A Methodology to Create Parking Plans in a Public Transportation Company

Thé Van Luong and Éric D. Taillard

University of Applied Sciences of Western Switzerland, HEIG-VD, Department of Industrial Systems

---

Abstract

An adequate placement of vehicles in depots is essential for them to leave it smoothly for their service trips. This work presents the methodology adopted in an automated placement software used daily by one of the largest Swiss public transport companies. The problem has several hierarchical objectives and many constraints. The most important one is to place as many vehicles as possible on parking lanes. The second hierarchical objective is to maximize the number of free parking lanes, while grouping the vehicles of same type on contiguous lanes. The third objective is to spread the leaving of the vehicles while respecting timetable and topological constraints. The proposed methodology is to decompose the problem into independent sub-problems treating one objective at a time. Since the solution of a sub-problem may over-constrain the sub-problem at a further stage, several sub-optimal solutions are kept to increase the probability that one of them leads to a feasible solution for a subsequent stage.

Keywords: Public transportation; optimization; logistic; metaheuristics

---

1. Introduction

Creating parking plans for vehicles in a depot is a routine challenge for public transportation companies. The main goal of a good parking plan is to allow

---

[1] the-van.luong@heig-vd.ch, eric.taillard@heig-vd.ch

Preprint submitted to Elsevier *May 27, 2021*

vehicles to smoothly leave the depot for performing their services. However, the literature for a computer-aided generation of plans is very limited.

A tram-dispatching problem where an assignment of trams of various types in a single depot to morning departures is mathematically formalized in [1]. This simplified dispatching problem is shown to be NP-complete. A few variants of this tram-dispatching problem were introduced in [2]. In these versions, dispatching vehicles in the depot is formalized with length restrictions on lanes and generalized for a bus application.

An extension of the previous work for dispatching buses in a single depot is proposed in [3]. This model takes into consideration departure times as well as arrival times. Another application of bus dispatching is discussed in [4]. The authors introduced a few scenarios in which a solution may contain in the depot different types of buses on one lane or a grouping of the same type of buses.

Other works in the literature consider train applications that leave the depot [5, 6]. In these studies, a few cases are described to take into consideration trains composed of several train units and tracks that can be approached from two sides.

Apart from industrial problems, some depot models proposed in these previous works are rather simplified: unique vehicle length, single vehicle type, an unlimited capacity of lanes, no strict respect of departure times, no interaction between vehicles leaving the depot on contiguous lanes, allowing the repositioning of vehicles that are not well-placed, accepting departure delays and so on.

In fact, the models proposed in these works might only be restricted to academic problems.

To find other problems related to some real-world applications, one has to investigate in unrelated transportation areas such as a pickup and delivery problems [7], train scheduling problems [8], crew scheduling and rostering problems [9, 10] and passenger train operator problems [11].

This limited literature might be explained by the fact that each company has its own operational constraints and each depot its own characteristics.

So, we had to imagine a methodology from scratch when one of the largest Swiss public transportation companies asked us to design algorithms for automatically placing vehicles in its depots. The company owns several hundred vehicles of different types and has almost 10 kilometers of lanes in depots for parking them.

At first glance, the problem looks like a bin-packing, since the process is to decide which vehicle to place in which lane, knowing the length of each vehicle and the capacity of each lane.

However, numerous operational constraints make it much more complex than a simple bin-packing problem:

- With the exception of a few tram-cars, the vehicles can be driven only forward in the depots.
- Given vehicles must be placed on lanes with specific equipment such as rails for tram or electrical lines for trolleybuses.
- A vehicle cannot leave a lane if another one is in front of it; in a few situations, the vehicles on contiguous lanes may also prevent a vehicle from leaving.
- The departing hour of a vehicle must be very precisely respected, but the hour at which it returns to the depot may fluctuate.
- When a vehicle enters the depot, it must be parked in a position where it will leave the depot for the next service.

2. Problem presentation

When observing from outside a depot on a working day, we remark that few vehicles are leaving the depot very early in the morning. Then, there is a rush hour where almost all vehicles go out, roughly between 6 and 7 a.m. After the first rush hour, several vehicles come back to the depot before 10 a.m. A second rush hour occurs at noon and a third one in the late afternoon.

