LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

Multicriteria analysis of a real vehicle scheduling problem under travel time uncertainty

Pedro Vinícius A. B. de Venâncio
Operations Research and Complex Systems Laboratory
Undergraduate in Systems Engineering, Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil
pedbrgs@gmail.com

Lucas S. Batista
Operations Research and Complex Systems Laboratory
Department of Electrical Engineering, Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, Minas Gerais, Brazil
lusoba@ufmg.br

RESUMO

Este artigo propõe uma análise multicritério para o problema de programação de veículos sob condições de incerteza dos tempos de viagem. Uma metaheurística proposta anteriormente é usada para estimar a distribuição de viagens a veículos que produza o menor custo operacional possível. Durante a otimização, todas as soluções são armazenadas em uma estrutura de memória e os mínimos locais são extraídos para uma análise posterior. Tais soluções são avaliadas em um conjunto de indicadores de confiabilidade propostos, via simulações de Monte Carlo, permitindo que as opções de projeto sejam comparadas de forma mais adequada por um método de tomada de decisão. A abordagem utilizada para otimização mono-objetivo foi validada usando dados de empresas de transporte da cidade de Belo Horizonte/Brasil e os resultados da análise multicritério mostram superioridade em relação às soluções utilizadas na prática. Além disso, os resultados mostram que nem sempre as soluções de menor custo são as mais adequadas às situações reais de incerteza.

PALAVRAS CHAVE. Otimização combinatória, tomada de decisão, análise multicritério. Tópicos: Logística e Transportes, Otimização Combinatória, Apoio à Decisão Multicritério.

ABSTRACT

This paper addresses a multicriteria analysis for the vehicle scheduling problem under conditions of uncertain travel times. A previously proposed metaheuristic is used to estimate the distribution of trips to vehicles that produce the lowest possible operating cost. During the optimization process, all solutions are stored into a memory structure, and the local minima are extracted for further analysis. These solutions are evaluated in a set of proposed reliability indicators, via Monte Carlo simulations, allowing the design options to be compared more adequately by a decision-making method. The approach used for mono-objective optimization was validated using data from transport companies of the city of Belo Horizonte/Brazil and the results of the multicriteria analysis show superiority in relation to the solutions used in practice. In addition, the results of the simulation show that not always the least cost solutions are the most appropriate to the real situations of uncertainty.

KEYWORDS. Combinatorial optimization, decision making, multicriteria analysis.
Topics: Logistics and Transportation, Combinatorial Optimization, Multicriteria Decision.

https://proceedings.science/p/106744?lang=pt-br

1. Introduction

The flow of automobiles has grown dramatically in recent years, requiring the elaboration of public policies, such as the rotation of license plates in some municipalities in the brazilian territory. Prioritizing the use of public transport in these conditions is essential to improve the daily locomotion of the inhabitants in the cities. However, the current situation of the public transport system has many challenges that are a consequence of the dynamics of cities, including demand compliance, cost reduction and service reliability.

Given the complexity of the public transport system, the planning process is usually subdivided into five sequential problems: (i) route definition; (ii) preparation of travel schedules; (iii) vehicle scheduling; (iv) crew scheduling; and (v) crew rostering [Souza et al., 2006]. The system block diagram is shown in Figure 1.

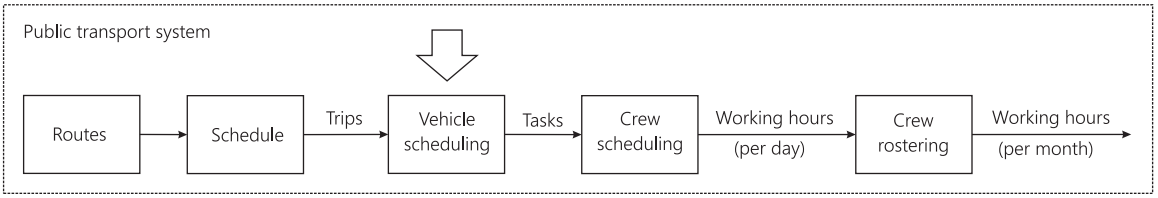


Figure 1: Relations between the subproblems of the public transport system.

