Automated Driver Scheduling for Vehicle Delivery

Shashika R. Muramudalige(✉) and H. M. N. Dilum Bandara

Department of Computer Science and Engineering,
University of Moratuwa, Katubedda, Moratuwa 10400, Sri Lanka
{shashika.10,dilumb}@cse.mrt.ac.lk

Abstract. Vehicle delivery is a major business where third-party drivers are hired to deliver vehicles when they are relocated, sold, or while returning rental cars. This is a complex process due to the wide variation in collection/delivery locations, time bounds, types of vehicles, special skills required by drivers, and impact due to traffic and weather. We propose an automated driver scheduling solution to maximize the number of vehicle deliveries and customer satisfaction while minimizing the delivery cost. Proposed solution consists of a rule checker and a scheduler. Rule checker enforces constraints such as deadlines, license types, skills, and working hours. Scheduler uses simulated annealing to assign as many jobs as possible while minimizing the overall cost. Using a workload derived from an actual vehicle delivery company, we demonstrate that the proposed solution has good coverage of jobs while minimizing the cost and having flexibility to tolerate breakdowns, excessive traffic, and bad weather.

Keywords: Scheduling · Simulated annealing · Vehicle Delivery

1 Introduction

While taxis and rental cars are well-known services, there are other forms of B2B services such as Vehicle Delivery (VD) using third-party drivers. When a customer wants to move a vehicle (e.g., due to a sale or to return a rental car), he/she inquires a VD service to identify a suitable driver and a schedule. This is motivated due to the busy schedules, convince, and cost saving. A VD company typically operates in a chosen geography varying from a region of a country to a set of nearby countries. Hence, the drivers are also geographically dispersed. The VD company needs to allocate a job to the most suitable driver based on a set of parameters such as vehicle collection and delivery location and time, type of vehicle, drivers' location and availability, labor laws, traffic, and weather constraints. Moreover, driver allocation should focus on increasing customer satisfaction, operational efficiency, and company profit. This is a dynamic environment where the schedules may change or even canceled due to reasons such as customer changing the pickup/delivery time, canceling a job, arrival of a last-minute job from a high-priority customer, vehicle breakdown, excessive traffic, or driver unavailability due to a sickness. Therefore, driver scheduling in VD is also a complex problem, though the volumes are not as high as taxies or rental cars.

T. Kováčiková et al. (Eds.): INTSYS 2017, LNICST 222, pp. 215–225, 2018.
https://doi.org/10.1007/978-3-319-93710-6_23

Currently the scheduling is mostly manipulated manually by an experienced scheduling manager, who creates the next day's schedule at the end of the previous working day based on the jobs received. The scheduling manager also needs to keep track of the progress of jobs (usually by calling drivers and customers) and make necessary adjustments due to dynamism as the day progresses. However, as the number of jobs and drivers increase, it becomes difficult to decide on the most appropriate driver for a job such that both the customer and company goals are optimally satisfied. Moreover, last-minute schedule changes could trigger a chain reaction to subsequent jobs. Therefore, the industry is in need for scalable and automated scheduling solutions that increase customer satisfaction, efficiency, and profits. However, as the driver, route, and vehicle scheduling problems are known to be NP hard, we cannot get the optimal solution within polynomial time [1–3]. Therefore, the objective should be to identify a suitable heuristic-based solution that can still maximize the customer satisfaction, efficiency, driver earnings, and profit.

An automated driver and vehicle scheduling solution for a limousine renting company is proposed in [4]. It used a two-phase algorithm where a constrained model is first used to get an initial solution, which is then optimized using simulated annealing. Column generation based hyper-heuristic solution in [5] addressed the bus-driver scheduling problem. Authors demonstrated that their solution is more scalable and outperform all the other well-known scheduling algorithms. Feasibility of using artificial neural systems for delivery truck scheduling using a small scale, dynamic routing and scheduling problem is presented in [6]. In [7], using a real dataset, authors demonstrated that ready-mix concrete truck dispatching can be accurately automated using decision trees and k-nearest neighbor techniques than neural network and support vector machine based techniques. Detailed discussion of the truck driver scheduling problem is presented in [8]. However, there are several notable differences between the VD and other delivery problems. For example, in VD problem both the vehicles and drivers are geographically dispersed, where as in limousine renting and bus scheduling problems vehicles and drivers are dispatched from a specific depot. Moreover, in VD problem most jobs are one way, types of vehicles and driver skills required to operate them drastically vary, some vehicles are not delivered on the same day as pickup (depending on distance and receiver availability), more flexible pickup/delivery times, and drivers are willing to work only on chosen geographies and times. Therefore, it is essential to formulate the optimization problem for the specific domain and devise a more fitting solution.

