



HAL
open science

Sharing a ride time constraint in a multi-trip dial-a-ride problem. An application to the non-urgent patient transportation problem

Timothée Chane-Haï, Samuel Vercaene Vercaene, Thibaud Monteiro

► To cite this version:

Timothée Chane-Haï, Samuel Vercaene Vercaene, Thibaud Monteiro. Sharing a ride time constraint in a multi-trip dial-a-ride problem. An application to the non-urgent patient transportation problem. MOSIM 2020 - 13th International Conference on Modeling, Optimization and Simulation, Nov 2020, Agadir, Morocco. hal-03124964

HAL Id: hal-03124964

<https://hal.science/hal-03124964v1>

Submitted on 29 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sharing a ride time constraint in a multi-trip dial-a-ride problem. An application to the non-urgent patient transportation problem

T. CHANE-HAÏ, S. VERCRAENE, T. MONTEIRO

Université de Lyon, INSA Lyon, Laboratoire DISP EA 4570, Lyon, France

timothee.chane-hai@insa-lyon.fr, samuel.vercraene@insa-lyon.fr, thibaud.monteiro@insa-lyon.fr

ABSTRACT: *This article presents a method to find close-to-Pareto-optimal solutions for a multi-objectives round-trip dial-a-ride problem (DARP) decomposed into two dependent sub-problems. In the studied problem, each user expresses two requests, a morning request and an evening request, that represent a round-trip itinerary on one day. The two objectives are the cost and excess ride time minimization. Instead of integrating all the requests into one single DARP, the requests are split between two smaller DARPs: one for the collection of morning requests and one for the collection of evening requests. However, the two DARPs cannot be solved independently because they share a constraint: the user maximal daily ride time. The developed heuristic aims to find a global close to optimal solution within reasonable computation time. The method is applied on real data from the NOMAd project which aims to improve the transportation of disable children from their home to medical centers in the city of Lyon.*

KEYWORDS: *Dial-a-ride problem, distributed decision, shared constraint, multi-trip, healthcare.*

1 INTRODUCTION

The non-emergency transportation of patients from their home to medical facilities is an important part of the healthcare system. It concerns disabled patients who are not able to go to medical facilities by themselves or who are not able to reach a sufficient assistance from a family member to do so. Many aspects are considered in this activity. The first aspect is the human aspect. Many of the patients use this service daily, and the trips can take more than an hour. Transportation takes a considerable amount of time in the daily life of the patients. So, it is important to make it as pleasant as reasonably possible. Good quality of service should be provided for the well-being of the patients. The second aspect is financial. In France, patients transportation cost 3.5 billion euros in 2010 and 5 million of persons were using this service (Tellez et al. 2020). The transportation cost is charged on the patients or on the healthcare system. This represents a considerable amount of money that can be reduced. Other aspects can be considered : the organisational aspect, the technical aspect, the environmental aspect, etc. But overall, the well-being and the financial aspect are the two most important. This transportation problem can be modelled as a dial-a-ride problem.

The dial-a-ride problem is itself a variant of the vehicle routing problem (VRP) which is a NP-hard problem, making computation time increase exponentially with the instance size. DARP and its variants have been extensively studied in the literature with cost reduction as the most studied objective (Cordeau and Laporte 2007; Gendreau et al. 2008; Braekers et al. 2016). In medical context, the well-being of the patients is essential. As a consequence, a second

objective affiliated with this notion is often added. In our problem, we decided to use the unaccepted excess ride time as the service quality objective. The excess ride time corresponds to the ride time minus the minimal ride time. The unaccepted excess ride time corresponds to the ride time minus the maximal allowed ride time. All the requests are known in advance, so the problem is static and deterministic (Ho et al. 2018). Time windows constraints at pick-up and delivery points have been considered (Psaraftis 1983; Jaw et al. 1986; Toth and Vigo 1997). A fleet size and mix component has also been considered: the fleet is constituted of an unlimited number of heterogeneous capacitated vehicles (Golden et al. 1984; Ulusoy 1985). Vehicles are able to go to back to their depot more than once a day, adding a multi-trip component to the problem (Azi et al. 2007; Macedo et al. 2011; Hernandez et al. 2014). The multi-trip component has been previously studied in medical environment (Wong and Bell 2006; Lim et al. 2016) but no formulation took into account a maximal daily ride time constraint for the users. In previous studies, requests from a user were independent as the maximal ride time constraint were only associated with a single trip. In that case, it is possible to have an excessively long round-trip duration for a user despite respecting the individual maximal ride time constraint. This situation can potentially be avoided to improve the quality of service.