The system begins in the Vehicle Routing Problem (VRP), which consists of defining a set of routes for the vehicles with the lowest operational cost possible, so that the demand of customers at each point is met. Each fleet vehicle starts and ends in a garage and has limited capacity. Several formulations of this problem have already been presented and the most common objectives are to minimize travel costs, total distance traveled and total operating time.

After determining the routes, it is necessary to define the timetable of the trips to be carried out. Like vehicle routing, this subproblem depends on external factors such as demand for the service, availability of infrastructure, and certain essential services to users. These definitions are not the scope of this paper and are considered to be known inputs.

The next subproblem and object of study of this work is the Vehicle Scheduling Problem (VSP). According to Ball et al. [1983], the problem is to define the allocation of trips to as few vehicles as possible, so that the cost of the daily operation routine is minimized. The vehicle scheduling solution results in the travel sequences to be performed by the vehicles in the minimum fleet. This, in turn, is simplified in a set of tasks assigned to drivers and on-board agents, being input from the subsequent subproblem.

Thus, the Crew Scheduling Problem (CSP) aims to determine the daily work schedules for crews, respecting the company's operational rules and labor legislation. Although the constraints are distinct, both vehicle scheduling and crew scheduling may be reduced to designation problems. In the case of the VSP, an allocation of trips to vehicles is sought, while in the CSP an allocation of tasks to crews. This allows both problems to be solved by the same algorithms, given some adaptations [Souza, 2009].

Finally, the system ends up in the Crew Rostering Problem (CRP), which consists of preparing the crews work schedules in a planning period, usually one month, minimizing fixed costs and overtime. As in crew scheduling, labor and operational constraints must also be met [Martins and Silva, 2017].

In view of the constant need to reduce costs in planning activities of the urban public

https://proceedings.science/p/106744?lang=pt-br

transport system, a solution is proposed for the VSP based on an integration between Iterated Local Search (ILS) and multiple-criteria analysis. During the optimization process, local optimal solutions are mapped to a later multicriteria analysis, where additional proposed reliability indicators are considered to ensure more assertive decision making with support of the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). The solutions found by the proposed integrated tool, considering possible operational restrictions and real scenarios of uncertainty, are quite promising when compared to the real solutions employed in the practice by the evaluated transport companies.

The remainder of the paper presents the following structure: Section 2 presents some of the main works related to the vehicle scheduling problem in the literature; Section 3 deals with the formal definition of the problem, adapted from Souza et al. [2006]; Section 4 shows all the particularities considered in the approach applied to the mono-objective optimization problem [Venâncio et al., 2017]; Section 5 presents the decision support methodology and the criteria proposed in this work; the results and their respective analyzes are discussed in Section 6; the paper is closed in Section 7 with the conclusions.

2. Related work

Vehicle scheduling is a common designation problem in the literature, where proposals based on heuristics are the most relevant [Saha, 1970; Wren, 1972; Atkinson, 1998]. Even in the face of an extensive literature, not all practical requirements were considered. Baita et al. [2000] incorporate constraints related to dead travel between terminals, line changes and terminal wait times, and compare the traditional designation method with two new heuristics based on genetic algorithms and logic programming.

More recent studies have demonstrated frequent investigations of metaheuristics to various combinatorial optimization problems. The metaheuristic Iterated Local Search (ILS), initially proposed by Baxter [1981] and later perfected by Lourenço et al. [2003], was used for the VSP and obtained better solutions than those adopted by public transport system companies in the city of Belo Horizonte/Brazil [Souza et al., 2006]. The algorithm generates an initial solution through a greedy procedure and disturbs it through exchanges and travel allocations between any two vehicles. Finally, the solution is subjected to a refinement through the Random Descent Method (RDM).

A variation of the previous approach incorporates a memory structure into metaheuristics to store all the important information of a solution, such as the distribution of trips per vehicle and the total cost of scheduling [Venâncio et al., 2017]. This allows a more efficient exploration of the search space, where only the unpublished solutions are evaluated.

The current work is an extension of the approach proposed in [Venâncio et al., 2017], in which it presents an integrated methodology regarding optimization and decision-making support. For this, three additional reliability criteria were proposed: failure rate, mean time to repair and unavailability. These criteria were evaluated in scenarios with Gaussian perturbations associated to end time of each trip and later passed as input to a decision-making method.