We propose an automated, driver scheduling solution to cover all the jobs while maximizing the customer satisfaction and company profit. Our solution is inspired by [4]; however, it is significantly extended to capture the domain differences and complexity of scheduling. The proposed solution consists of a rule checker and a scheduler. Rule checker enforces constraints such as deadlines, license types, driver skills, and labor laws (e.g., number of working hours per day/week) while assigning a job. Scheduler uses simulated annealing to assign as many jobs as possible while minimizing the overall cost. Using a multi-day workload derived from a real VD company, we demonstrate that the proposed solution has good coverage of jobs while minimizing the cost and having flexibility to tolerate unexpected events such as breakdowns, traffic, and bad weather.

The rest of the paper is organized as follows. Problem formulation is presented in Sect. 2. Rule checker and scheduler are presented in Sect. 3. Performance analysis and concluding remarks are presented in Sects. 4 and 5, respectively.

2 Problem Formulation

Let **J** be the set of VD jobs, where each job $j \in \mathbf{J}$ has a pickup and delivery location and time, type of vehicle, and a preferred set of drivers (typically for recurrent jobs). These jobs are to be processed by a set of drivers **D**, where each driver $d \in \mathbf{D}$ has a set of required skills and is licensed to drive a set of vehicle categories. A driver has a home location, preference to work only on a set of geographies, preferred working days and times (e.g., part-time drivers), and is willing to accept only a given number of jobs per day/week. Moreover, regulatory requirements such as maximum number of driving/working hours a day and days per week/month need to be met. Let f_j be the fee for job j, which depends on the distance between pickup and delivery locations. Whereas cost per driver d assigned for job $j(c_d^j)$ depends on the distance driven, driver's skill level, and driver's personal expenses to reach the pickup location and return to the next job's pickup location or home. Our objective is to cover all jobs **J** with drivers **D**, such that $f_j - c_d^j$ is maximized across all the jobs. Next, we describe each of the parameters (listed in Table 1) and constraints in details, and then formulate the optimization problem.

Table 1. List of symbols.

Symbol	Description
d_{id}	Driver ID
d_{type}	Set of vehicle types d can drive with a valid driving license
$d_{categories}$	Set of vehicle categories d can drive based on a special training
$d_{time\_max\_day}/d_{time\_max\_week}$	Maximum allowed driving time per day/week for driver d
d_{work_time}	Working days and hours of driver d
d_{job_areas}	Set of preferred geographies driver d is willing to accepts jobs
d_{job_count}	No of jobs assigned so far to driver d on a given day
$d_{max\_jobs\_day}$	Maximum number of jobs driver d prefers to handle on a day
$d_{drive\_time\_day}/d_{drive\_time\_week}$	Total driving time of driver d for a given day/week
f_j	Fee charged from customer for job j (depends on job distance)
j_{id}	Job ID
j_{type}	Vehicle type demanded by job j
$j_{category}$	Vehicle category demanded by job j
$j_{pickup\_loc}/j_{delivery\_loc}$	Vehicle pickup/delivery location of job j
$j_{pickup\_time}/j_{delivery\_time}$	Vehicle pickup/delivery day/time of job j
$j_{preferred\_drivers}/j_{excluded\_drivers}$	Customers may specify a set of drivers that they prefer/don't prefer to work with based on the past experiences
j_{travel_time}	Estimated travel time for job j

2.1 Constraints

To be eligible for a job j, a driver d need to satisfy the following set of constraints:

Vehicle Type Constraint – Driver d must have a valid driving license to drive the vehicle type defined in job j, for e.g., car, van, or truck. Therefore,

$$j_{type} \in d_{type} \rightarrow d \tag{1}$$

Vehicle Category Constraint – Some jobs demand specific driver skills and trainings, e.g., specific training may be required to operate certain luxury and heavy vehicles. Moreover, while delivering a high-end car or special-purpose vehicle to a new buyer, the manufacturer may train the driver on customer service and vehicle maintenance tips. Therefore, d must have trainings and skills to take the job:

$$j_{category} = \phi \cup j_{category} \in d_{categories} \rightarrow d \tag{2}$$

Number of Jobs Constraint – As the VD company may also rely on part-time drivers, drivers have flexibility to mention how many jobs they prefer to complete per day, without considering the complexity of jobs or income. Therefore, for a given day

$$d_{job\_count} + 1 \leq d_{max\_jobs\_day} \rightarrow d \tag{3}$$

Job Area Constraint – Drivers may mention their preferred area of work. Therefore, the driver should be assigned to jobs only within his/her preferred area:

$$j_{pickup\_loc} \subseteq d_{job\_areas} \cap j_{delivery\_loc} \subseteq d_{job\_areas} \rightarrow d \tag{4}$$

Driver Availability Constraint – Drivers have the flexibility to work on any day any time, hence their availability should match the job's timeline as follows:

$$j_{pickup\_time} \in d_{work\_time} \cap j_{deliver\_time} \in d_{work\_time} \rightarrow d \tag{5}$$

Travel Time Constraint – For regulatory purposes, usually there are limits on how many hours a driver can drive per day and a week. Therefore, for a given day and a week, total driving time of d and estimated driving time of the new job j should be within the maximum allowed:

$$\begin{aligned} d_{drive\_time\_day} + j_{travel\_time} \leq d_{time\_max\_day} \cap d_{drive\_time\_week} \\ + j_{travel\_time} \leq d_{time\_max\_week} \rightarrow d \end{aligned} \tag{6}$$

where estimated travel time can be more accurately calculated today using services such as the Google Maps API based on pickup/delivery location and time, traffic, and weather. Therefore, we define a job's travel time as follows:

$$j_{travel\_time} = ttf(j_{pickup\_loc}, j_{delivery\_loc}, j_{pickup\_time}) \tag{7}$$

Feasible Sequences Constraint – Once a job is completed, it may not be possible to start the next job immediately, therefore, a driver needs sufficient time to travel to the pickup location of the next job. This depends on many parameters such as delivery time of previous job, availability of public transportation, and traffic. Therefore,

$$j_{next\_pickup\_time} = j_{delivery\_time} + ttf(j_{delivery\_loc}, j_{delivery\_time}, j_{next-pickup\_loc}) \quad (8)$$

Pairing Constraint – Frequent customers may specify a list of preferred or excluded drivers based on the past experiences. Therefore, a preferred driver should be assigned when possible, while a driver in the excluded list of a job should not be assigned at all. Thus, the following constraints need to be satisfied:

$$d_{id} \notin j_{excluded\_drivers} \rightarrow d \quad (9)$$

$$d_{id} \in j_{preferred\_drivers} \rightarrow d \quad (10)$$

However, a preferred driver cannot be assigned under all costs, as cost minimization is an objective. Therefore, (10) is enforced only when the cost of assigning a preferred driver is within a reasonable difference (say λ) from the other drivers with less cost.

2.2 Objectives

Given **J** and **D**, our main objective is to cover as many jobs as possible. This is required to improve customer satisfaction as some of the customers are engaged in a long-term business relationship. We further satisfy a secondary objective, namely to maximize VD company's overall profit. To minimize the cost, it is imperative to find the most appropriate driver who can do the job within the customer requested time frame with minimum cost. Drivers should not be idle, as they do not make any income when they are not driving. Hence, their income should be proportional to their availability. Hence, driver earning should be good, else they are likely to leave the company. Therefore, the objective function can be formulated as follows:

$$\forall j \in \mathbf{J}, \forall d \in \mathbf{D}\, \mathrm{Max}(|j \text{ with assigned } d|) \quad (11)$$

$$\mathrm{Max} \sum_{j \in J, d \in D} f_j - c_d^j \quad (12)$$

$$d_{income} \propto d_{work\_time} \quad (13)$$

Equation 11 maximize the number of jobs with an assigned driver. Profit is given as $f_j - c_d^j$.

