes within 90 minutes. Szmerekovsky and Venkateshan (2012) [24] studied four integer programming formulations for irregular time-cost trade-offs and achieved optimal solutions for instances including up to 90 activities.

In spite of existing studies, there is a gap in the literature in terms of exact methods that are applicable to medium to large scale DTCTPs reflecting the size of real-life construction projects. Liberatore et al. (2001) [25] indicates that a real-life construction project consists more than 300 activities. Hence, the main objective of this paper is to generate medium and large size problem instances including delay penalty and to provide a MIP model for solving these instances.

2. Problem Set Generation

There are few popular benchmark instances for DTCTP in the literature. The network generated by Feng et al (1997) [26] is one of them. However, problem includes only 18 activities. Within the scope of this study, medium and large scale problem instances are generated using ProGen/max random network generator developed by Schwindt (1995) [27].

Project networks are developed with four different complexity indexes. In ProGen/max, complexity index is represented by Thesen restrictiveness coefficient. Accordingly, four different Thesen restrictiveness coefficients of; 0.2, 0.4, 0.6, and 0.8 are used for networks. For each of these coefficients, networks including 50, 100, 200, 500 and 1000 activities are generated. Details of the problem sets are provided in Table 1.

Table 18: Parameter Inputs Entered to ProGen/max for Different Number of Activities.

Parameter	50 Activities	100 Activities	200 Activities	500 Activities	1000 Activities
Minimal Number of Initial Activities	1	1	1	1	1
Maximal Number of Initial Activities	12	20	20	20	20
Minimal Number of Terminal Activities	1	1	1	1	1
Maximal Number of Terminal Activities	12	20	20	20	20
Maximal Number of Predecessor Activities	12	20	20	20	20
Maximal Number of Successor Activities	12	20	20	20	20
Degree of Redundancy	0.1	0.1	0.1	0.1	0.1

ProGen/max is actually designed to generate problem networks for resource constraint project scheduling. The network generator does not create time-cost modes for DTCTP. Hence, it is utilized only for formation of networks. Time-cost modes for the created networks are prepared in Microsoft Excel 2010. Determination of the modes is done according to Akkan et al. (2005) [22]. For the time-cost modes, three intervals including 2-5, 6-10, and 11-15 modes are used. Duration of each time-cost mode is selected randomly between 3 days and 123 days. This interval is divided to the number of modes accordingly. For instance, if number of modes for an activity is five:

- duration of the first mode is chosen between 99 days and 123 days,
- duration of the second mode is chosen between 75 days and 98 days,
- duration of the third mode is chosen between 51 days and 74 days,
- duration of the fourth mode is chosen between 27 days and 50 days,
- duration of the fifth mode is chosen between 3 days and 26 days.

The amount of direct cost for the first mode is determined randomly between 100 USD and 50000 USD. Costs for the remaining modes are determined according to following formula used by Akkan et al. (2005) [22].

$$c_{k+1} = c_k + s_k * (d_k - d_{k+1}) \quad (2.1)$$

where,
c_{k+1} : cost value for time-cost mode $k+1$
c_k : cost value for time-cost mode k
s_k : randomly generated time-cost slope value
d_k : duration value for time-cost mode k
d_{k+1} : duration value for time-cost mode k

Randomly generated time-cost slope values are determined between 10 and 100. In all the created networks, there are two dummy activities representing the start and finish of the project, which do not have a duration and cost. Time-cost modes of a sample network is given in Table 2.

Table 19: Time-Cost Modes for Activities.

Activity ID	# of Modes	Dur#1	Cost#1	Dur#2	Cost#2	Dur#3	Cost#3	Dur#4	Cost#4	Dur#5	Cost#5
1	4	95	40683	74	41959	44	43617	20	44503		
2	4	108	14530	71	14901	38	17063	14	19013		
3	3	120	5601	75	6847	12	7696				
4	3	117	47388	69	49194	35	51688				
5	5	115	12836	97	13791	52	18146	43	18371	6	21693
6	3	92	29250	74	30592	22	32809				
7	5	100	48919	80	49690	60	49926	38	50190	17	52280
8	4	95	14053	63	16392	59	16499	15	19174		
9	4	96	3924	70	5456	56	5610	27	6352		
10	4	120	28039	86	30314	58	32393	13	33346		
11	5	110	38588	77	40499	69	41254	35	42681	17	44280
12	5	115	34151	95	34925	61	36344	40	37703	18	39410
13	3	95	3014	44	4742	6	8034				
14	3	119	31825	50	38677	26	41035				
15	5	118	13988	97	16003	57	16404	28	18819	17	19032
16	5	114	13620	83	16051	67	16458	42	18418	10	19754
17	5	106	7972	80	8538	56	9683	37	10291	17	10960
18	4	114	44660	92	45669	42	49528	24	50825		
19	3	107	48807	64	50401	38	50807				
20	4	121	27062	72	28688	42	29492	5	31251		
21	4	98	40094	77	40641	47	43503	31	44124		
22	4	104	7081	84	7618	45	10789	30	10949		