3. Problem statement

Since the system as a whole is represented by a cascade model, the entries in the vehicle schedule are the maximum number of vehicles in the fleet and the timesheet with all the daily trips to be performed. An example of a travel chart with its respective properties is shown in Table 1.

https://proceedings.science/p/106744?lang=pt-br

Table 1: Example of a timesheet piece.

Trip	Line	Start Time	Start Point	End Time	End Point	Boarding Time	Landing Time
1	617	5	10	35	10	1	1
2	617	10	10	40	10	1	1
3	619	10	20	65	20	2	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Every trip is identified by a number and contains start and end times, place of departure and arrival, boarding line (platform) and total boarding and landing times in minutes. Schedules are represented by the number of minutes elapsed on the day. For example, trip 3 starts 00:10 and ends 01:05. The boarding and landing times are already included in the duration of 55 minutes [Souza, 2009].

The travel times between the stopping points and the garage to the departure and arrival points are additional information provided as input. These data make it possible to determine the time spent on dead travel, that is, journeys in which the vehicles are out of operation and without passengers. Usually, these trips are made at the transitions from the garage to the starting point and the point of arrival in the garage.

In order to ensure the feasibility of vehicle scheduling, restrictions from operational policies must be adhered to. A solution is feasible only if each vehicle is capable of performing all of its travel responsibilities and if each trip is attributed to only a single vehicle. In addition, the following restrictions were considered in order to make the daily routine possible [Souza et al., 2006]:

1. All vehicles are identical and always available.
2. The vehicles always return to the original garage that is unique.
3. A vehicle should not exceed two hours in the terminal waiting for the next trip. If the waiting time exceeds this limit, the vehicle must return to the garage or go to another place where it can wait the beginning of the next trip, characterizing a double journey.
4. Before commencement and after the end of a trip, each vehicle must comply with a minimum time in the terminal equivalent to the times of boarding and landing passengers.
5. All trips of responsibility of the company, without exception, must be carried out by some vehicle present in the fleet.
6. Every vehicle must remain for at least thirty consecutive minutes in the garage per day, used for maintenance, supply and cleaning. This survey time can be completed between two trips or after the end of all trips allocated to the vehicle.
7. The same vehicle can not carry out two trips or more simultaneously, which would represent an overlap. Similarly, a vehicle may only travel if its start time is later than the end of the current trip.

Given this, it is expected that by minimizing the total cost of vehicle scheduling, there will also be a reduction in the idle time of vehicles, the number of dead trips and the number of allocated vehicles.

https://proceedings.science/p/106744?lang=pt-br

4. Mono-objective optimization

Vehicle scheduling, as well as the other subproblems of the public transport system, is a combinatorial optimization problem and therefore has a discrete and finite set of feasible solutions. However, verifying all possible solutions aiming at the best among them is impractical, since the problem belongs to the NP-Hard class and, consequently, the processing time for its resolution grows exponentially with the dimension of the problem [Bunte and Kliewer, 2009]. Given these particularities, metaheuristics are commonly used to solve the problem.

Metaheuristic is an approximate method that guides a subordinate heuristic, intelligently combining different concepts to explore the search space. In particular, the Iterated Local Search (ILS), developed by Baxter [1981], consists of an iterative solution refinement, in which it investigates, at each step, improvements in a current solution through local search and perturbations.

The initial solution, assumed to be the current solution, undergoes a disturbance and is subjected to local search, in which successive attempts at improvement are made. The new solution is then evaluated and, when the proposed acceptance criterion is satisfied, it is considered the current solution. Otherwise, the current solution is kept unchanged. The procedure is repeated until a stop condition is satisfied, such as processing time or number of evaluations in the objective function.

4.1. Solution representation

A solution s for vehicle scheduling consists of a list of vehicles, where each vehicle is associated with a list of trips to be performed by it during a day. The vehicles are arranged in ascending order of identifier and the trips in ascending order of start times.