On Saturday, on Sunday, and during holidays, the number of vehicles engaged is limited. This is also the case on Wednesday afternoon since children are not in school at that time in Switzerland. It means that the depot manager in charge of designing parking plans must conceive several reference plans. These plans must be adapted each day to accommodate additional services that are not on the regular timetable, or temporary modifications of the timetable for some lines (e.g. due to road works). For a given day, the parking manager has to deliver three parking plans (i.e. morning, noon and evening) for each depot.

Next, observing the process inside a depot, we note that during the rush hour, hundreds of drivers must take a vehicle of appropriate type and characteristics to perform a given service, that must start exactly at the time specified by the timetable. Vehicles must go out of the depot, generally from a single door, at a frequency higher than one vehicle per minute. To guarantee a smooth and reliable process, the position of each vehicle in the depot is carefully chosen according to operational constraints (good practices imposed by the company), which may look artificial at first glance.

To avoid an ingoing vehicle having to wait for another vehicle of a different type that is late and that must be parked in front of it in the same lane, an adequate strategy is to have lanes composed of a single vehicle type. So, a vehicle entering the depot can immediately be parked at the right place for its next departure. To avoid the drivers having to look everywhere in the depot for a vehicle of the right type and characteristics, another good practice is to group the lanes containing the same vehicle type and features.

A vehicle type or model is called a *series* by the company. A physical vehicle servicing a given line following a given timetable is called a *schedule*. In addition to a set of vehicle series that must be used for a given schedule, the latter may also be associated with additional characteristics: e.g. a vehicle equipped with a video camera or an automatic ticket distributor; or the fact that the vehicle must be back to the depot before 10 a.m.

Notice a line is generally serviced by several vehicles simultaneously, each of them having its own timetable and schedule type.

The company owns several depots which can be managed independently. In fact, each schedule is assigned to a given depot and the vehicles are always supposed to return to the depot where they started their service.

2.1. Typical data set

Here is a typical example of data size for a week:

- There are a dozen distinct deliveries to deal with (e.g. Monday-Thursday morning, Wednesday noon, Friday evening or Sunday morning). The resulting parking plans contain about 1750 schedules to handle for an entire week.
- For the most constrained day, there are about 450 schedules to position. This represents about 8 kilometers of vehicles.
- There are 5 depots, for a total of 155 lanes whose length varies from 18 to 134 meters. Some depots are dedicated to tram-cars or trolleybuses while others do not contain any specialized equipment and can only be occupied by buses.
- There are 17 different vehicles series: 6 types of trolleybuses, 6 types of buses and 5 types of tram-cars. The vehicle length varies from 12 to 53 meters according to the vehicle series.
- There are currently 7 available schedule types (vehicles equipped with video camera, ticket distributor, vehicles that must return before 10 a.m., and so on).

In addition to this, all vehicles cannot be placed in garages for morning deliveries since their global capacity is not sufficient. Therefore, the logistics manager uses a few more lanes that are between the garage and the depot exit. They allow parking additional vehicles. However, these vehicles are blocking all other vehicles in the depot, meaning they must be the first to leave in the morning.

3. Problem description

This section presents the prerequisites to model and understand the problem. First are given the raw data that characterize it. Then are presented the constraints that must be satisfied to have an operating parking plan. Finally, a few good practices performed by the company are given (soft constraints), leading to better organization in the depots.

3.1. Input data

A part of the data is the set of schedules (or logical vehicles). This set is directly extracted from a timetable database. Another part of data describes the specification of vehicle series. The last part of data provides the set of lanes available in each depot.

3.1.1. Schedules

Each schedule is defined by:

- A day type that is either specific to one day (e.g. Wednesday) or a set of days (e.g. Monday-Thursday).
- A time period that determines the 3 different deliveries that occur during a day: morning, noon and evening.
- A departure depot, already established by the company.
- A departure time that is the exact hour at which a vehicle must leave the depot.
- A vehicle series that specifies its particular type (i.e. tram-car, bus or trolleybus) and model.
- A schedule type that gives additional characteristics for the vehicle (e.g. equipped with a ticket distributor) or the fact that the vehicle must have light maintenance in the evening such as a cleaning.
- A line number.

- A direction for tramways since there are lanes that can service only a subset of lines, while others can service all lines.

As input, a file extracted from the database contains a list of schedules ordered by day type, by time period and by departure time. Hence, one can consider an output file as a permutation of these schedules with the lanes assigned to each schedule.

As a consequence, a parking solution associates each schedule with:

- A lane identifier.
- A position in the assigned lane.