Generally, construction projects include a delay penalty. Therefore, a delay penalty is included in the problems. The project deadline and delay penalty are determined by:

$$Deadline = \frac{CPMMax - CPMMin}{2} + CPMMin \quad (2.2)$$

$$\text{Cost of } Delay\ Penalty = Indirect\ Cost * 2 \quad (2.3)$$

where

$CPMMax$: Maximum CPM duration calculated by taking the longest duration in time-cost modes of the activities

$CPMMin$: Minimum CPM duration calculated by taking the shortest duration in time-cost modes of the activities

$Deadline$: Project deadline

Using this model, delay penalty is added to project networks. Cost of delay penalty is determined as the double of indirect costs for each network. If the project duration exceeds the defined deadline, delay penalty is added to the total cost.

There are four different complexity indexes and 3 different mode intervals for project networks having 50, 100, 200, 500, and 1000 activities. 10 instances are prepared for each set. Details of the sets are shown in Table 3. A total of 600 test instances are prepared. The daily indirect cost is set as 250 USD for networks having 50 activities. For the rest of the networks, daily indirect cost is set as 500 USD.

Table 3: Number of Problem Sets Prepared for DTCTP Analyses

	Number of Time-Cost Modes												
	2--5				6--10				11--15				
	Thesen Restrictiveness Coefficient				Thesen Restrictiveness Coefficient				Thesen Restrictiveness Coefficient				
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	
Number of Activities	Number of Test Instances				Number of Test Instances				Number of Test Instances				Daily Indirect Cost (USD)
50	10	10	10	10	10	10	10	10	10	10	10	10	250
100	10	10	10	10	10	10	10	10	10	10	10	10	500
200	10	10	10	10	10	10	10	10	10	10	10	10	500
500	10	10	10	10	10	10	10	10	10	10	10	10	500
1000	10	10	10	10	10	10	10	10	10	10	10	10	500

3. Model Description

In order to minimize the total cost of the projects comprised of direct and indirect costs, the following model based on De et al. (1995) [28] is proposed.

3.1. *Sets*

Pd_j : Predecessors of activity j
S : Activities in the network (exluding dummy activities)

3.2. *Parameters*

dc_{jk} : cost of activity j for time-cost mode k
i_c : daily indirect cost
d_{jk} : duration of activity j for time-cost mode k
$m(j)$: number of time-cost modes for activity j
d_p : daily delay penalty cost

3.3. *Variables*

Ft_j : finish date of activity j
Ft_h : finish date of activity h
x_{jk} : binary variable which is 1 if time-cost mode k is chosen to realize activity j, if not 0.
D : total duration of the project
D_{delay} : amount of delay
$D_{deadline}$: project deadline

3.4. *Mixed Integer Programming Model*

$$\min \sum_{j=1}^{S} \sum_{k=1}^{m(j)} dc_{jk} x_{jk} + D i_c + d_p D_{delay} \tag{3.1}$$

Constraints:

$$\sum_{k=1}^{m(j)} x_{jk} = 1, \quad \forall j \in S \tag{3.2}$$

$$Ft_j \geq Ft_h + \sum_{j=1}^{S} \sum_{k=1}^{m(j)} d_{jk} x_{jk}, \quad \forall h \in Pd_j \text{ and } \forall j \in S \tag{3.3}$$

$$D \geq Ft_{S+1} \tag{3.4}$$

$$Ft_0 = 0 \tag{3.5}$$

$$D - D_{delay} \leq D_{deadline} \tag{3.6}$$

$$D \geq 0 \tag{3.7}$$

$$D_{delay} \geq 0 \tag{3.8}$$

$$x_{jk} \in \{0,1\}, \quad \forall j \in S, \text{ and } \forall h \in m(j) \tag{3.9}$$

$$Ft_j \geq 0, \quad \forall j \in S \tag{3.10}$$

Objective function (3.1) aims to minimize the total cost of the project that equals to the summation of direct and indirect costs. Constraint (3.2) ensures that only one time-cost mode is chosen for each activity. For instance, if the second mode of Activity 3 is chosen in a sample network, x_{32} value is equal to 1 and x_{31}, x_{33}, x_{34}, x_{35} are equal to 0. Since these values are the multipliers of activity costs, only the selected time-cost mode affects the total project duration. Constraint (3.3) defines that an activity cannot finish earlier than the date represents the summation of activity's duration of the selected mode and the finish date of its predecessors. Constraint (3.4) explains that the project cannot be completed until the final dummy activity is finished. In constraint (3.5), finish date of initial dummy activity is set to 0. Constraint (3.6) represents the relation between the deadline and the amount of delay. The next two constraints (3.7), (3.8) explain that both the total project duration and the amount of delay should be positive values respectively. (3.9) indicates that xjk is a binary variable. The last constraint (3.10) shows that finish dates of all activities must be greater than 0. Finally, the dummy activities (Activity 0 and Activity S+1) do not have any duration and cost.