4.2. Evaluation function

The model that describes the total cost of the assumed vehicle schedule is defined by a set of equations that qualify the solutions through penalties for undesirable situations [Souza et al., 2006]. Given a vehicle v referring to the fleet F of a certain company, T the list of its n daily trips and i and j consecutive trips of its journey, the cost of the vehicle v in scheduling is obtained as (1):

$$C(v) = CD_{g1} + CD_{ng} + \sum_{(i,j)\in T} C_{ij} \tag{1}$$

where CD_{g1} and CD_{ng} correspond to the costs of the dead trips from the garage to the starting point of the first trip and from the end point of the last trip to the garage, respectively. The cost C_{ij} is associated with the realization of the consecutive trips i and j and is calculated by (2):

$$C_{ij} = \begin{cases} CD_{ij} + CO_{ij} \text{ if } t_{ij} < 0; \\ CT_{ij} + CD_{ij} \text{ if } 0 \leq t_{ij} \leq MIT; \\ CD_{ig} + CD_{gj} \text{ if } t_{ij} > MIT; \end{cases} \tag{2}$$

where CT_{ij} is the terminal stay cost, CO_{ij} is the overlap cost between trips and CD_{ij} is the travel cost of the dead trip. MIT is the maximum time that a vehicle can remain idle at the terminal waiting next trip and t_{ij} corresponds to the time between the end of the trip i and the beginning of the trip j, after deducting the time of boarding and landing. This idle time t_{ij} can be:

1. Negative ($t_{ij} < 0$): indicates that there is a time overlap between trips i and j. This way, the associated penalties refer to the overlap cost between them and the travel cost of the dead trip.

2. Non-negative and less than or equal to the maximum allowed terminal time ($0 \leq t_{ij} \leq MIT$): indicates that after the end of the trip i, the vehicle v will be retained in the terminal waiting for the start time of the trip j. Thus, performing the trips i and j consecutively provides a cost from the penalties for off-duty travel time and terminal stay time.

3. Greater than the maximum allowed terminal time ($t_{ij} > MIT$): indicates that the vehicle v will carry a double journey, that is, it will have to return to the garage between trips i and j, and there is no need to wait at the terminal. Therefore, the cost is calculated by penalizing travel times from the end point of the trip i to the garage and from the garage to the start point of the trip j.

In practice, it is interesting that extremely time-consuming stops at the terminal do not occur and that traveling times of an off-road vehicle are not large. In order to minimize these occurrences, a cost for each dead trip CD_{ij} is attributed, according to (3):

$$CD_{ij} = \begin{cases} (\alpha_1 \times d_{ij})^{\beta_1} \text{ if } d_{ij} \leq DTL; \\ d_{ij} \times w_{DT} \text{ if } d_{ij} > DTL; \end{cases} \tag{3}$$

where d_{ij} is the dead trip time performed by the vehicle v between trips i and j, DTL is the time limit for the duration of a dead trip, α_1 and $\beta_1 \in (0, 1)$ are penalty parameters for when the duration of a dead trip is acceptable and w_{DT} represents a penalty parameter for when the duration of a dead trip is greater than the limit. In addition, a cost is also assigned for each hold in the terminal CT_{ij}, according to (4):

$$CT_{ij} = \begin{cases} (\alpha_2 \times t_{ij})^{\beta_2} \text{ if } t_{ij} \leq STL; \\ t_{ij} \times w_{ST} \text{ if } t_{ij} > STL; \end{cases} \tag{4}$$

where STL is the timeout of a terminal stop, α_2 and $\beta_2 \in (0, 1)$ are penalty parameters for when stop time is acceptable and w_{ST} is a penalty parameter for when a stop at the terminal is greater than the limit.

The dichotomy of the penalty allows a differentiation in the level of non-viability generated by undesirable situations. The exponential function associates a lower cost to acceptable times, in which it penalizes smoothly when the duration is small and more intensely when it is raised. If the times are above the proposed thresholds, the cost is higher and is measured by a linear function.

Overlapping travel makes the solution impractical and therefore there is no need to gently penalize durations smaller than a threshold. Thus, its cost CO_{ij} is obtained through the linear penalty of each minute of overlapping by w_{OT}, according to (5):

$$CO_{ij} = -(t_{ij}) \times w_{OT} \tag{5}$$

The negative sign makes the cost positive, since t_{ij} is negative when there is overlap between trips i and j.

