Sigma J Eng & Nat Sci 7 (1), 2016, 1-7

Publications Prepared for Transist 2015,
8th International İstanbul Transport Congress
Transist 2015, 8. Uluslararası İstanbul
Ulaşım Kongresi için Hazırlanan Yayınlar

Research Article / Araştırma Makalesi

AUTOMATED VEHICLE SCHEDULING SYSTEM: A CASE STUDY OF METROBUS SYSTEM

İsmail SEVİM*[1], Hatice TEKİNER MOĞULKOÇ[2], Mehmet Güray GÜLER[1], Zeynep Pınar MUTLU[3]

[1]*Yildiz Technical University, Department of Industrial Engineering, Istanbul/TURKEY*
[2]*Istanbul Sehir University, Department of Industrial Engineering, Istanbul/TURKEY*
[3]*Istanbul Electricity, Tramway, and Tunnel General Directorate (IETT), Istanbul/TURKEY*

Received/Geliş: 14.03.2016 Accepted/Kabul: 03.04.2016

ABSTRACT

Multiple Depot Vehicle Scheduling Problem (MDVSP) is the problem of preparing vehicle schedules, a task of public transport companies. MDVSP is solved to decide on daily duties of vehicles emanating from multiple depots. It aims to minimize number of vehicles used and total deadhead kilometers. Since there are large number of trips to be covered by vehicles emanating from multiple depots, manually prepared vehicle schedules are usually far from optimality. Therefore, using automatic scheduling systems can reduce number of vehicles used and total deadhead kilometers. In this study, 6254 daily trips belong to 2014-2015 Winter Timetable of IETT General Directorate's Metrobus System are assigned to 476 vehicles instead of existing 496 vehicles by solving a Single Depot Vehicle Scheduling Problem (SDVSP) model, actually a reduced model of MDVSP. As further study, Lagrangian relaxation is going to be used to solve MDVSP to minimize total deadhead kilometers.
Keywords: Multiple depot vehicle scheduling problem, lagrangian relaxation, binary integer programming, public transport planning.

OTOMATİK ARAÇ ÇİZELGELEME SİSTEMİ: METROBÜS SİSTEMİ İÇİN BİR ÖNERİ

ÖZ

Çok Garajlı Araç Çizelgeleme Problemi (ÇGAÇP), toplu ulaşım işletmelerinin hazırlamakla yükümlü oldukları araç çizelgelerinin oluşturulmasında karşılaşılan bir problemdir. Birden fazla garajda park etmekte olan araçlardan her birinin günlük sefer tarifesinde yer alan servislerden hangilerini gerçekleştireceğine karar vermek için çözülmektedir. Gün içerisinde kullanılan araç sayısını ve yapılan ölü kilometreyi enküçüklemeyi amaçlar. Uygulamada çok sayıda servis, garaj ve otobüs olmasından dolayı elle üretilen araç çizelgeleri optimumdan uzak olmaktadır. Bu nedenle bu çizelgelerin bilgisayar yardımıyla hazırlanması hem araç sayısı hem de ölü kilometre maliyetlerinden tasarruf edilmesini sağlayacaktır. Bu çalışma kapsamında Metrobüs Sistemi için planlanan kış-işgünü sefer çizelgelerinde yer alan 6254 sefer için ÇGAÇP, Tek Garajlı Araç Çizelgeleme Problemi'ne (TGAÇP) dönüştürülmüş, aynı sayıda servisin Kurum'un kullandığı 496 araca karşılık 476 araçla yapılabileceği tespit edilmiştir. İkinci aşama olaraksa Lagrangian gevşetmesinden yararlanılarak ÇGAÇP çözülecek, ölü kilometre enküçüklemesi de sağlanacaktır.
Anahtar Sözcükler: Çok garajlı araç çizelgeleme problemi, lagrangian gevşetmesi, ikili programlama, toplu ulaşım planlama.

* Corresponding Author/Sorumlu Yazar: e-mail/e-ileti: ismailsevi@gmail.com, tel: (212) 383 28 65

1. INTRODUCTION

Public transport companies have to prepare four plans. These are: (1) network planning, (2) timetabling, (3) vehicle scheduling, (4) and driver scheduling [1]. Also, driver rostering may be added to this list as a fifth plan. Each of these plans are defined as a branch of public transport planning.