3 Solution Approach

As in many scheduling problems, we assume that every evening, the next day's schedule is determined based on the already confirmed jobs and available drivers. It is difficult to estimate the job end time because it depends on operational inefficiencies, traffic, and weather, which are hard to predict on a given day. While we investigate the flexibility of the automatically generated schedule in Sect. 4, dynamic scheduling to overcome last-minute changes is left as future work. Our solution consists of a *rule checker* that enforces the driver constraints and an *optimization phase* that attempts to cover as many jobs as possible while minimizing the overall cost. For optimization, we choose Simulated Annealing (SA) algorithm because it is used in global optimization problems in a wide range of areas including limousine renting in [4]. SA provides a reasonably optimized solution within a reasonable time, and can be optimized according to the context [9].

Given **J** and **D**, *rule checker* first evaluates the constraints in Sect. 2.1. This reduces SA search space, as the number of potential drivers to be assigned to a job depends on vehicle type, driver training, availability, etc. Reduction in search space is important to achieve an acceptable solution in NP-hard problems. Then the (j_{id}, d_{id}) pairs, which are identified as eligible for further processing, run through SA algorithm to find an optimal solution while maximizing the job coverage and minimizing the overall cost. Once, a driver is assigned to a job, his/her availability for other jobs will change. For example, once a job is assigned, a driver is not eligible for another job with a pickup time that is earlier than the delivery time of current job. Moreover, the cost of reaching the next pickup location can either reduce or increase. Therefore, respective driver's availability and cost for other unassigned jobs are recalculated using rule checker, before considering for further job assignment by SA algorithm.

All distances between pickup and delivery as well as estimated delivery times are taken from Google Distance API [10] to achieve more reliable and accurate estimates. Fee for a job f_j is calculated based on this distance estimate. We also define several parameters to make the solution more accurate. For example, a driver may need to use public transportation to reach the point of collection or to return to home or next job after a delivery. That journey may take more time than driving time to/from delivery/pickup location, hence needs to be accurately captured while assigning subsequent jobs, as well as to estimate the departure/return time to/from a job. While the cost of public transportation is usually lower than taking a taxi or rental car, it still needs to be considered while calculating the cost of a job c_d^j.

The main parameters of the SA algorithm are an initial temperature, a rule for accepting a damaging move (i.e., higher cost solution), the rate at which the temperature decrements, maximum number of neighboring solutions that can be generated at each temperature, and a stop criterion [11]. In our solution, we prioritize to cover all possible jobs and then optimize the overall cost.

4 Performance Analysis

4.1 Workload Creation

We used two different job datasets each with 80 jobs, against a set of 60 drivers. These datasets were created based on properties extracted from a dataset of a real VD company. This includes the distribution of job locations, pickup and delivery times (some jobs span across two days), driver skill distribution, and other constraints. Table 2 shows a summary of driver availability across a week. Figure 1 illustrates the job and driver locations. The size of circles represents the number of jobs and drivers of a city. As most of the jobs are from urban areas, recruited driver population also reflect somewhat matching behavior. Jobs in the second dataset are more diverse than the first dataset; hence, excepted to take more time to complete. For each day's workload, we plan the schedule in the previous evening.

Table 2. Driver availability by day.