4. Computational Experiments

GUROBI Optimizer 5.6.3 is used to solve the MIP model given in the previous section. First, problem networks given in Section 2 are modelled in terms of LP format. Then, they are analyzed in GUROBI, with a time limit of 600 seconds for each problem. All experiments are conducted with a desktop computer using Windows 7 Professional Edition (64-bit) operating system with an Intel Core i5 3.10 CPU GHz and a 4.00 GB random access memory (RAM). The number of problem instances that are solved to optimal are in the given in the following Table 4.

Table 4. Number of Problems Solved to Optimal.

	Number of Time-Cost Modes													
	2--5				6--10				11--15					
	Thesen Restrictiveness Coefficient				Thesen Restrictiveness Coefficient				Thesen Restrictiveness Coefficient					
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8		
Number of Activities	Number of Test Instances				Number of Test Instances				Number of Test Instances				Solved Number of Total Instances	Solution Percentage (%)
50	10	10	10	10	10	10	10	10	10	10	10	10	120	100
100	10	10	10	10	9	10	5	10	10	10	10	10	114	95
200	10	10	10	10	9	9	5	9	6	7	5	8	98	81,7
500	10	10	8	10	5	5	3	2	5	4	6	5	73	60,8
1000	7	6	8	7	3	2	5	4	4	6	1	4	57	47,5

All of the problems consisting of 50 activities are solved within the time limit. As the number of activities in a network increases, the number of solved problems decreases. The CPU time of the problems solved to optimal are shown in Table 5.

Table 5: Average CPU Time in Optimal Cost Solutions

	Number of Time-Cost Modes											
	2--5				6--10				11--15			
	Thesen Restrictiveness Coefficient				Thesen Restrictiveness Coefficient				Thesen Restrictiveness Coefficient			
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
Number of Activities	CPU Time (seconds)											
50	0.12	0.14	0.24	0.13	0.41	0.87	1.04	0.58	0.82	1.06	3.33	0.93
100	0.54	0.28	1.28	0.33	176.34	56.54	2.08	0.96	0.03	0.35	1.33	1.22
200	8.34	0.54	2.76	3.32	161.23	13.54	77.80	7.78	168.44	86.68	11.40	3.00
500	29.20	9.69	22.32	24.29	134.22	96.39	16.26	175.76	81.26	145.20	220.17	38.13
1000	20.75	54.02	68.28	90.38	303.93	267.66	201.60	119.37	316.39	87.62	58.37	285.16

Results given in Table 5 are compared to similar studies in the literature. Hazir et all (2011) [23] solve a 136-activity network with an MIP algorithm in 19139.61 seconds. Furthermore, the model proposed by Szmerekovsky and Venkateshan (2012) [24] require a CPU time of 206 seconds to solve a 90-activity network.

5. Conclusion

In this study a total of 600 problems DTCTP instances, including 1000 activities is generated. A MIP model including a delay penalty is used to solve the benchmark problems to optimal. The majority of the small and medium scale instances and some of the large scale instances of are solved to optimal, within 600 seconds with GUROBI Optimizer 5.6.3. Hence, optimal solutions can be achieved for the deadline problem for evaluating the results of the heuristic and meta-heuristic methods. The CPU time of the MIP model presented can be decreased by using parallel processing methods which is a potential area for a future study.

Acknowledgements

This research is funded by The Scientific and Technological Research Council of Turkey (TUBITAK), under grant #213M253.

References

[1] M. Vanhoucke and D. Debels, “The discrete time/cost trade-off problem: extensions and heuristic procedures,” *J. Sched.*, vol. 10, no. 4, pp. 311–326, 2007.

[2] P. De, E. J. Dunne, J. B. Ghosh, and C. E. Wells, “Complexity of the Discrete Time-Cost Tradeoff Problem for Project Networks,” *Oper. Res.*, vol. 45, no. 2, pp. 302–306, 1997.

[3] J. Moussourakis and C. Haksever, “Flexible Model for Time/Cost Tradeoff Problem,” *J. Constr. Eng. Manag.*, vol. 130, no. 3, pp. 307–314, 2004.

[4] N. Siemens, “A Simple CPM Time-Cost Tradeoff Algorithm,” *Manage. Sci.*, vol. 17, no. 6, p. B–354–B–363, 1971.