Finally, a solution s is evaluated according to the objective function $f(.)$ elaborated by Souza et al. [2006] and presented in (6):

$$f(s) = TDJ \times w_{DJ} + TLC \times w_{LC} + \sum_{v \in F} C(v) \tag{6}$$

where TDJ is the total amount of double journeys exceeding the limit, w_{DJ} is the related penalty, TLC is the total amount of line changes made in the schedule and w_{LC} is the associated penalty. The last therm regards the sum of the costs of all vehicles allocated for the trips demanded.

https://proceedings.science/p/106744?lang=pt-br

4.3. Initial solution

Initial solution for vehicle scheduling is established by a greedy constructive heuristic. This type of algorithm generates solutions iteratively, where the choice of the element to be inserted is based on the evaluation function.

The construction of the solution has as principle the fleet of available vehicles and the set of trips to be carried out in the day, ordered increasingly by the start time. At each iteration, the first trip is removed from the set and allocated to a vehicle, where the cost produced by the objective function (6) is as small as possible. The process ends after the assignment of each trip to some vehicle, complying the feasibility restrictions.

4.4. Neighborhood structures

In order to explore the solution space of the problem, two independent movements were considered: swap and reallocation. The swap consists of exchanging a trip for a given vehicle with a trip belonging to another vehicle. In this procedure, the number of trips of each vehicle remains unchanged. Reallocation consists of transferring a trip from a vehicle to another vehicle, that is, removing a trip from a donor vehicle and inserting it into a receiving vehicle.

4.5. Perturbation

The perturbation considered consists of the application of neighborhood structures with adaptive intensities. When a solution stays stagnant in a basin of attraction, the disturbance tends to be more intense. Otherwise, the disturbance tends to be smoother.

In view of this, 12 levels of disturbance were considered, in which quantity and types of movements are varied. In situations where the *k-th* level is even, $\frac{k}{2}$ swap moves are applied. And when the *k-th* level is odd, $\lfloor \frac{k}{2} \rfloor + 1$ reallocation moves are applied. This strategy is called the parity level [Souza et al., 2006].

4.6. Memory structure

Memory structures are data structures that store relevant information during the execution of an algorithm. When linked to metaheuristics, memory structures can store data from all solutions and therefore indicate the moment it is accessed if a solution has already been evaluated, reducing computational effort, avoiding unnecessary function evaluations and maximizing the exploration potential of the search space.

The memory structure adopted for the problem was the hash table. This is to perform quick searches by registers, in which they are addressed through an arithmetic transformation on the absolute value of the search key. While the sequential search has linear temporal complexity and the binary search has logarithmic complexity, the search time in the hash table has constant complexity, that is, virtually independent of the number of records stored.

The table size M was defined as the number of bits b_{ind} used for indexing and is given by the equation $M = 2^{b_{ind}} - 1$. For real problems, $b_{ind} = 16$ and, as a consequence, $M = 65535$ empty positions [Coelho, 2016].

The record contains all the essential information of a solution to the problem, i.e., a vehicle trip distribution vector and the total cost of the solution. The position of the record in the table is determined by a function $h(.)$ that modifies the associated key. The Algorithm 1, inspired by the work of Carrano et al. [2011], presents the function considered for mapping the key to a position address of the hash table [Coelho, 2016].

https://proceedings.science/p/106744?lang=pt-br

Algorithm 1: Hash function.

Let n be the number of trips of the schedule, $primes$ a vector containing the first n prime numbers and T a vector with random permutation of the sequence $[0, M-1]$.
Key receives the solution as a vector of travel distribution.
Calculate number of addresses in table: M $\leftarrow 2^{b_{ind}} - 1$.
Initialize factor: $H \leftarrow 0$.
for $i = 1$ *to* n **do**
    Each variable is multiplied by an element of the vector of prime numbers for the calculation of the variable C = $key(i) \times primes(i)$.
    Calculate the index: t = $modulo(H + C, M + 1)$.
    Update factor: $H = T(t + 1)$.
Return: pos = $H + 1$.

The main disadvantage of the hash table is that, for any function, distinct keys can map to the same position. A common tactic to circumvent the problem, which was also used for the present work, is to use a linked list, whose purpose is to store all the records referring to the same key in the order in which they are inserted. Access to records is performed by a sequential search in the position list and, in the worst case, comparisons are made with all stored records.

4.7. Local search

The two types of local search considered for solution refinement were based on the Random Descent Method (RDM). The first variant consists of investigating the search space using only reallocation movements and the second with only swap movements.

