2018 INFORMS International Conference
Taipei, Taiwan, June 17–20, 2018

https://doi.org/10.1287/infp.2018.XXXX
pp. 000–000

Mobile Mechanic Scheduling with Time Windows and Time-Dependent Travel Speeds

Xue Han
YourMechanic, Inc., lulu@yourmechanic.com

Rajat Agarwal
YourMechanic, Inc., rajat@yourmechanic.com

Abstract At YourMechanic, we work with mobile mechanics to provide car repair services at car-owner specified locations. In this paper, we formulate a novel mixed integer program that maximizes mechanic utilization and prescribes the optimal route for each mechanic, subject to time-dependent travel speeds and car-owner-specified time windows. This problem is challenging because mechanics are not identical in their service areas, skill sets and availability that could be intermittent. We perform sensitivity analyses of time-window widths using real-life data in our top 10 markets and provide business insights. We also propose a modification on our model if minimizing travel time is to be pursued.

Keywords Scheduling, Mixed integer programming, Assignment problem

1. Introduction

At YourMechanic, we work with a nationwide network of partner mobile mechanics to provide car repair services at car owners' specified locations. In our current customer-centric scheduling system, we allow car owners to see all the available time slots so they can book a job at the earliest available one or the one that works the best for their schedule. This, however, is sub-optimal for the mechanics because they may have consequent jobs at opposite ends of their service areas and end up spending long time in transit. On the other hand, if car owners are allowed to select a time window instead of a specific time, mechanics may be afforded the ability to move the starting times within the window so as to schedule as many minutes of jobs as possible. We define daily mechanic utilization as the ratio of total minutes of scheduled jobs over the total available minutes of mechanics. In this paper, we aim on maximizing mechanic utilization while prescribing the optimal route for each mechanic, subject to time-dependent travel speeds and car-owner-specified time windows.

Each mechanic has a base zip code to start his/her travel of assigned jobs, and serves a set of zip codes within the same designated market area (DMA) region of his base zip code. Mechanics in the same DMA region could serve different sets of zip codes. We say a mechanic is *qualified* to do a job if s/he serves the zip code of the job and meets the skill requirements to perform the job. On each day, mechanics availability of the next day is given as a priori in terms of 15-minute time slots, which could be intermittent during the business hours. Our problem is challenging due to imposed real-life constraints: (1) mechanics are not identical, differing from each other in service areas, skill sets and available time slots; (2) travel speeds between jobs are different in rush hours and regular hours; and (3) mechanics must arrive at each customer's location within the hard time window.

In the literature of time dependent vehicle routing problem (TDVRP) with time windows (e.g., [1], [5], [6], [7]), formulations were based on directed graphs and several near-optimal heristics were proposed. [8] proposed a network model for assignment problem with dynamic demand. However, we cannot use a network formulation because arcs between jobs are not shared by all mechanics, that is, not every mechanic is qualified to perform a customer-requested job. [2] proposed an integer programming model to assign trucks to docks given the time window within which the trucks are present at cross dock, and derived several classes of valid inequalities in a branch-and-cut framework; but they did not consider time-dependent travel speeds or intermittent truck availability. In this paper, we propose a novel mixed-integer-program (MIP) to maximize mechanic utilization and tackle all the three requirements as discussed in the last paragraph.

This paper is structured as follows. In Section 2, we propose the MIP model for mechanic scheduling with time windows and time-dependent travel speeds on a single day in a single DMA region. In Section 3, we conduct case studies to show improvement on mechanic utilization compared to the current customer-centric scheme, and provide sensitivity analyses of time-window widths using the real-life data of our top 10 DMA regions. Section 4 concludes our study and discusses further researches, providing a modification on our model for an objective of minimizing travel time.

2. MIP Model of Mechanic Scheduling

Our serving areas are disjoint by DMA regions (i.e., each DMA region has its own set of mechanics). Therefore, in this section, we model a modular MIP on a single day in a single DMA region, which maximizes the total minutes of jobs that are assigned to mechanics and prescribes the optimal route of each mechanic with time windows and time-dependent travel speeds. Note that mechanic availability is given as a priori, the objective is equivalent to maximizing mechanic utilization. Before we introduce the model, there are certain assumptions that we hold: (1) travel time from base zip code to the first job and from the last job to the base zip code do not consume mechanic available time slots; and (2) travel times between jobs adhere triangle inequality rule.