In this work, we tackle this issue by adding a maximal daily ride time constraint which binds together a set of requests made by a user. The term daily ride time refers to the set of requests and the term ride time refers to a single request. For each user, setting the maximal daily ride time to a value smaller than the sum of the maximal ride times is the key to reduce the inequalities. Each user calls for a

round-trip, which is a pair of pick-up and delivery needs, with a maximal daily ride time constraint. Consequently, it is no more possible to consider the morning request (from home to the medical facility) and the evening request (from the medical facility back home) of a user as independent requests. The goal is to improve the quality of service and reduce inequities among patients. All those considerations make our problem a fleet size and mix round-trip dial-a-ride problem with a round-trip constraint. In order to compute a feasible solution within reasonable computation time, we decomposed the problem into two smaller dependent sub-problems with a shared constraint. The first sub-problem is a DARP on the morning requests, the second sub-problem is a DARP on the evening requests. This decomposition has two main advantages. Firstly, the size of the problem is reduced by twofold. As the problem is NP-hard, it exponentially reduces the computation time. Secondly, it is possible to compute the sub-problems in parallel. Subdividing the problem or using parallel computation has been done in the past. Decomposing the problem into independent sub-problems is a common practice in the school bus routing problem which is a variant of the DARP (Ellegood et al. 2020). The first sub-problems could be a scheduling problem or/and an assignment problem and the last sub-problem is a routing problem. However, the sub-problems were independent as feasible sub-solutions were certain to lead to a feasible global solution. In another hand, cooperative parallel computing on the same problem (Crainic et al. 1997) has also shown great improvement in computation time. But again, the parallel computations towards a feasible solution were independent. In our problem, the sub-problems are dependent. As a consequence, the global solution made by the combination of the morning and evening results may not respect the maximal daily ride time constraint. In that case, we designed a heuristic that gradually modifies the problem's data and recomputes sub-solutions in order to reach a feasible daily solution. The method used to solve this problem is applied on real data from the NOMAd project. The NOMAd project aims to improve the transportation of disabled children from their home to medical facilities in the metropolis of Lyon.

The remainder of this article is structured as follows: section 2 presents the mathematical model while section 3 describe the heuristic used to solve the problem. The experiments are presented in section 4. Finally, the results are given in section 5.

2 PROBLEM SETTINGS AND MATHEMATICAL MODEL

2.1 Problem settings

As the problem is split into two sub-DARPs, the following is the mathematical model of a single sub-DARP.

A sub-DARP is modeled by a graph $G = (V, A)$. The set V of nodes contains the set of starting depots O^+ , the set of arrival depots O^- , the set of pick-up nodes P and the set of

delivery nodes D . Each node $i \in V$ has a service duration s_i and a time windows $[a_i, b_i]$. The service duration s_i is equal to 0 for any node in O^+ and in O^- . The arc between two nodes i and j represents the fastest and shortest path from i to j . Thus, each arc is associated with a duration t_{ij} and a distance Δ_{ij} . In reality, two nodes may have the same geographical position (e.g., starting depot and ending depot)

K is the set of vehicle types. Each vehicle has a starting depot $O_k^+ \in O^+$, an arrival depot $O_k^- \in O^-$, a maximal capacity Q^k , a fixed cost f_k , a duration cost α_k and a distance cost γ_k . The maximal shift length of a driver is T_k .

R is the set of requests. Each request $r \in R$ has a pick-up node $p_r \in P$, a delivery node $d_r \in D$, a maximal ride time T_r^+ , a minimal ride time T_r^- and a quantity q_r of users to be transported. The minimal and maximal ride time settings are described in section 4.1. The load variation at a node is notated as ϕ , having $\phi_{p_r} = q_{p_r}$ at a pickup node and $\phi_{d_r} = -q_{d_r}$ at a delivery node. At the starting depot and ending depot, the load variation is equal to 0 ($\phi_{O_k^+} = \phi_{O_k^-} = 0$) and there is no user inside a vehicle. This notation allows to translate the load variation from a request to a node.

D is the set of daily requests. Each daily request $d \in D$ is composed of the morning $r_m \in R$ and evening request $r_e \in R$ made by a user: $d_m = r_m, d_e = r_e$. A daily request has a maximal daily ride time T_d^+ and a minimal daily ride time T_d^- . The minimal daily ride time is the sum of the morning and evening minimal ride time ($T_d^- = T_{d_m}^- + T_{d_e}^-$). The maximal daily ride time setting is described in section 4.1.