[5] S. K. Goyal, “A Note on ‘A Simple CPM Time-Cost Tradeoff Algorithm,’” *Manage. Sci.*, vol. 21, no. 6, pp. 718–722, 1975.

[6] O. Moselhi, “Schedule compression using the direct stiffness method,” *Can. J. Civ. Eng.*, vol. 20, no. 1, pp. 65–72, 1993.

[7] T. Hegazy, “Optimization of construction time-cost trade-off analysis using genetic algorithms,” *Can. J. Civ. Eng.*, vol. 26, no. 6, pp. 685–697, 1999.

[8] R. Sonmez and Ö. H. Bettemir, “A hybrid genetic algorithm for the discrete time–cost trade-off problem,” *Expert Syst. Appl.*, vol. 39, no. 13, pp. 11428–11434, Oct. 2012.

[9] H. Zheng, “Proceedings of the Ninth International Conference on Management Science and Engineering Management,” J. Xu, S. Nickel, C. V. Machado, and A. Hajiyev, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 337–346.

[10] L. Zhang, X. Zou, and J. Qi, "A trade-off between time and cost in scheduling repetitive construction projects," *J. Ind. Manag. Optim.*, vol. 11, no. 4, pp. 1423–1434, 2015.
[11] E. Elbeltagi, T. Hegazy, and D. Grierson, "A modified shuffled frog-leaping optimization algorithm: applications to project management," *Struct. Infrastruct. Eng.*, vol. 3, no. 1, pp. 53–60, 2007.
[12] A. Abdel-Raheem, M. and Khalafallah, "Using Electimize to Solve the Time-Cost-Tradeoff Problem in Construction Engineering," in *Computing in Civil Engineering*, 2011, pp. 250–257.
[13] M. Tavana, A.-R. Abtahi, and K. Khalili-Damghani, "A new multi-objective multi-mode model for solving preemptive time–cost–quality trade-off project scheduling problems," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1830–1846, Mar. 2014.
[14] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Adv. Eng. Informatics*, vol. 19, no. 1, pp. 43–53, Jan. 2005.
[15] O. H. Bettemir, "Optimization of time-cost-resource trade-off problems in project scheduling using meta-heuristic algorithms," Middle East Technical University, 2009.
[16] W. L. . Meyer and L. R. Shaffer, "Extensions of the Critical Path Method Through the Application of Integer Programming," Urbana, Illinois., 1963.
[17] G. L. T. W. Crowston, "Decision CPM: A Method for Simultaneous Planning, Scheduling, and Control of Projects," *Oper. Res.*, vol. 15, no. 3, pp. 407–426, 1967.
[18] W. B. Crowston, "Decision CPM: Network Reduction and Solution," *Oper. Res. Q.*, vol. 21, no. 4, pp. 435–452, 1970.
[19] R. T. Harvey and J. H. Patterson, "No TitlAn Implicit Enumeration Algorithm for the Time/Cost Trade-off Problem in Project Network Analysis," *Found. Control Eng.*, no. 4, pp. 105–117, 1979.
[20] C. Liu, L., Burns, S., and Feng, "Construction Time-Cost Trade-Off Analysis Using LP/IP Hybrid Method," *J. Constr. Eng. Manag.*, vol. 121, no. 4, pp. 446–454, 1995.
[21] M. Vanhoucke, "New computational results for the discrete time/cost trade-off problem with time-switch constraints," *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 359–374, Sep. 2005.
[22] C. Akkan, A. Drexl, and A. Kimms, "Network decomposition-based benchmark results for the discrete time-cost tradeoff problem," *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 339–358, 2005.
[23] Ö. Hazır, E. Erel, and Y. Günalay, "Robust optimization models for the discrete time/cost trade-off problem," *Int. J. Prod. Econ.*, vol. 130, no. 1, pp. 87–95, Mar. 2011.
[24] J. G. Szmerekovsky and P. Venkateshan, "An integer programming formulation for the project scheduling problem with irregular time–cost tradeoffs," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1402–1410, Jul. 2012.
[25] M. J. Liberatore, B. Pollack-Johnson, and C. A. Smith, "Project management in construction: Software use and research directions," *J. Constr. Eng. Manag.*, vol. 127, no. 2, pp. 101–107, 2001.
[26] C.-W. Feng, L. Liu, and S. a. Burns, "Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems," *J. Comput. Civ. Eng.*, vol. 11, no. 3, pp. 184–189, 1997.
[27] C. Schwindt, "ProGen/max: A new problem generator for different resource-constrained project scheduling problems with minimal and maximal time lags. Report WIOR-449," p. 57, 1995.
[28] P. De, E. James Dunne, J. B. Ghosh, and C. E. Wells, "The discrete time-cost tradeoff problem revisited," *Eur. J. Oper. Res.*, vol. 81, no. 2, pp. 225–238, 1995.