There are other raw data not presented here since they have no direct influence on the parking plan construction.

3.1.2. Vehicle series

The series is the main identifier of a particular model of vehicle. Each series is distinguished by:

- A vehicle type that indicates if it is a tram-car, a bus or a trolleybus.
- A length that corresponds to the vehicle length measured in meters.

For instance, the S06 series is a set of tram-cars of 44 meters and the S59 series refers to a set of buses of 18 meters. When renewing the vehicle fleet, the company generally buys several new vehicles sufficient to replace those servicing several lines. The vehicles replaced can either be sold or used on other lines. Thus, there are a few series that are considered as equivalent and have identical length, even if the actual vehicle model is not the same.

3.1.3. Lanes

A depot is a location where vehicles are stored, generally composed of a garage building, access ways, maintenance areas and administrative buildings. For the problem of designing parking plans, a depot is simply considered as a set of lanes. Basically, the characteristics of a lane are the following:

- A lane identifier associated with a given depot.
- The total length available in the lane.
- A piece of possible equipment, such as rails for tramways or electrical overhead wires for trolleybuses.
- A set of directions for tramways. The tram-cars placed on some lanes can only go in one direction (without performing a complicated maneuver that may block several lanes for a while) whereas other lanes can be switched for tram-cars to go in different directions.

3.2. Constraints

Three types of constraints must be taken into consideration in the management of the depot: first are the physical hard constraints that cannot be violated; then the topological constraints, which must also imperatively be satisfied to avoid complicated and time-consuming maneuvers; and finally, soft constraints, or good practices, must be considered as well. These last constraints may occasionally be transgressed at the price of a harder depot management.

3.2.1. Hard constraints

A solution violating any single one of these constraints is not considered as feasible by the logistics manager.

- Vehicles in a lane must be processed in chronological order. This is done according to departure times, and requires the vehicle in front to go first to let the other vehicles of the lane to leave.
- The total length of vehicles stored in a lane cannot exceed the lane capacity. Most lanes in depots are physically bounded by real walls.
- There must be a minimal space that separates two vehicles in the same lane. In practice, this distance corresponds to half a meter.

- The vehicle type positioned in a lane must be compatible with the equipment that can be found on this lane. Whereas buses can be placed everywhere, tram-cars require rails and trolleybuses need overhead wires.

- There are some vehicle series that are forbidden on a lane. This occurs for instance for mega buses that cannot be easily maneuvered. Another example is that given areas in a depot are forbidden to buses whose engines could pollute agents working on adjacent maintenance areas.

- The tramway direction must be in accordance with the lane direction. Yet only two directions exist for a schedule and a lane: either South only or North and South. For bidirectional lanes, there are no issues since any schedules can be put there. However, for unidirectional lanes, only schedules that are compatible with the lane direction can be put on these lanes. For instance, tramway schedules going South can be placed on any lane while those going North must be placed on a bidirectional lane. Figure 1 provides an illustration for these lanes.

3.2.2. Topological constraints

The topological constraints are related to a specific configuration of a depot. Even if they are not as important as the previous ones, the violation of topological constraints requires repositioning vehicles and provokes potential delays. The logistic manager wants to avoid such a scenario. Here are the two main topological constraints that can be encountered:

- Weak blocking. The vehicles in front of some lanes must leave before the vehicles located in front of other lanes. This is the case for trolley lanes where the overhead wires of three adjacent lanes join in front of the central lane. The trolleys of the vehicle in front of the central lane block the trolleys of the vehicles on adjacent lanes. A similar situation occurs for tramway lanes where the rails of three lanes join together in front of the central lane (see Figure 1 for an illustration).

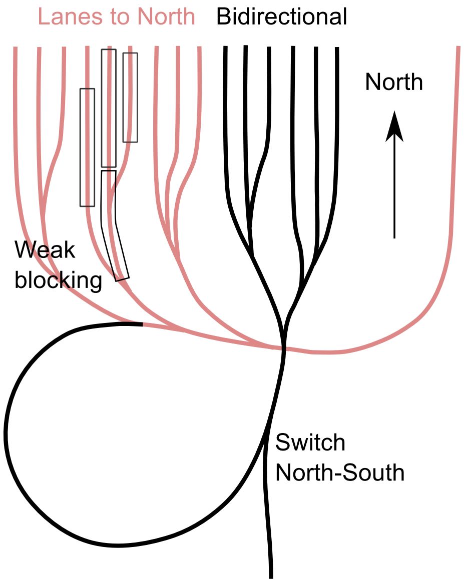


Figure 1: Portion of a tramway depot with uni- and bidirectional lanes. The figure also illustrates a weak blocking. The tram-car in front of each central lane must leave before the vehicles on the adjacent lanes.