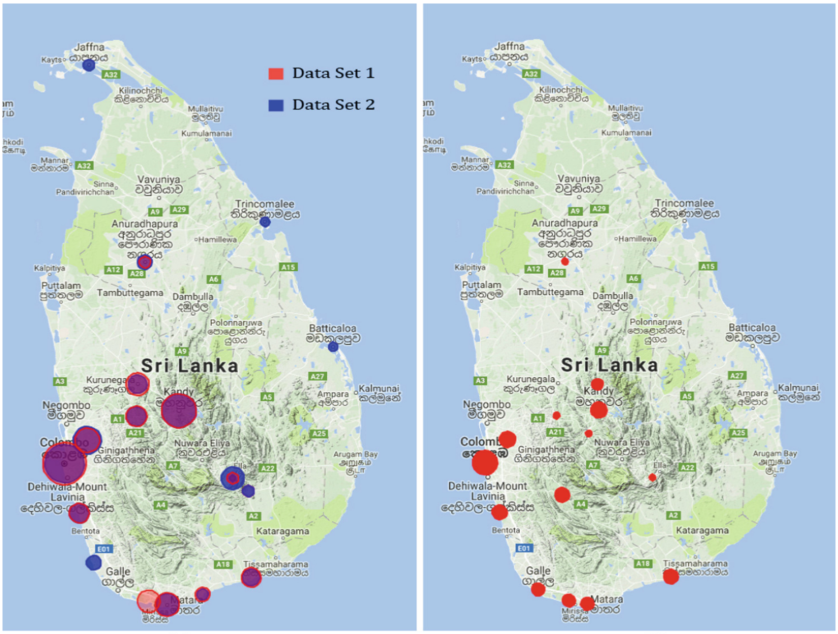
Day	No of drivers	Average available time (H)
Sunday	13	7.23
Monday	58	9.56
Tuesday	48	8.48
Wednesday	49	7.51
Thursday	50	8.34
Friday	39	8.00
Saturday	33	9.59

Several parameters need to be considered as the problem runs under a dynamic environment. Instead of assigning a currency value for job fee and costs, we calculate it in terms of the distance to be driven and distance on public transportation. We assume that the cost of taking a unit distance on public transportation is 0.7 of the cost of driving a vehicle. Moreover, the time required to take public transportation is calculated by multiplying the estimated time from Google Distance API by a factor of 1.2 (for the region considered in the analysis, Google Distance API does not give the time for public transportation). Moreover, a job will be assigned to a preferred driver, if the cost of assigning a preferred driver λ is within 50 form the lowest cost driver.

The tuning process of the SA algorithm is a delicate issue. It depends on the cooling strategy such as linear or exponential, cooling rate, and the energy of the system [12]. Therefore, different combinations were tested on datasets resulting with an initial temperature of 10^4, cooling rate of 0.003, and terminating condition of temperature >1. We ran the simulation five times with same job and driver dataset, and same configurations while varying the random seed.

4.2 Results

In the VD business, pickup and delivery times on most jobs are not strict, and can be advanced or delayed by a couple of hours, as far as the client is kept informed.



(a) (b)

Fig. 1. (a) Job distribution and (b) Driver distribution.

This enables flexibility in optimally assigning jobs while minimizing costs, as well as dealing with unexpected delays due to breakdowns, traffic, and weather. Table 3 shows a summary of the resulting solution under varying levels of flexibility in job pickup and delivery times. Number of jobs covered (aka., job coverage) depends on constraints among jobs and drivers, driver availability, job collection/delivery time, and time gap allowed to complete jobs. The job coverage is highest (92.5% and 87.5% for dataset 1 and 2, respectively) when the time window is ±1 h on Monday. While an increased time window provides more flexibility, it also reduces job coverage and increases driver idle time. Cost is a measurement of the distance traveled by the driver to deliver the vehicle. This is the reason that the cost reduces as the number of assigned jobs reduces with increasing time window. SA execution time increases as the time window increases, because of the increased search space in SA. Table 4 shows the summary of

Table 3. Results against different time windows on Monday for dataset 1 and 2.

Time window (H)	0		±1		±2		±3		±4	
Dataset	1	2	1	2	1	2	1	2	1	2
Job coverage	4	7	74	70	68	65	62	59	54	52
SA execution time (s)	392	406	205	244	234	254	243	269	254	297
Cost (×100)	6	9	170	166	157	154	146	149	138	133
Profit (×100)	11	22	205	206	171	185	144	146	126	144

Table 4. Results against different time windows across a week.