2.2 Mathematical formulation

Variables

- $x_{i,j}^k$: is a binary variable which is equal to 1 if vehicle k uses arc (i,j) and 0 otherwise, $k \in K, (i, j) \in A$
- w_i^k : is a continuous variable representing the arrival time of vehicle k at node i , $k \in K, i \in V$
- l_i^k : is an integer variable representing the number of users in vehicle k after visiting node i , $k \in K, i \in V$
- ww_r : is a continuous variable representing the excess ride time of the request r , $r \in R$

Objectives

- The cost of a solution represents its transportation cost which the sum of all the fixed cost, the duration cost and the distance cost.

$$\begin{aligned} \min Cost = \min & \sum_{k \in K} f^k \sum_{i \in P} x_{o_k^+ i}^k + \\ & + \sum_{k \in K} a^k (w_{o_k^-}^k - w_{o_k^+}^k) \\ & + \sum_{k \in K} \sum_{(i,j) \in A} \gamma^k \Delta_{ij} x_{ij}^k \end{aligned} \quad (1)$$

- The excess ride time of a solution is the sum of the excess ride times of the users.

$$\min ExcessRideTime = \min \sum_{r \in R} w w_r \quad (2)$$

Constraints

The big-M values are defined in section 4.1.1.

- One and only one vehicle ensures pick-up and delivery for a request.

$$\begin{aligned} \sum_{(p_r, j) \in A} x_{p_r j}^k - \sum_{(j, d_r) \in A} x_{j d_r}^k = 0, \\ \forall r \in R, \forall k \in K \end{aligned} \quad (3)$$

$$\sum_{k \in K} \sum_{(j, p_r) \in A} x_{j p_r}^k = 1, \quad \forall r \in R \quad (4)$$

- A vehicle leaves and returns to its depot no more than once.

$$\sum_{i \in V} x_{o_k^+ i}^k \leq 1, \quad \forall k \in K \quad (5)$$

$$\sum_{i \in V} x_{i o_k^-}^k \leq 1, \quad \forall k \in K \quad (6)$$

- No vehicle enters its starting depot or leaves its ending depot.

$$\sum_{i \in V} x_{i o_k^+}^k \leq 0, \quad \forall k \in K \quad (7)$$

$$\sum_{i \in V} x_{o_k^- i}^k \leq 0, \quad \forall k \in K \quad (8)$$

- The flow is conserved.

$$\sum_{(j,i) \in A} x_{ji}^k - \sum_{(i,j) \in A} x_{ij}^k = 0,$$

$$\forall i \in P \cup D, \forall k \in K \quad (9)$$

- Any vehicle which leaves its depot arrives at its arrival depot.

$$\sum_{i \in P} x_{o_k^+ i}^k - \sum_{i \in P} x_{i o_k^-}^k = 0, \quad \forall k \in K \quad (10)$$

- The arrival time at each node is greater than the time at the previous node plus the service time at the previous node and the transportation time between the two nodes.

$$w_j^k \geq w_i^k + t_{ij} + s_i - M_{11}(1 - x_{ij}^k),$$

$$\forall (i, j) \in A, \forall k \in K \quad (11)$$

- The arrival time at any node is within its time window.

$$a_i - M_{12}(1 - \sum_{j \in V} x_{ij}^k) \leq w_i^k \leq b_i,$$

$$\forall i \in V, \forall k \in K \quad (12)$$

- For any request, pick-up happens before delivery.

$$w_{p_r}^k + s_{p_r} + t_{p_r d_r} - M_{13}(1 - \sum_{j \in V} x_{ij}^k) \leq w_{d_r}^k,$$

$$\forall (i, j) \in A, \forall r \in R, \forall k \in K \quad (13)$$

- A vehicle leaves from its starting depot.

$$w_{o_k^+}^k + s_{o_k^+} + t_{o_k^+ o_k^-} - M_{14}(1 - \sum_{j \in V} x_{o_k^+ j}^k) \leq w_{o_k^-}^k,$$

$$\forall k \in K \quad (14)$$

- The load when leaving a node is equal to the previous load plus the load variation.

$$l_j^k \geq l_i^k + \phi_j - M_{15}(1 - x_{ij}^k),$$

$$\forall (i, j) \in A, \forall k \in K \quad (15)$$

- The load cannot exceed the vehicle capacity.

$$l_i^k \leq \sum_{j \in V} Q^k x_{ij}^k, \quad \forall i \in V, \forall k \in K \quad (16)$$

- The transportation time of a user cannot exceed his/her maximal ride time.

$$w_{d_r}^k - w_{p_r}^k - s_{p_r} \leq T_r^+, \quad \forall r \in R, \forall k \in K \quad (17)$$

- The shift length of a driver cannot exceed its maximal shift length.

$$w_{o_k^-}^k - w_{o_k^+}^k \leq T_k, \quad \forall k \in K \quad (18)$$

- The excess ride time is the duration difference between solution path and the shortest path. This definition is used to clarify the second objective notation.

$$ww_r = \sum_{k \in K} (w_{d_r}^k - w_{p_r}^k) - T_r^-, \quad \forall r \in R \quad (19)$$

- $x_{i,j}^k$ is a binary variable, w_i^k is a continuous variable, l_i^k is an integer variable, ww_r is a continuous variable,

$$\begin{aligned} x_{i,j}^k &\in \{0, 1\} & w_i^k &\in \mathbb{R}^+ \\ l_i^k &\in \mathbb{Z}^+ & ww_r &\in \mathbb{R}^+ \end{aligned} \quad (20)$$

3 SOLUTION METHOD

3.1 Sub-DARP Pareto computation

A sub-DARP is solved as a mono-objective problem. For one sub-solution computation, only the cost or the unaccepted excess ride time objective is considered.