- Strong blocking. Vehicles positioned in a few lanes block all other vehicles in the depot. This constraint is satisfied by extracting the schedules with earliest departure times and assigning them to these lanes. Such a situation happens for a few depots where buses are parked outside the garage building, in front of the doors.

3.3. Good practices

Previous constraints must be satisfied for solution feasibility. The next good practices must be respected, as far as possible, to facilitate the depot management.

3.3.1. Series and schedules grouping

A good practice is to group schedules of the same type. This improves the general management of the depot. For instance, each driver knows in which area to take his vehicle. The assignment of vehicles is also improved since it is hard to predict in advance which vehicle will return to the depot before another; the maintenance of vehicles of the same type is also facilitated.

There are groupings that must be done in priority. By order of importance, we have:

1. First of all, a lane must be occupied by a single series (or series considered as equivalent). A parking plan satisfying this constraint enables arriving vehicles at the end of a service to be immediately parked in the right place for the next service.
2. The schedules of the same schedule type (i.e. with a given particularity) should be grouped as possible in contiguous lanes. For example, the vehicles of the S59 series equipped with a video camera should be placed near each other. If all reference deliveries have an identical schedule type grouped at the same position, all employees will know where to find the vehicles with specific characteristics.
3. A few bus schedules with given line numbers should be grouped as well. For instance, the buses of lines 5 and 8 that go to the international organizations district must be flagged and it is convenient for the staff in charge of putting flags on buses to have all vehicles nearby.

3.3.2. Maximization of free lanes

A second good practice is to maximize the number of free lanes. These released lanes may be occupied by vehicles that must return rapidly to the depot. These lanes can also be dedicated to temporarily store a vehicle out of order when all repairing stations are in use. An unoccupied lane is always preferred to two lanes partially filled that could be merged into a single one with mixed schedule types.

3.3.3. Spread of schedules according to departure times

The next practice in order of importance is to regularly distribute schedules in the same lane in accordance with their departure times.

- A minimal time interval (e.g. 10 minutes) should separate two departures in the same lane. It yields a certain flexibility for the agents working on the lanes, especially when a vehicle is out of order and needs to be removed.

- A maximal time interval (e.g. 1 hour) should also be set between two departures in the same lane. This constraint prevents the last schedule to occupy a lane when all other vehicles are already outside. Since morning and noon deliveries are built independently, it also prevents a vehicle of a morning schedule that returns early (for a noon schedule) from being placed in front of a vehicle of a late morning schedule that has not yet started its service. Such a situation happens especially during the weekend since fewer vehicles are in service.

- An ideal time period (e.g. 20 minutes) is defined between two departures. The goal is to release the lane within a reasonable time so that it can be used afterwards for any other use.

3.3.4. Refinement of bus schedules according to departure times

Another good practice is the refinement of departure times for bus of a given schedule type. For each block of vehicles, the company operates two distributions of bus departures: ”Z” on weekdays and ”I” on the weekend.

Figure 2 (a) gives an illustration of a Z repartition. The idea is to consider a block of lanes containing the same schedule type and to have all vehicles in front of each lane leaving before those in the second row and so on. Such a method increases the minimal interval separating two departures.

During the weekend, since fewer vehicles are needed, I departures tend to diminish the maximal time interval between departures on the same lane. Here, schedules are placed in chronological order following an ”Indian file” repartition (one vehicle after the other), from the leftmost lane to the rightmost one (see Figure 2 (b). Such a repartition completely releases the lanes one after the other.