We first define three index sets given a particular DMA region on a particular day. Let $J = \{1, 2, ..., n'\}$ be the set of jobs to be scheduled, $M = \{1, 2, ..., m'\}$ be the set of available mechanics and $S = \{s_1, ..., s_2\}$ be the set of time slots within the business hours during the day, starting from slot s_1 and ends at slot s_2.

Input parameters to the model are denoted in capitalized letters:

- $B_{ms} = 1$ if mechanic m is available in slot s, 0 otherwise, for $m \in M, s \in S$,
- L_j = duration of job j in minutes, for $j \in J$,
- $H_{jm} = 1$ if mechanic m is qualified to do job j, 0 otherwise, for $j \in J$, $m \in M$,
- $D_{j_1 j_2}$ = travel distance between jobs j_1 and j_2 in miles, for $j_1 \in J$, $j_2 \in J$,
- $[T_1, T_2]$ ($[T_3, T_4]$)= duration of morning (evening) rush hours in terms of time slots,
- F (G) = travel speed in terms of minutes/mile for rush hours (regular hours),
- $[C_{j0}, C_{j1}]$ = time window that customer selected to start job j, C_{j0} (C_{j1}) represents the first (last) time slots in this time window.

We define two sets of decision variables to determine job assignments:

- $y_{jms} = 1$ if job j is assigned to mechanic m to start from time slot s, 0 otherwise, for $j \in J$, $m \in M$, $s \in S$.
- $z_{jms} = 1$ if job j uses time slot s of mechanic m, 0 otherwise, for $j \in J$, $m \in M$, $s \in S$

To depict travel time between jobs, we define several auxiliary decision variables that prescribe the order of jobs, including the following sets of binary variables:

- $v_{j_1 j_2 m} = 1$ if jobs j_1 and j_2 are assigned to mechanic m and j_1 starts before j_2, 0 otherwise, for $j_1 \in J$, $j_2 \in J$, $m \in M$

- $\alpha_j = 1$ if jobs j's ending time slot $\geq T_1$, 0 otherwise, for $j \in J$
- $\beta_j = 1$ if jobs j's ending time slot $\leq T_2$, 0 otherwise, for $j \in J$
- $\gamma_j = 1$ if jobs j's ending time slot $\geq T_3$, 0 otherwise, for $j \in J$
- $\theta_j = 1$ if jobs j's ending time slot $\leq T_4$, 0 otherwise, for $j \in J$
- $r_j = 1$ if jobs j's ending time slot is in rush hour, 0 otherwise, for $j \in J$

and continuous variables:

- e_j = ending slot of job j, for $j \in J$, $c_j \geq 0$, $c_j \leq s_2 - L_j + 1$
- $\tau_{j_1 j_2}$ = travel minutes between jobs j_1 and j_2, for $j_1 \in J$, $j_2 \in J$, $\tau_{j_1 j_2} \geq 0$

Now, we propose the modular MIP model, denoted as problem (P), that maximizes the total assigned job minutes in the given DMA region on a given day.

$$\max \quad t = \sum_{j \in M} \sum_{m \in M} \sum_{s \in S} L_j y_{jms}$$

$$\text{(P)} \quad \sum_{m \in M} \sum_{s \in S} y_{jms} \leq 1 \qquad \forall j \in J \tag{1}$$

$$z_{jms'} \geq y_{jms} \qquad \forall j \in J, m \in M, s \in S, \; s \leq s' \leq \min\{s + L_j - 1, s_2\} \tag{2}$$

$$\sum_{j \in J} z_{jms} \leq B_{ms} \qquad \forall m \in M, s \in S \tag{3}$$

$$\sum_{s \in S} z_{jms} \leq L_j H_{jm} \qquad \forall j \in J, m \in M \tag{4}$$

$$\sum_{m \in M} \sum_{s \in S} z_{jms} \leq L_j \qquad \forall j \in J \tag{5}$$

$$e_j = \sum_{s \in S} \sum_{m \in M} s y_{jms} + L_j - 1 \qquad \forall j \in J \tag{6}$$