A large neighborhood search algorithm (LNS) from the previous work on the NOMAd project has been used to find sub-solutions (Tellez et al. 2018). As our problem is multi-objective, we needed an approach to consider the two objectives. It exists three main approaches: weighted objectives, lexicographic optimization and Pareto frontier optimization (Ho et al. 2018). The weighted approach is the simplest to implement. However, it requires to know beforehand the trade-offs between the different objectives in order to implement proper weights. The lexicographic approach is appropriate when one objective is by far more important than the second. The Pareto frontier is the correct approach when a global picture of the trade-off between the objectives is needed. However, the computational time is the longest because of a larger number of solution computations. In our case, we used the Pareto frontier optimization because the trade-offs between the cost and the unaccepted excess ride time are unknown.

To find the extreme points of the Pareto front, we used a lexicographic approach. For the first extreme point, cost is minimized then its value becomes the upper boundary for minimizing the unaccepted excess ride time. The objective \vec{g} used to find this extreme point is defined in equation (21).

$$\text{lex min } \vec{g} = (\text{Cost}, \text{UnacceptedExcessRideTime}). \quad (21)$$

The objective minimization order is swapped for the second extreme point.

The lexicographic method requires to add two boundary constraints for the maximal cost value and for the unaccepted excess ride time value, respectively $Cmax$ (22), $Dmax$ (23).

- The cost is smaller than $Cmax$ and the unaccepted excess ride time is smaller than $Dmax$

$$\begin{aligned} \sum_{k \in K} f^k \sum_{i \in P} x_{o_k^+ i}^k + \sum_{k \in K} \alpha^k (w_{o_k^-}^k - w_{o_k^+}^k) \\ + \sum_{k \in K} \sum_{(i,j) \in A} \gamma^k \Delta_{ij} x_{ij}^k \leq Cmax \end{aligned} \quad (22)$$

$$\sum_{r \in R} ww_r \leq Dmax \quad (23)$$

The other points of the Pareto front are found by relaxing the constraint related to the first objective. We used the augmented-epsilon constraint method (Mavrotas 2009).

3.2 Heuristic

The computation of a solution is done in a four steps loop: separate computation, combination, verification and modification. This loop is illustrated in the figure 1.

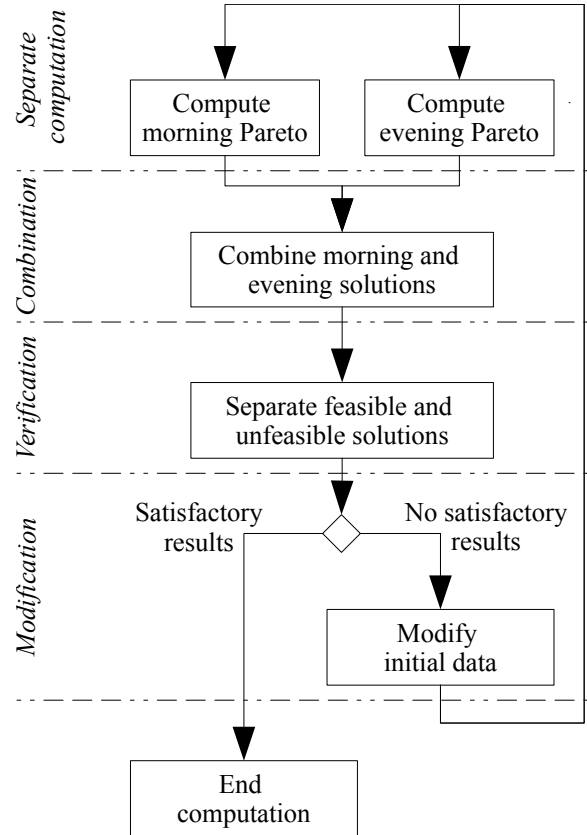


Figure 1 – Heuristic for daily solution

During the separate computation phase, the morning and evening Pareto fronts are computed using the LNS heuristic.

Then during the combination phase, all the combinations of the morning and evening sub-solutions are computed. C is the set of the solutions c from the combination of the sub-solutions. The daily cost of the solution is the sum of the costs of the morning sub-solution and the evening sub-solution. The same calculation is done for the daily unaccepted excess ride time. It is equal to the sum of the morning unaccepted excess ride time and the evening unaccepted excess ride time.