Vehicle scheduling, as a medium-term plan, deals with deciding on daily duties of vehicles belong to a public transport company. It is desired to ensure that all timetabled trips are covered by minimum number of vehicles and total deadhead kilometers while scheduling vehicles. Cost improvements in number of vehicles may occur up to amount of 8% where cost improvement in total deadhead kilometers may occur up to amount of 6% through efficient scheduling practices [2]. When the case is large size public transport companies of crowded cities, amounts of cost improvements in terms of dollars may add up to high figures.

Scheduling vehicles of Metrobus System governed by IETT (Istanbul Electricity, Tramway, and Tunnel) General Directorate is done manually. This fact indicates that there exist cost improvement opportunities in assigning vehicle scheduling task to a computer and developing a suitable scheduling software. On the other hand, scheduling staff of Metrobus System denote that timetabling activities are done by number of available vehicles. Therefore, reducing number of vehicles to cover the trips of same timetable also allows adding more trips into the timetable and increases customer satisfaction. Thus, minimizing number of vehicles is more beneficial than minimizing total deadhead kilometers.

Cost improvement opportunities by doing vehicle scheduling automatically are appraised in this study. Accordingly, assigning 6254 trips to vehicles emanating from 4 depots are modelled as Multiple Depot Vehicle Scheduling Problem (MDVSP) and feasibility of this model is checked by solving a Single Depot Vehicle Scheduling Problem (SDVSP) equivalent.

In this study, MDVSP and related solution methodologies found in literature are discussed in Chapter 2. Then, mathematical models of MDVSP and SDVSP are given in Chapter 3. Daily duties of vehicles of Metrobus System are decided by using such a methodology explained in Chapter 4 and results of the application are given in Chapter 5. Finally, Chapter 6 has concluding remarks.

2. MULTIPLE DEPOT VEHICLE SCHEDULING PROBLEM

MDVSP is the problem of assigning n number of timetabled trips to vehicles emanating from m number of depots. Three rules must be followed while assigning trips to vehicles [3]:

1. Each trip must be covered by only one vehicle.
2. A suitable set of constraints must be satisfied.
3. A suitable objective function must be optimized.

First of these rules must be satisfied obviously. Suitable set of constraints mentioned at second rule may differ for different problems. However, compatibility constraints and depot capacity constraints are common for all problems. At the same time, route time constraints [2], and constraints related to multi-vehicle type usage [4] are also found in literature. Two cost components are aimed to be minimized while solving MDVSP: (1) Number of vehicles, and (2) total deadhead kilometers.

If number of depots $m = 1$ in MDVSP, then the problem is called SDVSP. SDVSP can be solved in polynomial-time [5] where MDVSP is an NP-Hard problem [3]. This means that there is no polynomial-time algorithm to solve MDVSP. Although this fact doesn’t constitute a serious problem for smaller MDVSP instances, when the problem size increases existing optimization software applications lose their ability to solve such problems. Thus, instance-specific exact optimization methods are needed to be devised [2] [6]. Heuristic and metaheuristic methods are also showed to be effective for MDVSP.

When it is impossible to obtain an optimal solution, heuristic and metaheuristics are used to find a feasible solution. However, it is not ensured that these feasible solutions are optimal. In [7], performance of three different ant colony algorithms are compared on solving MDVSP with route time constraints. MDVSP is also modelled as an asymmetric travelling salesman problem and solved by an ant colony algorithm [8]. A two-phase particle-swarm optimization is used to solve the problem in [9]. It is also showed that iterated local search based on block move neighborhood [10] is used effectively to solve MDVSP [11]. Another local search metaheuristic solution may also be found at [12]. On the other hand, five different heuristics are compared on solving MDVSP and it is noted that large neighborhood search is superior to tabu search and genetic algorithms [13]. In [2], [14], and [15] size-reduction heuristics are used to make the problem smaller enough to be solved by existing optimization software applications.

MDVSP is modelled as integer program [16], multi-commodity flow network model [6], multi-commodity matching problem [3], and time-space network [15]. When MDVSP is modelled as network flows, usually two main approaches are utilized. Arc-oriented models have principle of deciding on consecutive trips where path-oriented models solve MDVSP by deciding on all of the trips assigned to a schedule. In [2], MDVSP is modelled as an arc-oriented model and one exact and two heuristic approaches are offered. A path-oriented model is solved by a column generation approach [17]. A large sized instance modelled as an arc-oriented model is solved to optimality by using column generation [6].