$$e_j \leq s \qquad \forall j \in J \tag{7}$$

$$e_j - L_j + 1 \leq C_{j1} \qquad \forall j \in J \tag{8}$$

$$e_j - L_j + 1 \geq C_{j0}\left(\sum_{m \in M} \sum_{s \in S} y_{jms}\right) \qquad \forall j \in J \tag{9}$$

$$e_j - T_1 + 1 \leq U\alpha_j \qquad \forall j \in J \tag{10}$$

$$e_j - T_1 \geq -U(1 - \alpha_j) \qquad \forall j \in J \tag{11}$$

$$T_2 - e_j + 1 \leq U\beta_j \qquad \forall j \in J \tag{12}$$

$$T_2 - e_j \geq -U(1 - \beta_j) \qquad \forall j \in J \tag{13}$$

$$e_j - T_3 + 1 \leq U\gamma_j \qquad \forall j \in J \tag{14}$$

$$e_j - T_3 \geq -U(1 - \gamma_j) \qquad \forall j \in J \tag{15}$$

$$T_4 - e_j + 1 \leq U\theta_j \qquad \forall j \in J \tag{16}$$

$$T_4 - e_j \geq -U(1 - \theta_j) \qquad \forall j \in J \tag{17}$$

$$r_j \geq \alpha_j + \beta_j - 1 \qquad \forall j \in J \tag{18}$$

$$r_j \geq \gamma_j + \theta_j - 1 \qquad \forall j \in J \tag{19}$$

$$\tau_{j_1 j_2} = G D_{j_1 j_2} + (F - G) D_{j_1 j_2} r_{j_1} \qquad \forall j_1 \in J, j_2 \in J \tag{20}$$

$$v_{j_1 j_2 m} \leq \sum_{s \in S} y_{j_1 ms} \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{21}$$

$$v_{j_1 j_2 m} \leq \sum_{s \in S} y_{j_2 ms} \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{22}$$

$$v_{j_1j_2m} + v_{j_2j_1m} \leq 1 \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{23}$$

$$v_{j_1j_2m} + v_{j_2j_1m} \geq \sum_{s\in S} y_{j_1ms} + \sum_{s\in S} y_{j_2ms} - 1 \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{24}$$

$$\sum_{s\in S} s y_{j_2ms} - \sum_{s\in S} s y_{j_1ms} \geq -2U(1 - v_{j_1j_2m}) + \frac{(L_{j_1} + \tau_{j_1j_2})}{15} \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{25}$$

$$y_{jms} \in \{0,1\}, \quad z_{jms} \in \{0,1\} \qquad \forall j \in J, m \in M, s \in S \tag{26}$$

$$v_{j_1j_2m} \in \{0,1\} \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{27}$$

$$\alpha_j \in \{0,1\}, \beta_j \in \{0,1\}, \gamma_j \in \{0,1\}, \theta_j \in \{0,1\}, r_j \in \{0,1\} \qquad \forall j \in J \tag{28}$$

$$e_j \geq 0 \qquad \forall j \in J \tag{29}$$

$$\tau_{j_1j_2} \geq 0 \qquad \forall j_1 \in J, j_2 \in J \tag{30}$$

Constraints (1)-(5) are the assignment constraints. Constraints (1) make sure that each job is assigned to at most one mechanic. Constraints (2) state that if job j is assigned to mechanic m to start at slot s, this job must occupy a consecutive number of L_j slots starting from slot s on mechanic m's calendar. Constraints (3) state that for each available slot s of mechanic m, at most one job could use this slot. Constraints (4) ensure that each job can only be assigned to a qualified mechanic who is able perform the job. Constraints (5) guarantee that the total number of slots a job occupies do not exceed the duration of the job.

Constraints (6)-(9) impose constraints on the starting and ending time slots of each job. Noting that $\sum_{s\in S}\sum_{m\in M} s y_{jms}$ is the starting slot of job j, constraints (6) define the ending slot of job j (i.e., the last time slot that job j occupies). Constraints (7) ensure that the ending time slot of each job does not exceed the business hours, preventing a situation in which a job starts within business hours but there are not enough time slots left on mechanic calendar to complete the job.