Tuesday										
Time window (H)	0		±1		±2		±3		±4	
Dataset	1	2	1	2	1	2	1	2	1	2
Job coverage	3	6	62	58	55	53	51	49	47	42
SA execution time (s)	372	381	227	290	238	293	261	304	268	316
Cost (×100)	5	11	136	132	130	135	119	121	109	92
Wednesday										
Time window (H)	0		±1		±2		±3		±4	
Dataset	1	2	1	2	1	2	1	2	1	2
Job coverage	3	5	58	55	48	47	42	40	38	37
SA execution time (s)	362	373	223	285	240	297	274	344	289	389
Cost (×100)	5	11	119	133	99	135	97	121	78	92
Thursday										
Time window (H)	0		±1		±2		±3		±4	
Dataset	1	2	1	2	1	2	1	2	1	2
Job coverage	3	6	67	62	59	56	53	50	49	42
SA execution time (s)	371	373	217	264	225	299	242	324	262	329
Cost (×100)	5	12	154	140	132	132	118	132	120	95
Friday										
Time window (H)	0		±1		±2		±3		±4	
Dataset	1	2	1	2	1	2	1	2	1	2
Job coverage	2	5	51	48	44	42	40	38	34	32
SA execution time (s)	348	359	224	265	236	299	252	317	271	326
Cost (×100)	3	9	109	112	95	86	85	86	64	73
Saturday										
Time window (H)	0		±1		±2		±3		±4	
Dataset	1	2	1	2	1	2	1	2	1	2
Job coverage	3	6	61	55	52	49	45	42	40	39
SA execution time (s)	350	339	211	258	219	269	222	289	228	307
Cost (×100)	5	12	129	122	121	123	105	100	106	96
Sunday										
Time window (H)	0		±1		±2		±3		±4	
Dataset	1	2	1	2	1	2	1	2	1	2
Job coverage	2	4	18	17	16	14	14	13	13	13
SA execution time (s)	287	323	255	295	252	294	245	306	254	302
Cost (×100)	3	6	32	32	33	28	21	20	16	19

results over a week. It shows that the number of assigned jobs is proportional to the available number of drivers and their available times (see Table 2). When we increase the cooling rate (say to 0.03), we still got an acceptable solution with less execution time. However, the resulting solution was less stable across different simulations, compared to when the cooling rate is 0.003.

Table 5. Job coverage with varying job delays with ±1 h time window for Monday.

Delay (H)	0		5		10		15	
Dataset	1	2	1	2	1	2	1	2
(No of jobs, Distance) affected								
(5, 392)	74	70	71	67	71	67	71	67
(10, 618)	74	70	62	57	62	57	62	57

More than one job can be assigned to a driver depending on his/her availability. However, if a job gets delayed due to reasons such as an accident, breakdown, traffic, and customer delay, all subsequent jobs of that driver gets affected. If the driver is allowed to take the vehicle home, as it needs to be delivered on the following day, such delays could impact jobs in the following day as well. Moreover, a driver may suddenly become unavailable due to sickness. In those scenarios, it is hard to assign another driver because they may be already assigned to other jobs. Thus, this could result in a chain reaction. Table 5 shows how different job delays and number of affected jobs impact the overall job coverage. When increasing the delay, only few jobs in the overall solution get affected as jobs are not too tightly packed.

Usually drivers are paid only for the number of kilometers the vehicle is driven, and subsistence expenses are reimbursed. Hence, a driver's income is proportional to the total distance of assigned jobs. Figure 2 shows that the weekly average of a driver's income is proportional to his/her availability. Gini coefficient [13] of *income*:*availability* ratios of all drivers is 0.194 and 0.218 for the two datasets. Thus, further confirms that driver income is balanced based on their willingness to contribute.