Basically, the method analyzes arbitrary solutions, coming from a single modification in the current solution, in which the solution referring to the lowest objective function (6) value is chosen. The algorithm repeats until a maximum number of iterations without improvement is reached or solutions already evaluated are generated a consecutive number of times.

4.8. Proposed algorithm

Finally, regarding the previous steps, Algorithm 2 presents the pseudocode of the complete approach, adapted to the Vehicle Scheduling Problem with single garage and homogeneous fleet.

Algorithm 2: ILS with memory (ILS-WM) used for VSP optimization.

Construct an initial solution s_0 from the greedy constructive heuristic.
Perform the local search from the initial solution with only reallocation movements and then with only swap movements.
Update the best solution found s^*.
while *processing time less than the maximum* **do**
    Perform perturbation in s^* with intensity $level$ and assigns to the current solution s'.
    if $s' \notin$ *hash* **then**
        Perform the local search from the current solution s' with only reallocation movements and then with only swap movements.
        if $f(s') \leq f(s^*)$ **then**
            Update best solution generated so far $s^* \leftarrow s'$ and restarts perturbation level to 1.
        else
            If the perturbation level is less than the maximum level, increase the level. Otherwise, reset the level to 1.
Return the best solution found.

5. Multicriteria decision making

Deciding which is the best solution based only on operating cost is a pretty pretentious behavior. In practice, a number of other factors must be considered in order for the implementation of a solution to be successful, since complex problems are commonly related to scenarios surrounded by uncertainties. In the vehicle scheduling problem, in particular, travel may be delayed or advanced, resulting in a sequence of time overlaps. Such conflicts can make it impossible to implement the best solution found in the optimization process, just as a reasonably costly solution can easily adapt.

Given the need to ensure that the chosen solution is flexible to the adversities promoted by real situations, several local optimal solutions were stored in a list during the optimization process. Then, each solution of a given instance was submitted to a proposed multicriteria analysis, where reliability indicators were used to measure the sensitivity of the solutions. The uncertainty scenarios were generated through the Monte Carlo simulation, considering Gaussian perturbations ($\mu = 0, \sigma = 1$) at the end time of each trip present in all vehicles of the fleet. The values found for the indicators, as well as the cost of the solution, were used as criteria of the decision-making method.

5.1. Reliability indicators

The results of Monte Carlo simulation are used to evaluate the performance of each solution, considering three relevant criteria besides the operational cost.

1. Failure rate (λ): to ensure similarity to the actual public transport scenarios, the system failure was modeled as the delays generated during the trips. For example, if a vehicle arrives at the terminal 10:30 and its next trip should have begun 10:28, there was a system failure. The frequency of delays in a simulation is known as the failure rate.

2. Mean time to repair ($MTTR$): the system was considered repairable, where the repair of a delay consists of assigning the start time of the next trip as the end time of the previous trip. Mean time to repair is given by the mean of these delay times in hours.

3. Unavailability ($1 - A$): is the complement of availability. Percentage of the day in which the system is not in operation.

5.2. Multicriteria decision method

The chosen method of decision-making was The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Initially developed by Yoon and Hwang [1981], its principle is to rank the existing alternatives according to the chosen criteria. The ordering considers the best solution as the one with the least distance to the ideal solution (PIS) and, at the same time, a greater distance from the nadir solution (NIS).

The nadir solution is responsible for minimizing benefit criteria and maximizing cost criteria. The ideal solution is responsible for maximizing benefit criteria and minimizing cost criteria. In other words, the ideal solution is made up of all the best achievable criteria values and the nadir solution for all the worst values [Krohling and Pacheco, 2015].

6. Results

The experiments considered a set of eight real entry instances made available by two companies responsible for public transportation in Belo Horizonte/Brazil. The solutions employed by companies in practice were also made available for comparison purposes and the process of constructing them can be understood as a constructive heuristic based on the experience of specialists.

https://proceedings.science/p/106744?lang=pt-br

The algorithms were implemented in MATLAB R2017b. For each instance considered, 10 executions of the metaheuristic were performed with a maximum processing time of 60 minutes. In the local search, the number of maximum iterations was equal to the product of the fleet size by the total of trips, and the maximum number of consecutive solutions already evaluated was 200. For simplicity, all boarding and landing times were defined as zero. Another consideration was that the size of the fleet should not exceed the total number of vehicles used by the company's solution and that the maximum number of double journeys vehicles without a penalty is 60% of the fleet. Tables 2 and 3 present the parameters used during the mono-objective optimization [Souza et al., 2006].