Constraints (10)-(20) deal with travel time definitions in rush hours and regular hours. Constraints (10) force α_j to be 1 if job j finishes after the morning rush hours *start* (i.e., $e_j \geq T_1$). On the other hand, if job j finishes before the morning rush hours *start* (i.e., $e_j \leq T_1 - 1$), the left-hand-side of constraint (11) is negative, and α_j must be 0 to make the right-hand-side a more negative value. Analogously, constraints (12) and (13) ensure that $\beta_j = 1$ if job j finishes before the morning rush hours *end*, and $\beta_j = 0$ otherwise. Constraints (18) indicate that job j finishes in the morning rush hours if the ending slot of job j is in range $[T_1, T_2]$. Similarly, Constraints (14)-(17) and (19) indicate whether the ending time slot of a job is in the evening rush hours. Constraints (20) indicate that if job j_1 finishes in regular hours (i.e., $r_{j_1} = 0$), the travel time from j_1 to another job j_2 is $GD_{j_1j_2}$. Otherwise, if $r_j = 1$, the travel time is defined as $FD_{j_1j_2}$ using rush-hour speed.

Constraints (21)-(25) relate the order of jobs and ensure that there is enough time allocated between jobs. Constraints (21) and (22) indicate that if either job j_1 or job j_2 is not assigned to mechanic m, the job order indicator $v_{j_1j_2m}$ must be 0. Constraints (23) state that there is at most one order of jobs between j_1 and j_2. This set of constraints also force $v_{jjm} = 0$ for $\forall j \in J$. Together with (23), constraints (24) are tight only when both jobs j_1 and j_2 are assigned to mechanic m, indicating that one of $v_{j_1j_2m}$ and $v_{j_2j_1m}$ must be 1, that is, either j_1 starts before j_2, or j_2 starts before j_1. Constraints (25) is non-trivial only when $v_{j_1j_2m} = 1$, ensuring that before job j_2 starts, there is enough time to complete job j_1 and travel from job j_1 to job j_2. Note that we assume triangle inequalities of travel times, so as long as mechanic m performs job j_1 before j_2, constraints (25) impose enough travel time between these two jobs. For depicting *adjacent* pair of jobs (j_1, j_2) assigned to a mechanic, please see the discussion in section 4. Decision variables type and domain are indicated in constraints (26) - (30).

3. Case study and sensitivity analyses

This section performs numerical studies to show how much utilization could be improved using the proposed model in Section 2 compared to our current customer-centric scheme and conducts sensitivity analyses on time window widths. We use real-life data in the week of 2018-01-08 in our top 10 DMA regions in terms of booked jobs. Repair jobs typically have a one-day wait time to get the parts ready. We say a job is *not waiting on parts* if the first available, qualified mechanic makes the job wait longer than a day. Let $\mathbb{Q}_i$ be the set of jobs that are considered to be scheduled for day i, in which jobs will be fed into problem (P) iteratively. We define the following sets of jobs to construct set $\mathbb{Q}_i$:

- set $\mathbb{A}_i$ comprises jobs that have been completed on day i;
- set $\mathbb{B}_i$ comprises not-waiting-on-parts jobs that were booked on day $i-1$ for a future appointment date in range $[i+1, i+6]$. This set of jobs could have started on day i but did not in our current customer-centric scheme;
- set $\mathbb{D}_{i-1}$ comprises the jobs that were included in $\mathbb{Q}_{i-1}$, but could not be assigned to any mechanic. This set of jobs are rolled over to day i.

Set $\mathbb{D}_{i-1}$ could overlap with sets $\mathbb{A}_i$ and/or $\mathbb{B}_i$. Also, any jobs from $\mathbb{A}_i \cup \mathbb{B}_i \cup \mathbb{D}_{i-1}$ could have been included in $\mathbb{Q}_1$, $\mathbb{Q}_2$, ..., $\mathbb{Q}_{i-1}$ and prescribed to a mechanic by solving (P) on a day prior to day i. Let $\mathbb{J}$ be the set of jobs that have been scheduled by far. Then, we have

$$\mathbb{Q}_i := (\mathbb{A}_i \cup \mathbb{B}_i \cup \mathbb{D}_{i-1}) - \mathbb{J} = \mathbb{A}_i \cup \mathbb{B}_i \cup (\mathbb{D}_{i-1} - (\mathbb{A}_i \cup \mathbb{B}_i)) - (\mathbb{J} \cap (\mathbb{A}_i \cup \mathbb{B}_i \cup \mathbb{D}_{i-1})) \tag{31}$$

For each job in $\mathbb{Q}_i$, we know its original appointment date, original start time and duration. Qualified mechanics for each job are computed using a matcher program in our system. Travel distance between two jobs are computed by the arc distance on earth using their longitudes and latitudes. We also obtain mechanic calendars on each day to define availability per mechanic per time slot.