During the verification phase, the respect of the maximal daily ride time constraint is checked for every user in every solution. The maximal daily ride time constraint is formulated as in the formula (24). Every solution that does not respect this constraint is rejected. Within the set of feasible solutions, the dominated solutions are also ignored. The final set of solutions constitutes the Pareto front of the global daily feasible solution. However, it is possible that no combination constitutes a satisfactory solution (e.g., high daily cost). In that case, we proceed to the modification phase.

- Maximal daily ride time constraint

$$w_{d_r m}^{k_m} - w_{p_r m}^{k_m} - s_{p_r m} + w_{d_r e}^{k_e} - w_{p_r e}^{k_e} - s_{p_r e} \leq T_d^+ \quad \forall d \in D, \forall k_m, k_e \in K \quad (24)$$

In essence, the modification phase aims to modify the initial data to find new solutions. In the process, one or some of the maximal ride time constraints are decreased. Different strategies have been tested in order to reach a global satisfactory Pareto. Those modification strategies are described in section 4.1.5 .

4 EXPERIMENTS

4.1 Data pre-processing

4.1.1 Setting the big-M

Several big-M are used in the model. In order to restrict the research space, they are set to a minimal value. When indexes are used, their values and domain of definition are set by the corresponding constraints.

$$\begin{aligned} M_{11} &= M_{ij} = b_i + t_{ij} + s_i \\ M_{12} &= M_i = a_i \\ M_{13} &= M_r = s_{p_r} + t_{p_r d_r} \\ M_{14} &= M_{o_k^+ o_k^-} = s_{o_k^+} + t_{o_k^+ o_k^-} \\ M_{15} &= \max_{o \in P} \phi_o \end{aligned}$$

4.1.2 Setting the minimal ride time

The minimal ride time for a user is set by the formula (25). It corresponds to the greatest value between the

shortest path and the time window opening duration minus the service time at the pick-up node. The figures 2 and 3 illustrates this concept. The time window opening duration is the difference between the upper bound of the pick-up node time window and the lower bound of the delivery node time window. Indeed, if the time window opening duration is greater than the shortest path, taking the shortest path does not respect the time window constraints (12).

$$T_r^- = \min[t_{p_r d_r}; a_{d_r} - b_{p_r} - s_{p_r}] \quad (25)$$

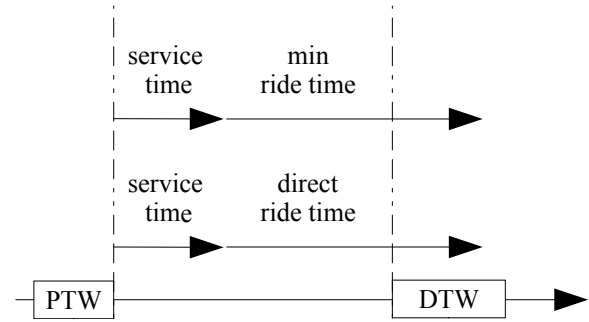


Figure 2 – Long min ride time setting

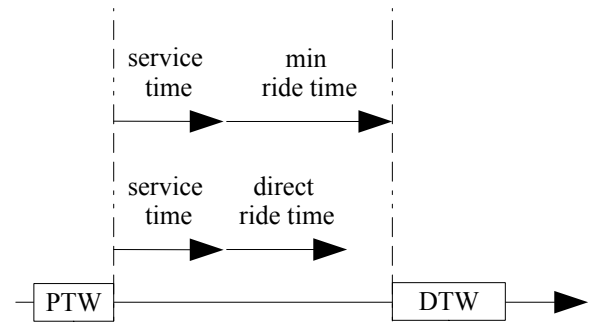


Figure 3 – Short min ride time setting

4.1.3 Setting the maximal ride time

The maximal ride time duration can be modified before the solution computation to avoid some of the feasible sub-solutions that lead to infeasible global solution. The maximal ride time is set by the formula (26). Initially, the maximal ride time is set to 150% of the minimal ride time value. If the initial maximal ride time is greater than the minimal ride time plus the maximal daily excess ride time, it can be reduced. Indeed, any ride time greater than this value in a sub-problem will not be a feasible solution for the global problem. This concept is illustrated by the figure 4.