3. MATHEMATICAL MODELS

In this study, the MDVSP is modeled as a multi-commodity flow network where each depot is treated as a commodity. Nodes of the network is classified into three groups: (a) trip nodes, (b) starting nodes, and (c) ending nodes.

(a) Each trip node corresponds to a timetabled trip. A node has four characteristics: (1) starting station of the trip, (2) ending station of the trip, (3) starting time of the trip, and (4) ending time of the trip.

(b) Starting nodes represents the depots (one node for each depot). These nodes help on deciding first trips of vehicle schedules.

(c) Ending nodes represents the depots (one node for each depot). These nodes help on deciding last trips of vehicle schedules.

Arcs of the directed graph are grouped into three categories: (1) pull-out arcs connect starting nodes to trip nodes, (2) deadhead trip arcs connect nodes of compatible trips, and (3) pull-in arcs connect trip nodes to ending nodes. Note that deadhead trip arcs are replicated to number of depots. Below binary integer program is given to model the network defined above. Model can be found at [18]. Let C be the set of compatible trips (If $(i,j) \in C$, then a vehicle can cover trip j immediately after trip i).

$$(P): \quad \min \quad \sum_d \sum_i l_{d,i} L_{d,i} + \sum_i \sum_j \sum_d c_{i,j,d} x_{i,j,d} + \sum_d \sum_i a_{d,i} A_{d,i} \tag{1}$$

$$s.t. \quad \sum_j L_{d,i} \le r_d \qquad \forall d, \tag{2}$$

$$L_{d,i} + \sum_j x_{j,i,d} - y_{i,d} = 0 \qquad (j,i) \in C \quad \forall i, d, \tag{3}$$

$$A_{i,d} + \sum_j x_{i,j,d} - y_{i,d} = 0 \qquad (i,j) \in C \quad \forall i, d, \tag{4}$$

$$\sum_j A_{i,d} \le r_d \qquad \forall d, \tag{5}$$

$$\sum_d y_{i,d} = 1 \qquad \forall i, \tag{6}$$

$$All\ variables\ are\ binary. \tag{7}$$

Variable definitions of model (P) are given below.

- $L_{d,i}$: It is equal to 1 if trip i is the first trip of a vehicle emanating from depot d and 0 elsewhere.
- $x_{i,j,d}$: It takes a value of 1 if a vehicle emanating from depot d covers trip j immediately after trip i and 0 elsewhere.
- $A_{i,d}$: It is equal to 1 if trip i is the last trip of a vehicle emanating from depot d and 0 elsewhere.
- $y_{i,d}$: It takes a value of 1 if trip i is covered by a vehicle emanating from depot d.

Parameters of model (P) are given below.

- $l_{d,i}$: Cost of deadhead trip from depot d to the starting station of trip i (pull-out costs).
- $c_{i,j,d}$: Cost of deadhead trip from ending station of trip i to the starting station of trip j. Note that this cost is equal for each d.
- $a_{i,d}$: Cost of deadhead trip from ending station of trip i to depot d (pull-in costs).
- r_d: Maximum number of vehicles depot d provides (depot capacities).

Objective function aims to minimize total deadhead kilometers. Fixed cost of a vehicle is added to each pull-out arc and pull-in arc costs. If fixed cost is adequately large, then aim of minimizing number of vehicles is prioritized. Constraints (2) and (5) are capacity constraints. Where constraints (3) and (4) are flow conservation constraints of multi-commodity network models. Constraints (6) ensures that each trip is covered by only one vehicle emanating from only one depot. Finally, constraints (7) are binary constraints.

It is already denoted above that when number of depot $d = 1$, MDVSP is called as SDVSP. Let C be the set of compatible trips and M is an adequately large number. SDVSP model is given below.

$$(T) \quad \min \quad \sum_i l_i L_i + \sum_i \sum_j c_{i,j} x_{i,j} + \sum_i a_i A_i \tag{8}$$

$$s.t. \quad \sum_j L_i \leq M, \tag{9}$$

$$L_i + \sum_j x_{j,i} = 1 \qquad (j,i) \in C \quad \forall i, \tag{10}$$

$$A_i + \sum_j x_{i,j} = 1 \qquad (i,j) \in C \quad \forall i, \tag{11}$$

$$\sum_j A_i \leq M, \tag{12}$$

$$All\ variables\ are\ binary \tag{13}$$

Variable definitions of model (T) are given below.