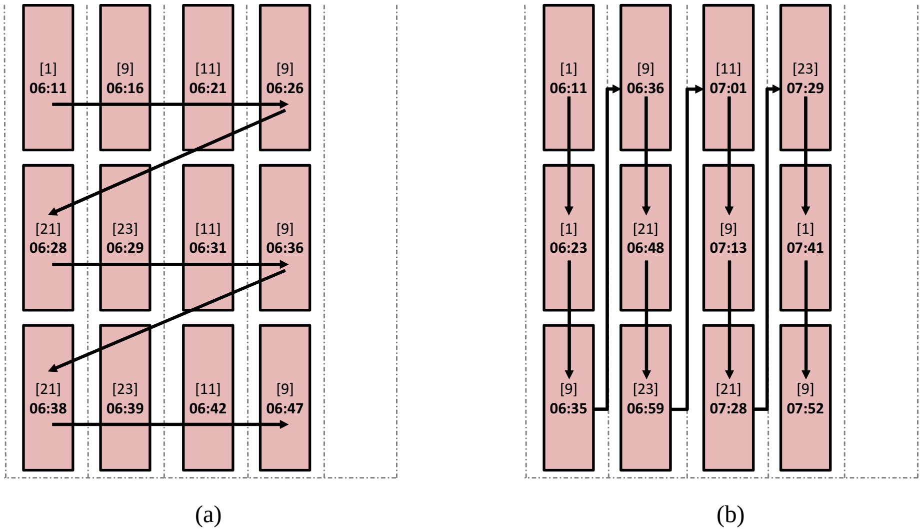


Figure 2: (a) During weekdays, bus schedules of the same schedule type are positioned following a Z repartition. In front of a block of lanes are all the earliest departures. (b) During the weekend, bus of a given schedule type are positioned in accordance with an I repartition. The leftmost lane contains all the earliest departures.

These "heuristic" repartitions on bus schedules achieved by the company might not be optimal and are more akin to habits that ensure a smooth process for vehicles leaving the depot.

Since trolleybuses and tram-cars are subject to topological constraints, these repartitions cannot be applied.

The reader interested by a mathematical description of the problem can find an integer linear programming model in [**?** 12].

4. Methodology to construct a parking plan from scratch

This section presents the different steps of a methodology to generate reference parking plans. The input of the problem is a file that contains all schedules. Providing a parking plan consists of producing a permutation of these schedules with a lane identifier and a position in this assigned lane for each schedule.

The construction of a solution can be processed independently for each depot since schedules are pre-assigned to a given depot by the company.

For the most constrained delivery and depot, the instance to solve can have more than 200 schedules and there are about 300 feasible positions on lanes for each bus schedule. The number of combinations is huge and a traditional permutation-based approach used, for example, in metaheuristics might fail to deliver a solution compatible with production requirements.

A key issue when solving a problem of large size with numerous constraints is to decompose it into a cascade of sub-problems easier to solve. This reduces the search space of the combinatorial problem which is prohibitively large. The sub-problem at the highest level is solved. Its optimal solution is provided as data for the sub-problem at the next level. If no feasible solution can be found at a level, it generally means that the solution produced at the previous level is too constrained and should be modified.

This decomposition approach has similarities with column generation. It is motivated by the hierarchical objectives presented before:

1. The grouping of schedules with identical vehicles series.
2. The maximization of free lanes.
3. The grouping of schedule type for the same vehicle series.
4. The respect of minimal time periods between two departures.

As a consequence, the resolution is divided into two major steps: the group positioning problem and the schedule assignment problem.

The first one is to maximize the grouping of vehicle series on lanes (objectives 1. and 2.).

Once the vehicles are placed on the lanes, the second major step is to assign departure times on vehicles and to refine the position of these schedules (objectives 3. and 4.).

4.1. Group positioning of vehicles series

The group positioning requires first finding the best arrangement of vehicles on lanes that maximizes the grouping of vehicles of the same series, then the number of free lanes and finally the occupied space. This approach is exclusively based on the vehicle length; other particularities such as schedule type are discarded at this step.

Let us mention a piece of general optimization software like *Gurobi* is only able to rapidly find a solution to position groups of vehicles in the depot for small instances (at least with default parameter settings).

For the largest instances the company faces, a general solver is not able to produce solutions in the reasonable computational time imposed by the transportation company. Such experiments are highlighted in [12].

For the positioning of groups of vehicle series, the key idea of the algorithm is to create contiguous blocks of schedules with the same vehicle series and to find the best position of these blocks on available lanes. Blocks of schedules are considered in this approach as an inseparable entity.

The goal is to arrange all the blocks of vehicles, while respecting hard constraints, by first, in priority, maximizing the number of unoccupied lanes and then, as a second hierarchical objective, the occupied space on each lane.

Since there are not so many distinct vehicle series involved in a depot, an exact method (advanced backtracking) is generally able to achieve this. Nevertheless, in a few instances an optimal solution might not be found within a reasonable computational time. In this case, the best positioning found so far is returned.