Table 2: Parameters of objective function.

Parameters	Values
α_1	2
α_2	1
β_1	$\log(16)/\log(\alpha_1 \times DTL)$
β_2	$\log(16)/\log(\alpha_2 \times STL)$
w_{OT}	80
w_{ST}	40
w_{DT}	50
w_{DJ}	80
w_{LC}	5
MIT (min)	119

Table 3: Parameters per instance[a].

Instance	Fleet size	DTL (min)	STL (min)
G02-SUN-90T	11	18	73
G02-MON-260T	41	18	118
G02-FRI-260T	40	18	113
G02-SAT-172T	23	18	62
G115-SUN-108T	18	50	44
G115-MON-206T	29	50	110
G115-FRI-203T	29	50	114
G115-SAT-130T	16	50	53

[a]Instance name was standardized as [company-day of the week-number of trips].

The results of the optimization for the companies G02 and G115 are presented in Figures 2 and 3, respectively. In order to compare the average cost of the solutions found with the cost of the solutions adopted by the companies, four days of the week were assumed: days of the weekend and two working days with the highest demand (Monday and Friday). Note that the proposed metaheuristic overcomes the real solutions in all considered instances with an average percentage of 27.47% (21.97% for G02 and 32.98% for G115).

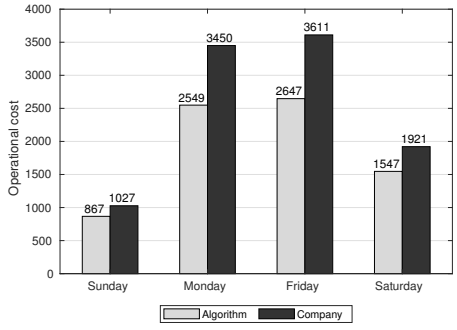


Figure 2: Company G02.

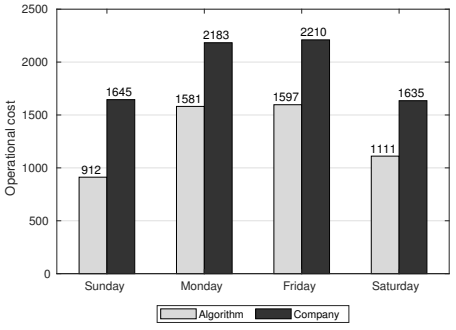


Figure 3: Company G115.

A multicriteria analysis was performed in the instance whose demand is the largest one, i.e., the Friday instance from the company G02 (G02-FRI-260T) with 260 trips and 40 vehicles available in the fleet. 20 local optimal solutions achieved by the employed metaheuristic have been submitted to a Monte Carlo simulation performed for 10000 scenarios in order to evaluate each

alternative according to the criteria. Table 4 shows the 6 of the 20 solutions that were not dominated by any other solution and their respective criteria evaluations.

The TOPSIS decision-making method was applied to rank non-dominated solutions. The weights of the criteria used were 40 % for cost, 15% for failure rate, 15% for mean time to repair and 30% for unavailability. The ordering of the solutions, presented in Table 5, follows the Closeness Coefficient (CC) from TOPSIS, which indicates how close the solution is to PIS and how far it is from NIS. The promising alternatives are those whose overall performance is closer to 1.

Table 4: Non-dominated solutions.

Solution	f	λ	$MTTR$	$1-A$
1	2640	1.226946	0.020411	0.012618
2	2645	1.206912	0.020425	0.012356
3	2648	1.169645	0.020425	0.011972
4	2649	1.198995	0.020370	0.012179
5	2650	1.201348	0.020374	0.012140
6	2670	1.174981	0.020330	0.012039

Table 5: TOPSIS ranking.

Solution	CC
3	0.9271
2	0.7630
5	0.6794
6	0.6508
4	0.4274
1	0.2063

Through this analysis it is possible to realize that in addition to the associated low costs, the solutions present extremely low failure indicators, which suggests a good adaptation of the solutions to the imposed scenarios. Another interesting fact is that the solution with the lowest cost appeared in the last position of the ranking of non-dominated solutions. One possible justification for this is that the solution probably has extremely small idle times between trips, where any kind of unforeseen event causes a major disturbance in the solution, making it infeasible.