$$T_r^+ = \min[1.5 * T_r^-; T_r^- + T_d^+ - T_d^-] \quad (26)$$

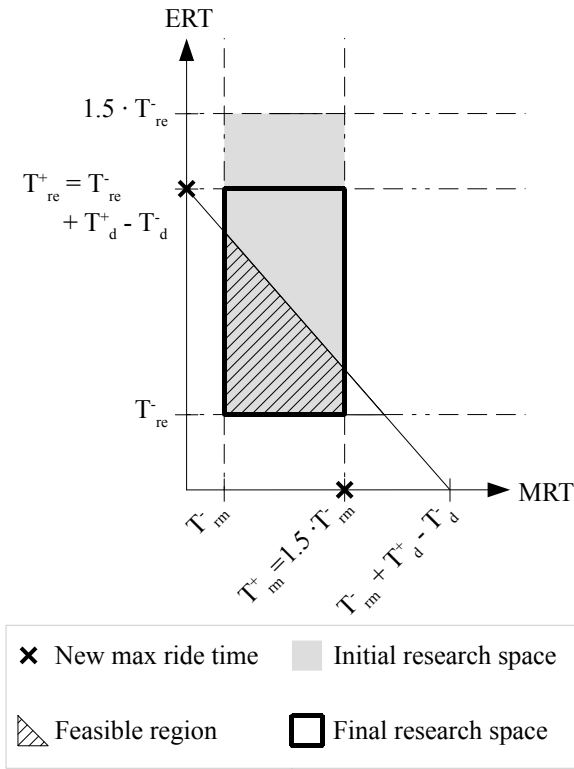


Figure 4 – Maximal ride time setting

4.1.4 Setting the maximal daily ride time (applied to NOMAd)

In the NOMAd project, the maximal daily ride time has been set to 150% of the daily minimal ride time. For example, a user shortest round-trip itinerary takes 2 hours, so his/her maximal daily ride time is equal to 3 hours.

But for users that are really close to their destination, this setting is not realistic and put too much constraint. Thus, a different setting is done for any user who resides less than 15 minutes away from his/her arrival. The maximal ride time value is changed to the minimal ride time plus 15 minutes. Thus, the maximal daily ride time value is equal to the minimal daily ride time value plus 30 minutes. For example, a user shortest itinerary takes 2 minutes, so his/her maximal ride time is set to 17 minutes (instead of 3 minutes) and his/her maximal daily ride time is set to 34 minutes.

Consequently, the maximal daily ride time is set by the following formula:

$$T_d^+ = \max[T_d^- + 30; 1.5 * T_d^-] \quad (27)$$

This method to set the maximal daily ride time is specific to the NOMAd project.

4.1.5 Description of modification strategies

The modification of the initial data has three components: data processing, selection and new maximal ride time setting.

		Strategy							
		1	2	3	4	5	6	7	8
Data proc.	avg value					x	x	x	x
	worst value	x	x	x	x				
Selection	worst user	x	x			x	x		
	bad users			x	x			x	x
Decrease	fixed	x		x		x		x	
	variable		x		x		x		x

Tableau 1 – Strategies for modifying the problem's data

Data processing aims to extract new information from the results. It can be applied or not depending on the chosen strategy. Further process is then based on the raw data or the processed data. When data processing is applied, the difference between the maximal daily ride time minus the mean of the daily ride times from all the solutions for each user is calculated ($T_r^+ - \sum_{c \in C} \frac{\text{dailyRideTime}_c}{|c|}$). The value obtained represents for each user his/her average daily ride time deviation from his/her maximal allowed daily ride time. A positive value indicates that in average, a user spends too much time in transports. A negative value indicates that in average, a user spend less than his/her maximal allowed ride time in transports. If the results are not processed, then the worst value from all the solution is considered for each user.

Selection is setting the scope of the initial data which is going to be modified. Two scopes have been explored: worst user and bad users. The worst user selection implies only selecting the user with the maximal deviation. The bad users selection implies selecting all the users that exceed their maximal daily ride time.

When the selection of users is complete, their maximal ride time value is decreased. This reduction happens only on a sub-request (morning or evening) depending on which sub-problem contributes the most to the maximal daily ride time constraint violation. For example, a user has a morning itinerary with small unaccepted excess ride time and an evening itinerary with huge unaccepted excess ride time, thus only the evening maximal ride time of this user is going to be decreased. The amount of the decrease can either be a fixed value ϵ , or a variable value ϵ_d . The variable value depends on the amount of constraint violation of the selected user. The bigger the constraint violation, the greater the reduction.

The table 1 describes the 8 strategies that have been tested. Strategy 0 designates a simple morning and evening sub-solutions combination without processing any further optimization.

5 RESULTS

The results are demonstrated using one example but other instances of the problem show similar outcomes. The example has 67 users and 134 requests. The experiments

were run on a laptop computer with an Intel(R) Core(TM) i5-4310U CPU @ 2.00 GHz processor and 7.90Go of RAM. The heuristic was coded in Python and the LNS was coded in C++.

5.1 Pareto fronts comparison

In a set of Pareto fronts, a Pareto is better than another Pareto if it has more non-dominated points on the lower daily cost interval. As we are not interested in high daily cost solutions, the higher daily cost interval is ignored by this method. The lower daily cost interval is the interval between the minimal daily cost in the set of Pareto fronts and the intermediate daily cost. The intermediate daily cost is the mean between the minimal and maximal daily costs in the set of Pareto fronts $((cost_{max} + cost_{min})/2)$. This concept is illustrated by a simple example in the figure 5. In the example, the blue Pareto has two non-dominated points in the lower daily cost interval and the red Pareto has one non-dominated points. This metric was used to compare the results.