Figure 3 illustrates the positioning of groups of vehicle series as blocks. Each rectangle illustrates a vehicle length associated with a schedule. In this example, the three blocks of schedules represented by the S37, S38 and S50 series were computed in such a way that the number of unoccupied lanes is maximized without cutting any vehicle series.

The result of this first phase satisfies the strong blocking topological constraints and few good practices: total length of vehicles on a lane (taking into account spacing between the vehicles), lane equipment and topology adapted to vehicle series, tramway direction (e.g. if there are schedules for the series that must go to North the group must contain enough bidirectional lanes) and

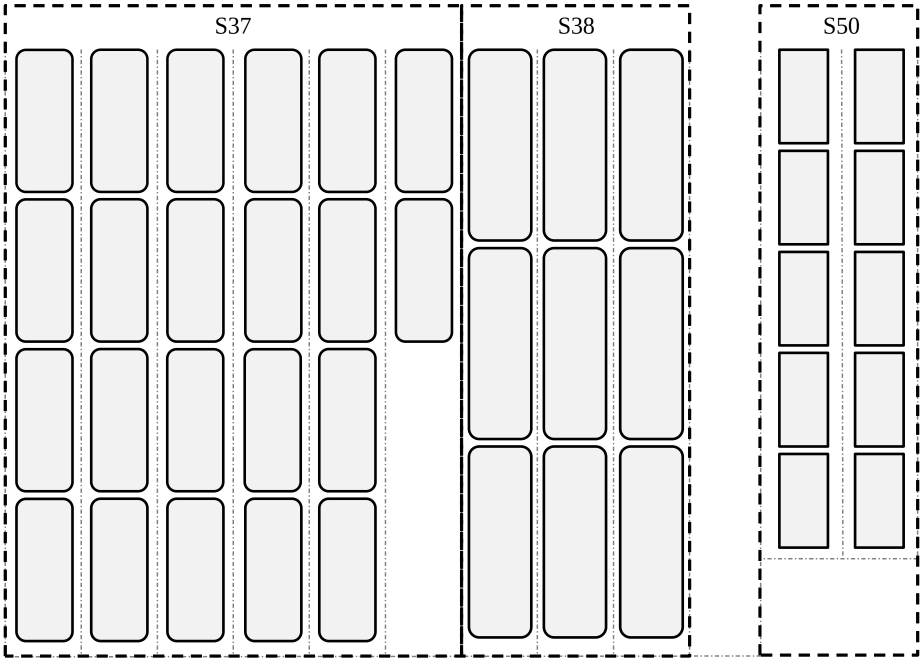


Figure 3: Positioning of groups of vehicle series in a sub-depot. It consists of finding the best position of each group of vehicle series that maximizes the number of free lanes and then the occupied space.

a single vehicle series on a lane.

4.2. Schedule assignment of vehicle series

Once all groups of vehicle series are placed, the second step of the resolution process is the assignment of schedules to these groups. It must be done in such a way that the remaining hard and topological constraints are respected: chronological departure times on each lane, tramway direction and weak blocking (i.e. vehicles in front of some lanes that must go first before vehicles in front of adjacent lanes).

To solve this problem, one needs to find the best permutation of schedules that meets a few criteria for each group: the grouping of schedules by types, a time period between two departures in the same lane and respect of topological constraints.