- L_i: It takes a value of 1 if trip i is a first trip of a vehicle schedule and 0 elsewhere.
- $x_{i,j}$: It is equal to 1 if a vehicle covers trip j immediately after trip i and 0 elsewhere.
- A_i: It takes a value of 1 if trip i is a last trip of a vehicle schedule and 0 elsewhere.

Parameters of model (T) are given below.

- l_i: Cost of deadhead trip from depot to starting station of trip i (pull-out costs).
- $c_{i,j}$: Cost of deadhead trip from ending station of trip i to starting station of trip j.
- a_i: Cost of deadhead trip from ending station of trip i to depot (pull-in costs).

Objective function aims to minimize total deadhead kilometers. A fixed cost for a vehicle is added to pull-out and pull-in arc costs. If fixed cost of a vehicle is adequately large, then aim of minimizing number of vehicles is prioritized as it is applied in model (P). Constraints (9) and (12) are capacity constraints where constraints (10) and (11) are flow conservation constraints of network flows. Finally, constraints (13) are binary constraints. Since number of depots $d = 1$ for SDVSP, equivalent of constraints (6) of model (P) are redundant for model (T).

4. METHODOLOGY

The fixed cost of a vehicle is added to pull-out and pull-in arc costs. The aim of minimizing number of vehicles is prioritized by assigning the fixed cost to an adequately large number. Therefore, whether number of depots equal to $|M|$ or $|k|$ it is assured that minimum number of vehicles are equal for each solution where M is set of depots and k is a set such that $k \subset M$. For instance, if one of the depots is chosen as a single depot from a MDVSP instance, then minimum number of vehicles obtained from SDVSP is equal to minimum number of vehicles of MDVSP solution. However, same idea is not true for minimum total deadhead kilometers.

Since MDVSP is a NP-Hard problem, it may be impossible to obtain optimum solutions of large instances. Fortunately, if aim of minimizing number of vehicles is prioritized it is possible to obtain a solution that provides minimum number of vehicles by solving a reduced MDVSP which is actually a SDVSP. If such priority exists then SDVSP is solved. Since minimum number of vehicles obtained from a SDVSP solution is same for MDVSP, feasibility of MDVSP and possible cost improvements may be provided by SDVSP solution.

In this study, the reduction strategy above is used to find the minimum number of vehicles for operating Metrobus System governed by IETT General Directorate. The developed vehicle schedules are compared to existing vehicle schedules to study feasibility of problem and detect possible cost improvement opportunities. Optimization models are solved by GUROBI® Solver on a laptop with an Intel® Core™ i7-4510U CPU @ 2.00 GHz processor with 6.00 GB RAM on a Microsoft® Windows® 64 bit operating system.

5. APPLICATION

2014-2015 Winter Timetable belongs to seven lines of Metrobus System are used for MDVSP application. The timetable contains 6254 trips daily. Line information and frequencies are given in Table 1.

6254 trips belong to 2014-2015 Winter Timetable are covered by 496 vehicles emanating from 4 different depots. Depot names and physical characteristics are given in Table 2.

Table 1. Metrobus lines and total number of daily trips

Line	Starting Station 1	Starting Station 2	Total Number of Trips (Direction 1->2)	Total Number of Trips (Direction 2->1)
34AS	Avcilar	Sogutlucesme	778	779
34	Avcilar	Zincirlikuyu	322	328
34BZ	Beylikduzu	Zincirlikuyu	887	896
34C	Beylikduzu	Cevizlibag	359	350
34G	Beylikduzu	Sogutlucesme	38	37
34Z	Zincirlikuyu	Sogutlucesme	738	583
34U	Uzuncayir	Zincirlikuyu	159	-

Table 2. Depots serve Metrobus System

Depot	Area (m²)	Closed Area (m²)
Ikitelli	192.000	28.000
Edirnekapi	60.000	6.720
Hasanpasa	37.000	4.000
Anadolu	58.200	10.000

Vehicle schedules of Metrobus System are prepared by using SDVSP model obtained as explained in Chapter 4. Minimum number of vehicles to cover 6254 trips of 2014-2015 Winter Timetable are found. Aforementioned SDVSP model has 16,107,698 binary variables and 18,164 constraints. CPU times for building the model and finding optimal solution is given in

Table 3

Table 3. CPU times for model building and solution of SDVSP model

Network Building (sec)	Model Building	Solution (sec)
133.24	>4 days	2400.8

It is shown that 6254 daily trips of 2014-2015 Winter Timetable of Metrobus System of IETT General Directorate can be covered by 476 vehicles by solving given mathematical model.