Procedure 1 Simulation of mechanic scheduling in a given DMA region in a given week

1: $\mathbb{D}_0 = \emptyset$, $\mathbb{J} = \emptyset$.
2: **for** $i \in [1, 7]$
3:     obtain $\mathbb{A}_i$ and $\mathbb{B}_i$
4:     obtain $\mathbb{Q}_i$ as defined in (31), sort $\mathbb{Q}_i$ in ascending order of booked time.
5:     $\mathbb{D}_i = \emptyset$
6:     **for** $j \in \mathbb{Q}_i$
7:         obtain time window $[C_{j0}, C_{j1}]$ of job j
8:         add job j to problem (P) and solve problem (P)
9:         **if** $\sum_{m \in M} \sum_{s \in S} y_{jms} = 1$
10:             $\mathbb{J} \leftarrow \mathbb{J} \cup \{j\}$
11:             change constraint (1) to equality for job j
12:         **else**
13:             $\mathbb{D}_i \leftarrow \mathbb{D}_i \cup \{j\}$

Procedure 1 describes the procedure of a 7-day simulation. We sort the jobs in $\mathbb{Q}_i$ by ascending order of the time they were booked, so as to preserve the same order of booking stream as our current system. For each job in $\mathbb{Q}_i$, we obtain the time window into which the original appointment start time falls, so as to retain the time preference of customers'. On day i, we add jobs in $\mathbb{Q}_i$ to (P) one by one and solve (P) iteratively. Line 11 makes sure that if an earlier-booked job was assigned to some mechanic on day i, it will always to able to be assigned to a mechanic (possibly a different mechanic) on day i.

TABLE 1. Results for DMA region 1 in a 7-day simulation

day (i)	mech. avail. (mins)	$\mathbb{A}_i$ (mins)	$\mathbb{B}_i$ (mins)	roll-over (mins)	prev. schd. (mins)	$\mathbb{Q}_i$ (mins)	t (mins)	orig. util (%)	opt. util (%)	orig. wait (hrs)	opt. wait (hrs)
1	4380	873	1071	0	0	1944	1257	20	29	102	92
2	4560	1254	648	495	180	2217	1791	28	39	110	100
3	4800	1314	1968	426	303	3405	2484	27	52	93	43
4	4185	1272	1209	738	282	2937	1416	30	34	97	72
5	3825	1368	975	1083	405	3021	1632	36	43	94	78
6	4170	969	717	1116	180	2622	1602	23	38	79	53
7	3780	606	639	951	180	2016	1095	16	29	93	63

Due to business confidentiality, we do not reveal the name of our DMA regions, which cover several major cities in California, Texas, Florida, Georgia, etc. in the United States. Now let's take DMA region 1 as an example. Our business hours are from 7:00AM to 9:00PM ($s_1 = 28$, $s_2 = 83$). Morning rush hours are from 7:00AM to 10:00AM ($T_1 = 24$, $T_2 = 39$) and evening rush hours are from 3:00PM to 7:00PM ($T_3 = 60$, $T_4 = 75$). Time-dependent travel speeds are 4 minutes/mile (F) for rush hours and 2.5 minutes/mile (G) for regular hours.

We model problem (P) using Pyomo 5.2, a Python-based, open-source optimization modeling language [4], [3]. Note that constraints (1) - (19), (26), (28) and (29) could be partitioned by each job in J, we use Pyomo's block component to structure the model in which each block includes the decision variables and constraints for a job, and use constraints (20) - (25) and (27) as the linking constraints between blocks. We use IBM ILOG CPLEX12.7® branch-and-bound solver with default settings to solve MIPs. We perform all tests on an Intel Core i7 CPU @ 2.5GHz, a 64-bit operating system with 16GB RAM.