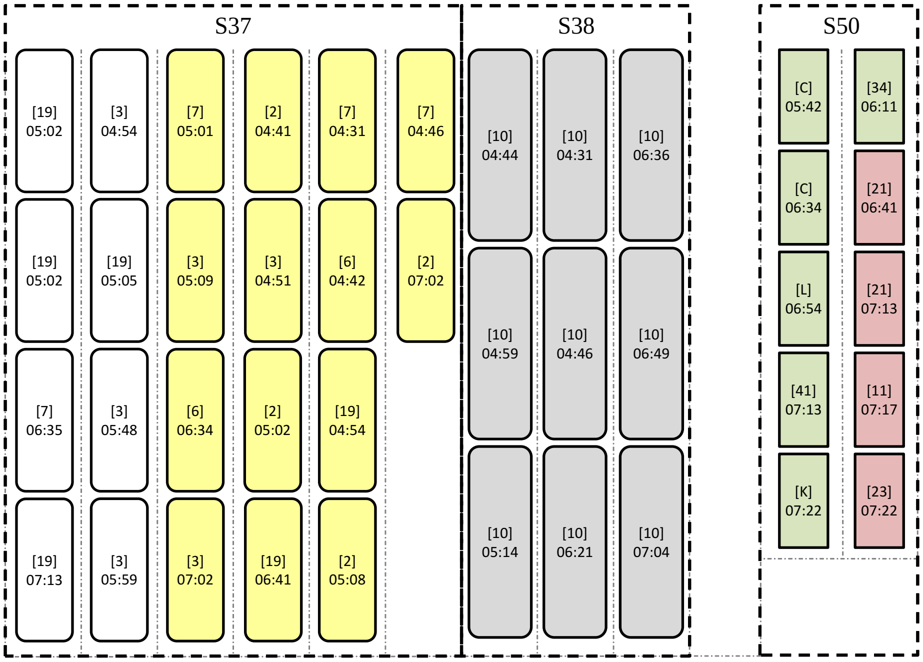


Figure 4: Assignment of schedules for each group in a sub-depot. Once the groups of vehicles series are assigned, the next step is to find the best assignment of schedules that fulfils several criteria.

Figure 4 gives an example of such a schedule assignment.

Each schedule is associated with a rectangle. The color of each rectangle determines the schedule type. Characters in brackets represent the line number and the departure time is written just below.

Since the assignment of schedules is executed independently for different series, the number of schedules to consider for a given group is relatively limited and can be solved exactly, at least for the smallest depots.

A branch-and-bound implemented with a depth-first search rapidly finds a feasible solution. If an optimal positioning cannot be found, a second phase consists of refining the position of schedules by applying small perturbations (pair-wise exchanges). This is done by a taboo search metaheuristic.

Good practices are taken into account by defining a hierarchical objective function that assigns a penalty for each practice non-respected. A large penalty is given if schedule types are not clustered (e.g. mixed schedules on the same lane or all vehicles of a schedule type not on adjacent lanes). A smaller penalty is given if the time spacing between two departures is not between the minimal and the ideal values. The penalties are defined in such a way that hierarchical objectives are obtained.

4.3. Pre-processing of a few depots

In a few situations, a pre-processing of the data is required before the general resolution.

Indeed, there may be some special lanes to release early either because they are used by the staff to park their private cars or because vehicles on these lanes are blocking the entire depot. Such a situation only happens for the earliest delivery in the morning.

To achieve this, the first departures of the day and these special lanes are selected and isolated. Then, group positioning and schedule assignment steps are applied to these subsets of lanes and schedules.

4.4. Post-processing to group bus lines and repartitions

The depots that contain buses may require a post-processing to group specific bus schedules with particular lines. This is typically the case of the buses of lines 5 and 8 that go to the international organizations district.

The idea is to run a taboo search from the partial solution found at the previous stage (i.e. schedule assignment) with a slightly modified objective function that takes into account the grouping of special lines without breaking the schedule type groupings previously found.

Then, the last step consists of refining the distribution of bus schedules in accordance with bus repartitions mentioned in Section 3.3.4.

The Z or I-departures are done separately for the lanes that only contain schedules with lines to group and for the lanes that do not contain any schedule with lines to group. This is not done otherwise.

4.5. Error processing and group cut

At each phase of the parking plan construction, 3 types of failure may occur:

1. There are more vehicle lengths than available places on lanes.
2. It is impossible to only place inseparable blocks of vehicle series that meet hard constraints.
3. The decisions taken at the group positioning step may over-constrain the problem for the schedule assignment step.

In the third case, the schedule assignment phase is repeated by modifying the group positioning. Up to 10 different group positioning are considered, corresponding the 10 best solutions found at the group positioning phase (all these solutions must have an identical number of free lane).

In case no solution is found, groups are cut during the group positioning phase. The idea is to find the best position for the first group of vehicle series with the most constraints on all available lanes, then the best arrangement for the second group on the remaining lanes and so on.

The best position (resulting into an entire block or a cut group) is defined by determining the minimal number of lanes necessary for this group. Then, as a second hierarchical objective, cuts are done in such a way that: a) the distance of grouping of vehicle series in the lanes is minimized; and b) the occupancy of each lane is maximized.

In practice, the groups that are likely to be cut are the series involving the most vehicles and the simplest constraints, i.e. buses.