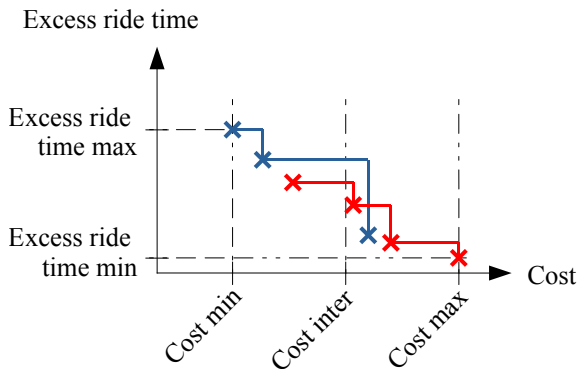


Figure 5 – Pareto fronts comparison using the intermediate daily cost

5.2 Heuristic performance and strategies comparison

First of all, whatever the strategy employed, the heuristic allow to find better solutions than a simple combination of morning and evening Pareto fronts (strategy 0). Obtained results have a wider range of non-dominated points and are more exhaustive. The figure 6 shows the obtained results for the previously cited example.

We discovered that some strategies lead in average to better solutions. Indeed, for the same parameters, some strategies give in average more non-dominated points. By running the set of experiments we had, we obtained the results presented in the table 2. The strategy 0 designate the simple combination of sub-solution without further optimization. The strategy that shows the best results is the strategy 1 in which the worst user is selected based on raw data and the amount of reduction is a fixed value. In average, this strategy allows to find 14.8 non-dominated points in the lower daily cost interval. It is five points more than the second best strategy.

Strategy	average number of non-dominated points
0	1.9
1	14.8
2	8.3
3	8.6
4	6.5
5	9.1
6	5.2
7	6.6
8	6.2

Tableau 2 – Average number of non-dominated points comparison

6 CONCLUSION

In this paper, we presented a new approach for solving a multi-trip dial-a-ride problem with dependent requests. We showed that the maximal daily ride time constraint greatly increases the complexity of the problem as it creates dependencies between the requests. The heuristic developed to solve this problem generates a set of non-dominated feasible global solution. Applied to real cases, the person in charge to organize the routes can select the best solution based on the heuristic results and his/her practical experience. Every strategy showed an improvement in the global solution. However, we discovered that the parameters of the heuristic (LNS runtime, number of computed points in a sub-Pareto, decrease value, ...) has a great influence on the final results. Further research can address this issue by finding the relationship between the parameters, the studied instance and the results. This would allow finer tuning of the parameters for each problem.

REFERENCES

- Azi, Nabila, Michel Gendreau, and Jean-Yves Potvin, 2007. "An exact algorithm for a single-vehicle routing problem with time windows and multiple routes". In: *European Journal of Operational Research* 178.3, pp. 755–766.
- Braekers, Kris, Katrien Ramaekers, and Inneke Van Nieuwenhuyse, 2016. "The vehicle routing problem: State of the art classification and review". In: *Computers & Industrial Engineering* 99, pp. 300–313.
- Cordeau, Jean-François and Gilbert Laporte, 2007. "The dial-a-ride problem: models and algorithms". In: *Annals of Operations Research* 153.1, pp. 29–46.
- Crainic, Teodor Gabriel, Michel Toulouse, and Michel Gendreau, 1997. *Towards a taxonomy of parallel tabu search algorithms*.
- Ellegood, William A., Stanislaus Solomon, Jeremy North, and James F. Campbell, 2020. "School bus routing problem: Contemporary trends and research directions". In: *Omega* 95, p. 102056.
- Gendreau, Michel, Jean-Yves Potvin, Olli Bräumlaysy, Geir Hasle, and Arne Løkketangen, 2008. "Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography". In: *The Vehicle Routing*