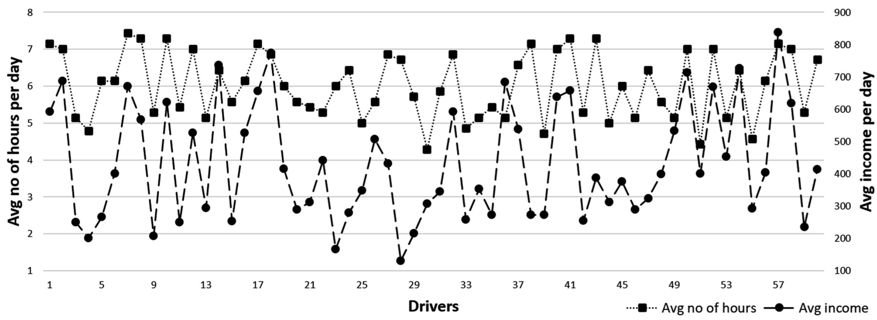


Fig. 2. Weekly average of driver availability and income with ±1 H time window for dataset 1.

5 Summary

We proposed a rule and Simulated Annealing based technique for a driver scheduling in the vehicle delivery industry. We considered an environment where jobs are scheduled in the previous evening based on a set of job and driver constrains. Simulation results using a workload trace derived from a real vehicle delivery company show that the proposed solution assigns drivers to jobs while maximizing job coverage and minimizing cost and computational time, and fairly distributes the income according to the availability. Moreover, the solution can tolerate unexpected delays in the process without a considerable impact on the majority of the already scheduled jobs. We plan to extend this work to further improve job coverage, capture last minute delivery requests arriving within the day, and to better tolerate unexpected events such as accidents, breakdowns, and traffic.

Acknowledgments. This research is supported in part by the Senate Research Grant of the University of Moratuwa under award number SRC/LT/2016/14.

References

1. Bertossi, A.A., Carraresi, P., Gallo, G.: On some matching problems arising in vehicle scheduling models. Networks **17**, 271–281 (1987)
2. Wren, A., Fores, S., Kwan, A.S.K., Kwan, R.S.K., Parker, M., Proll, L.G.: A flexible system for scheduling drivers. J. Sched. **6**(5), 437–455 (2003)
3. Fores, S., Proll, L.G., Wren, A.: An improved ILP system for driver scheduling. In: Wilson, N.H.M. (ed.) Computer-Aided Transit Scheduling, pp. 43–62. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-642-85970-0_3
4. Laurent, B., Hao, J.K.: Simultaneous vehicle and driver scheduling: a case study in a Limousine Rental Company (2007)
5. Hong, L., Ying, W., Shi, L., Sujian, L.: A column generation based hyper-heuristic to the bus driver scheduling problem. In: Discrete Dynamics in Nature and Society, pp. 1–10 (2015)
6. Davis, D.L., Gillenwater, E.L., Johnson, J.D.: An artificial neural systems framework for delivery truck scheduling. In: 23rd Annual Hawaii International Conference on System Sciences, vol. 3 (1990)
7. Maghrebi, M., Sammut, C., Waller, S.T.: Feasibility study of automatically performing the concrete delivery dispatching through machine learning techniques. Eng. Constr. Architectural Manag. **22**(5), 573–590 (2015)
8. Mayssa, K., Souhail, D., Diala, D., Abderrahman, E.M.: Truck driver scheduling problem: literature review. IFAC-PapersOnLine **49–12**, 1950–1955 (2016)
9. Yang, X., Sylvain, G., Florian, M.: Generalized simulated annealing, computational optimization in engineering - paradigms and applications. InTech (2017). https://doi.org/10.5772/66071
10. Google distance matrix API. https://developers.google.com/maps/documentation/distance-matrix. Accessed 5 Oct 2017
11. Anily, S., Federgruen, A.: Simulated annealing methods with general acceptance probabilities. J. Appl. Probab. **24**(3), 657–667 (1987)
12. Nourani, Y., Andresen, B.: A comparison of simulated annealing cooling strategies. J. Phys. A: Math. Gen. **31**, 8373–8385 (1998)
13. Income inequality and dualism. http://www.unc.edu/~nielsen/special/s2/s2.htm. Accessed 5 Oct 2017