Table 1 presents the results of the 7-day simulation for DMA region 1 using a 2-hr time window. Column 2 reports the total available minutes of mechanics on each day. Columns 3-7 report the composition of input jobs in terms of minutes, in which "roll-over" represents the minutes of jobs in set $\mathbb{D}_{i-1} - (\mathbb{A}_i \cup \mathbb{B}_i)$; and "prev. schd." represents the minutes of jobs in set $\mathbb{J} \cap (\mathbb{A}_i \cup \mathbb{B}_i \cup \mathbb{D}_{i-1})$ that should be deducted from the input minutes to day i. Column 8 reports the total minutes of jobs that are scheduled on day i in procedure 1. Columns 9 and 10 report the utilization in our current customer-centric scheme and the one prescribed by procedure 1, respectively. Columns 11 and 12 compare the wait time per job before and after we employ procedure 1, which is defined as the hours from booking the job to the start time of the job. We can tell that the more minutes of jobs are fed in $\mathbb{Q}_i$, the higher utilization could be achieved. Utilization improvement is more significant when the original utilization is lower. By scheduling more jobs on earlier days, the average wait time per job is shortened by 23.8 hours. The size of problem (P) is $\mathcal{O}(|J||M||S| + |J|^2|M|)$. For instance, the last iteration of (P) we solve on day 1 includes 22 jobs, involving 44836 decision variables (44330 binaries) and 57544 constraints.

Table 2 presents the settings before we implement procedure 1 for each DMA region. Column 2 presents the total mechanic available minutes. Columns 3 (4) are total minutes of jobs in set $\cup_{i=1}^{7}\mathbb{A}_i$ ($\cup_{i=1}^{7}\mathbb{B}_i$ - $\cup_{i=1}^{7}\mathbb{A}_i$). Column 5 sums up columns 3 and 4, representing the total minutes of jobs considered in procedure 1. Column 6 reports the number of qualified mechanics per job, indicating the competitiveness between mechanics. Column 7 computes the average daily mechanic utilization in customer-centric scheme. Table 3 reports the utilization improvement in each DMA region when we use 0.5-hr, 1-hr, 2-hr and 4-hr time windows in procedure 1, as well as the total run time for all 4 time-window widths.

Every DMA region sees an improvement on average daily utilization, ranging from 32% to 56% more utilization than the average daily utilization in table 2. Note that the utilization does not increase monotonically with broader time-window widths (see DMA regions 2, 6 and 8). This is because that we respect the order of booking. An earlier-booked, long-duration

TABLE 2. Basic information about the top 10 DMA regions

DMA Region (i)	mech. avail. (mins)	orig. jobs (mins)	extra jobs (mins)	total input jobs (mins)	# of qualified mechs /job (mins)	avg. daily util. (%)
1	29700	7656	4542	12198	3.81	26
2	34200	9000	4062	13062	3.15	26
3	41850	10044	4659	14703	4.77	24
4	39885	8514	3075	11589	8.38	21
5	9675	2373	1833	4206	3.00	25
6	20010	4356	1827	6183	3.67	22
7	26175	3741	1653	5394	4.25	14
8	18675	4983	2808	7791	2.83	27
9	16665	5313	4098	9411	3.35	32
10	15810	5019	2706	7725	7.07	32

TABLE 3. Time windows sensitivity on utilization

DMA Region (i)	avg. daily util.(0.5hr) (%) (mins)	avg. daily util.(1hr) (%) (mins)	avg. daily util.(2hr) (%) (mins)	avg. daily util.(4hr) (%) (mins)	avg. daily util. improved (%)	total run time (secs)
1	36	36	38	39	45	191.16
2	34	35	34	35	32	154.32
3	30	32	33	34	34	234.10
4	28	28	28	29	32	128.08
5	35	36	39	43	56	8.49
6	31	29	30	30	38	18.92
7	17	18	18	18	22	16.13
8	35	35	36	35	32	30.19
9	46	48	48	53	53	30.85
10	44	46	47	47	45	18.40

job could have been scheduled in a 4-hr time-window scheme, but not in a 2-hr time-window scheme, because 4-hr time-window provides more starting time slots for this job to choose from. Such a job in the 4-hr time-window scheme could prevent later jobs to be scheduled as it occupies a large number of consecutive time slots, making the overall utilization lower than the one in the 2-hr time-window scheme. Using table 3, we can find the shortest time-window width that provides the largest utilization improvement in particular DMA regions, if it exists, for example, 1-hr time-window for DMA region 2, and 2-hr time-window for DMA region 10. This is the optimal time-window width to balance between mechanic utilization and customer experience. Run time is increasing with mechanic available hours, which affect the number of tight inequalities in constraints (3).