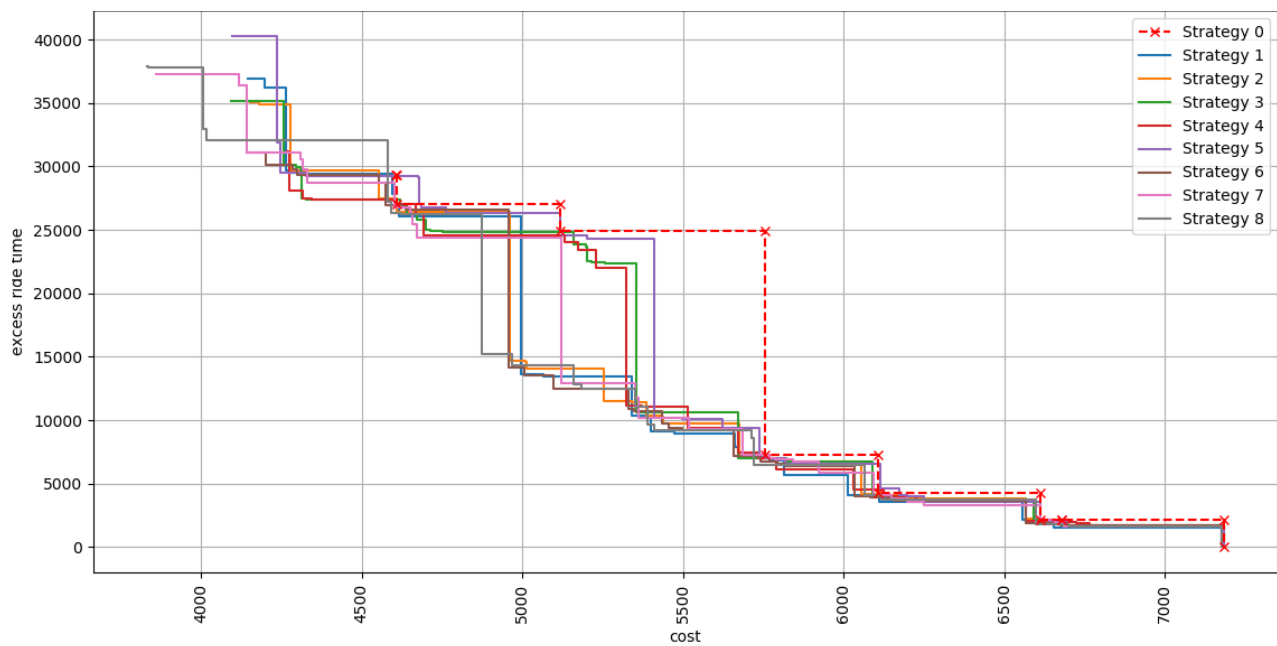


Figure 6 – Strategies comparison

Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces. Boston, MA: Springer US, pp. 143–169.

Golden, Bruce, Arjang Assad, Larry Levy, and Filip Ghysens, 1984. “The fleet size and mix vehicle routing problem”. In: *Computers & Operations Research* 11.1, pp. 49–66.

Hernandez, F., D. Feillet, R. Giroudeau, and O. Naud, 2014. “A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration”. In: *4OR* 12.3, pp. 235–259.

Ho, Sin C., W. Y. Szeto, Yong-Hong Kuo, Janny M. Y. Leung, Matthew Petering, and Terence W. H. Tou, 2018. “A survey of dial-a-ride problems: Literature review and recent developments”. In: *Transportation Research Part B: Methodological* 111, pp. 395–421.

Jaw, Jang-Jei, Amedeo R. Odoni, Harilaos N. Psaraftis, and Nigel H. M. Wilson, 1986. “A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows”. In: *Transportation Research Part B: Methodological* 20.3, pp. 243–257.

Lim, Andrew, Zhenzhen Zhang, and Hu Qin, 2016. “Pickup and Delivery Service with Manpower Planning in Hong Kong Public Hospitals”. In: *Transportation Science* 51.2, pp. 688–705.

Macedo, Rita, Cláudio Alves, J. M. Valério de Carvalho, François Clautiaux, and Saïd Hanafi, 2011. “Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model”. In: *European Journal of Operational Research* 214.3, pp. 536–545.

Mavrotas, George, 2009. “Effective implementation of the ϵ -constraint method in Multi-Objective Mathematical Programming problems”. In: *Applied Mathematics and Computation* 213.2, pp. 455–465.

Psaraftis, Harilaos N., 1983. “An Exact Algorithm for the Single Vehicle Many-to-Many Dial-A-Ride Problem with Time Windows”. In: *Transportation Science* 17.3, pp. 351–357.

Tellez, Oscar, Laurent Daguét, Fabien Lehuédé, Thibaud Monteiro, Geovanny Osorio Montoya, Olivier Péton, and Samuel Vercaene, 2020. “A stakeholder oriented approach to the optimization of transports of people with disabilities”. In: *Supply Chain Forum: An International Journal* 0.0, pp. 1–10.

Tellez, Oscar, Samuel Vercaene, Fabien Lehuédé, Olivier Péton, and Thibaud Monteiro, 2018. “The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity”. In: *Transportation Research Part C: Emerging Technologies* 91, pp. 99–123.

Toth, Paolo and Daniele Vigo, 1997. “Heuristic Algorithms for the Handicapped Persons Transportation Problem”. In: *Transportation Science* 31.1, pp. 60–71.

Ulusoy, Gündüz, 1985. “The fleet size and mix problem for capacitated arc routing”. In: *European Journal of Operational Research* 22.3, pp. 329–337.

Wong, K. I. and Michael G. H. Bell, 2006. “Solution of the Dial-a-Ride Problem with multi-dimensional capacity constraints”. In: *International Transactions in Operational Research* 13.3, pp. 195–208.