7. Conclusions

Any approach that is able to reduce the investments of a certain company in some sector becomes of common interest. However, most of the real problems are naturally multi-criteria, and decision-making becomes increasingly difficult. The proposed integrated approach aims to find not only low-cost solutions for vehicle scheduling, but also robust solutions given the fast changes of scenery that can occur in practice.

An approximation method based on neighborhood search was employed to generate a set of nearly-optimal solutions. These solutions were submitted to a Monte Carlo simulation and the uncertainty effect was evaluated through quality indicators. Finally, non-dominated candidate solutions were ranked by a decision-making method. The procedure showed to be effective for real instances and, in particular, pointed out that conventional procedures that consider only the ideal mean-scenario may be fragile in the face of uncertainties.

Acknowledgements

This work was conducted with support from the following Brazilian agencies: the National Council for Scientific and Technological Development (CNPq) and the State of Minas Gerais Research Foundation (Fapemig).

References

Atkinson, J. B. (1998). A greedy randomised search heuristic for time-constrained vehicle scheduling and the incorporation of a learning strategy. *The Journal of the Operational Research Society*, 49:700–708.

Baita, F., Pesenti, R., Ukovich, W., and Favaretto, D. (2000). A comparison of different solution approaches to the vehicle scheduling problem in a practical case. *Computers and Operations Research*, 27:1249–1269.

Ball, M., Bodin, L., and Dial, R. B. (1983). A matching based heuristic for scheduling mass transit crews and vehicles. *Transportation Science*, 17(1):4–31.

Baxter, J. (1981). Local optima avoidance in depot location. *Journal of the Operational Research Society*, 32(9):815–819.

Bunte, S. and Kliewer, N. (2009). An overview on vehicle scheduling models. *Public Transport*, 1: 299–317.

Carrano, E. G., Moreira, L. A., and Takahashi, R. H. (2011). A new memory based variable-length encoding genetic algorithm for multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, p. 328–342. Springer.

Coelho, D. G. (2016). *Estruturas de Memória e Operadores Adaptativos em Meta-Heurísticas*. PhD thesis, Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da Universidade Federal de Minas Gerais (UFMG).

Krohling, R. A. and Pacheco, A. G. (2015). A - TOPSIS - an approach based on TOPSIS for ranking evolutionary algorithms. *Procedia Computer Science*, 55:308 – 317.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). Iterated local search: Framework and applications. *Handbook of Metaheuristics*, p. 363–397.

Martins, L. C. and Silva, G. P. (2017). Aplicação de um algoritmo genético ao problema de rodízio de tripulações do sistema de transporte público urbano. *Transportes (Rio de Janeiro)*, 25:31–40.

Saha, J. L. (1970). An algorithm for bus scheduling problems. *Operational Research Quarterly*, p. 463–474.

Souza, M. J. F. (2009). Desenvolvimento de técnicas de otimização para a resolução do problema de programação integrada de veículos e tripulações de ônibus urbano. *Relatório Técnico Científico Final FAPEMIG, referente ao projeto TEC 679/06*.

Souza, M. J. F., Silva, G. P., and Simões, E. M. L. (2006). Programação de veículos de ônibus urbano: uma abordagem heurística in: Congresso de pesquisa e ensino em transportes, 2006, brasília. *Panorama Nacional da Pesquisa em Transportes. Rio de Janeiro: Associação Nacional de Pesquisa e Ensino em Transportes*, 1:705–716.

Venâncio, P. V. A. B., Batista, L. S., and Simões, E. M. L. (2017). Investigação de uma meta-heurística baseada em memória para resolução do problema de programação de veículos. *XIII Simpósio Brasileiro de Automação Inteligente*, p. 384–391.

Wren, A. (1972). Bus scheduling: an interactive computer method. *Transportation Planning and Technology*, p. 115–122.

Yoon, K. and Hwang, C. L. (1981). TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) - a multiple attribute decision making. *Berlin: Springer Verlag*.

https://proceedings.science/p/106744?lang=pt-br