4. Conclusions and future research

This paper formulates a novel MIP model that solves the mechanic scheduling problem with time window and time-dependent travel speeds, aiming on maximizing mechanic utilization. This formulation deals with many realistic business requirements of scheduling mechanics, allowing mechanics to have different service areas, skill sets and availability. Mechanic available time slots could be intermittent, too. Out model shows improvement on utlizations compared to our current customer-centric scheme by at least 32%, and providing business insights on finding the optimal width of time window to display to customers in different DMA regions.

If readers are interested to apply our model to minimize total travel time instead of maximizing utilization, we propose to identify whether a pair of jobs are *adjacently* assigned to a mechanic, because travel time is only counted between adjacent jobs. Let decision variables $u_{j_1j_2m}$ be 1 if job j_2 is the next job after j_1 to be serviced by mechanic m, 0 otherwise, for $j_1 \in J$, $j_2 \in J$, $m \in M$. To depict travel time between adjacent jobs, we replace constraints (20) with the following two sets of constraints:

$$\tau_{j_1j_2} \leq D(\sum_{m\in M} u_{j_1j_2m}) \qquad \forall j_1 \in J, j_2 \in J \tag{32}$$

$$\tau_{j_1j_2} - (GT_{j_1j_2} + (F-G)T_{j_1j_2}r_{j_1}) \geq D(\sum_{m\in M} u_{j_1j_2m} - 1) \quad \forall j_1 \in J, j_2 \in J, \tag{33}$$

where D is an upper bound on travel time.

To define *adjacency* between jobs, $u_{j_1j_2m}$ must be 1 if both jobs j_1 and j_2 are assigned to mechanic m, and there is exactly one more job after j_1 than the ones after j_2; otherwise $u_{j_1j_2m}$ must be 0. Thus, we impose the following three sets of constraints to relate $u_{j_1j_2m}$ with $v_{j_1j_2m}$:

$$u_{j_1j_2m} \leq v_{j_1j_2m} \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{34}$$

$$\sum_{j\in J} v_{j_1jm} - \sum_{j\in J} v_{j_2jm} - 1 \leq n'(1 - u_{j_1j_2m}) \qquad \forall m \in M, j_1 \in J, j_2 \in J \tag{35}$$

$$\sum_{j_1\in J}\sum_{j_2\in J} u_{j_1j_2m} \geq \sum_{j\in J}\sum_{s\in S} y_{jms} - 1 \quad \forall m \in M \tag{36}$$

In the future, our model could be applied to different on-demand, point-to-point services, such as scheduling technicians, drivers and specialists, to deliver a certain set of services. Run time could be improved by adding facet-defining cutting planes related to time windows.

References

[1] Figliozzi MA (2012) The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review* 48(3):616 – 636.

[2] Gelareh S, Monemi RN, Semet F, Goncalves G (2016) A branch-and-cut algorithm for the truck dock assignment problem with operational time constraints. *European Journal of Operational Research* 249(3):1144 – 1152.

[3] Hart WE, Laird CD, Watson JP, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD (2017) *Pyomo–optimization modeling in python*, volume 67 (Springer Science & Business Media), second edition.

[4] Hart WE, Watson JP, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation* 3(3):219–260.

[5] Huang Y, Zhao L, Woensel TV, Gross JP (2017) Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological* 95:169 – 195.

[6] Malandraki C, Daskin MS (1992) Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science* 26(3):185–200.

[7] Schilde M, Doerner K, Hartl R (2014) Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research* 238(1):18 – 30.

[8] Wilhelm W, Han X, Lee C (2013) Computational comparison of two formulations for dynamic supply chain reconfiguration with capacity expansion and contraction. *Computers & Operations Research* 40(10):2340 – 2356, ISSN 0305-0548.