6. CONCLUSION

In this study, it is shown that daily trips of 2014-2015 Winter Timetable of Metrobus System can be covered by 476 vehicles instead of existing 496 vehicles by solving a multi-commodity type MDVSP model. Solution is obtained by reducing MDVSP model to a SDVSP model. Since average purchasing cost of a vehicle is equal to ₺1,100,000 total cost reduction in number of vehicles incurs at amount of ₺20,900,000. Amount of cost reduction may become higher than ₺ 20,900,000 by incorporating indirect costs into cost reduction calculations.

SDVSP solution indicates that corresponding MDVSP model is feasible. Therefore, MDVSP model is going to be solved to optimality and aim of the total deadhead kilometers are also going to be minimized. As further study, it is planned to use Lagrangian relaxation strategy similar to one in [3]. Constraint (6) are going to be relaxed in model (P) to obtain and solve 4 separate SDVSP models. It is planned to find possible cost improvements in terms of total deadhead kilometers by using such a solution strategy.

REFERENCES / KAYNAKLAR

[1] Ceder A., Stern H. I., (1981) Deficit function bus scheduling with deadheading trip insertions for fleet size reduction, *Transportation Science,* 338-363.

[2] Haghani A., Banihashemi M., (2002) Heuritic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints, *Transportation Research Part A,* 309-333.

[3] Bertossi A., A., Carraresi P., Gallo G., (1987) On some matching problems arising in vehicle scheduling models, *Networks,* 271-281.

[4] Ceder A., (2011) Public-transport vehicle scheduling with multi vehicle type, *Transportation Research Part C,* 485-497.

[5] Bunte S., Kliewer N., (2009) An overview on vehicle scheduling models, *Public Transport,* 299-317.

[6] Löbel A., (1998) Vehicle scheduling in public transit and Lagrangean Pricing, *Management Science,* 1637-1649.

[7] Wie M., Jin W., Fu W., Xiao-ni H., (2010) Improved ant colony algorithm for multi-depot bus scheduling problem with route time constraints, *8th World Congress on Intelligent Control and Automation*, *WCICA 2010*, 07-09 July 2010, Jinan, China.

[8] Ramos J. A., Reis L. P., Pedrosa D., (2011) Solving heterogeneous fleet multiple depot vehicle scheduling problem as an asymmetric traveling salesman problem, *Progress in Artificial Intelligence,* 98-109.

[9] Wang S., Wang L., Yuan H., Ge M., Niu B., Pang W., Liu Y., (2009) Study on multi-depots vehicle scheduling problem and its two-phase particle swarm optimization, *Emerging Intelligent Computing Technology and Applications,* 748-756.

[10] Laurent B., Hao J. K., (2007) A study of neighborhood structures for the multiple depot vehicle scheduling problem, *Engineering Stochastic Local Search Algorithms,* 197-201.

[11] Laurent B., Hao J. K., (2009) Iterated local search for the multiple depot vehicle scheduling problem, *Computers & Industrial Engineering ,* 277-286.

[12] Otsuki T., Kazuyiki A., (2014) New variable depth local search for multiple depot vehicle scheduling problems, *Journal of Heuristics,* 1-19.

[13] Pepin A. S., Desaulniers G., Hertz A., (2009) A comparison of five heuristics for the multiple depot vehicle scheduling problem, *Journal of Scheduling,* 17-30.

[14] Guedes P. C., Lopes W. P., Rohde L. R., Borenstein D., (2015) Simple and efficient heuristic approach for the multiple-depot vehicle scheduling problem, *Optimization Letters,* 1-13.

[15] Gintner V., Kliewer N., Suhl L., (2005) Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice, *OR Spectrum,* 507-523.

[16] Desaulniers G., Lavigne J., Soumis F., (1998) Multi-depot vehicle scheduling problems with time windows and waiting costs, *European Journal of Operational Research,* 479-494.

[17] Ribeiro C. C., Soumis F., (1994) A column generation approach to the multiple-depot vehicle scheduling problem, *Operations Research,* 41-52.

[18] Forbes M. A., Hotts J. N., Watts A. M., (1994) An exact algorithm for multiple depot bus scheduling, *European Journal of Operational Research ,* 115-124.