An exact method based on advanced backtracking is executed to find the best cut of groups from the criteria mentioned above. In very few cases when an optimal solution cannot be found within a reasonable time, the best solution found so far is returned.

Once all sub-groups of vehicles are positioned, the phase of schedule assignment is applied on these groups. If a feasible solution cannot be produced, an error procedure is executed.

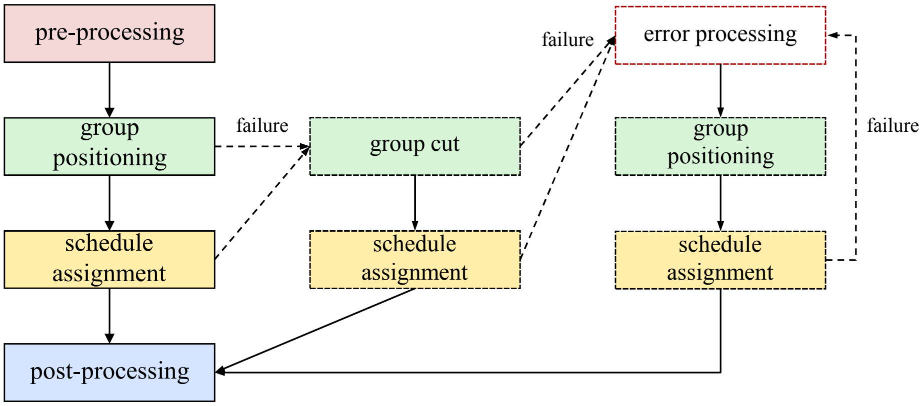


Figure 5: Current steps of the methodology to construct a new parking plan from scratch. Cutting groups of vehicle series is performed when a feasible solution cannot be constructed. In the worst scenario, an error processing mechanism ensures the construction a partial solution by removing a few schedules.

The last consists of removing few schedules and then re-launching the solving process. This operation may be iterated until a (partial) feasible solution is found. Thereafter, the logistics manager manually repositions the schedules that have not been assigned automatically.

Such a procedure might also be executed earlier if there are more vehicles to place than available places.

Figure 5 sums up what is done in our methodology to construct a new parking plan from scratch.

5. Experiments

The methodology presented in this paper was implemented in a piece of software designed for the company. It is daily used and has generated thousands of parking plans over the past six years.

This section gives some insight into the problem characteristics the company faces and some measurements of the quality of generated parking plans.

Table 1: Configuration of the different depots. The lane length is correlated with the instance difficulty.

	Depot				
	DB	DJ	DC	DK	DV
# lanes	29	70	43	6	11
Total length [m]	2982	4649	993	405	589
Average lane length [m]	102	66	23	67	53
Tram (22, 31, 42, 44 and 53 meters)	√		√	√	
Trolleybus (18, 19 and 24 meters)		√			
Bus (12, 18 and 24 meters)		√	√		√
# topological and blocking constraints	24	20	28	1	1

5.1. Depots configuration used for the experiments

Table 1 provides details on the different depots managed by the transportation company. The various lanes presented are those used every day.

The number of lanes, the total length of the depot or the average length of a lane basically determine the difficulty of the problem to solve for a given depot. A depot might contain lanes with distinct vehicle types and equipment (e.g. overhead wires on some lanes and no equipment on others).

The number of topological and blocking constraints is composed of the lanes that go to a specific direction for tramways, weak blocking that might occur between vehicles in adjacent lanes (tramways and trolleybus) and vehicle series that are forbidden on some lanes (e.g. mega buses that cannot be easily maneuvered).

All the experiments were done on a MacBook Pro Retina Intel Core i7-3840QM 2.8 Ghz with 16 GB ram using Mac OS X Sierra. Although this is a quad core laptop, only one CPU core was used for the experiments.

The algorithm parameters were set up by the company in such a way that a solution for a complete week (about sixteen parking plans) takes less than an hour on a desktop computer.

5.2. Performance evaluation

To give an idea about the performance of the algorithms, Figure 6 compares a parking plan generated by the software with one manually designed by the logistics manager before starting the project.

This figure illustrates only the half of the largest depot (DJ). Both parking plans respect all the hard constraints such as the equipment compatibility, the topology of the depot and the blocking between vehicles that occur.

This figure clearly shows that the algorithm is able to group the vehicles while maximizing the number of free lanes and the occupied space. The plan manually designed contains mixed schedule types and has no free lanes. The occupied space is less optimized.

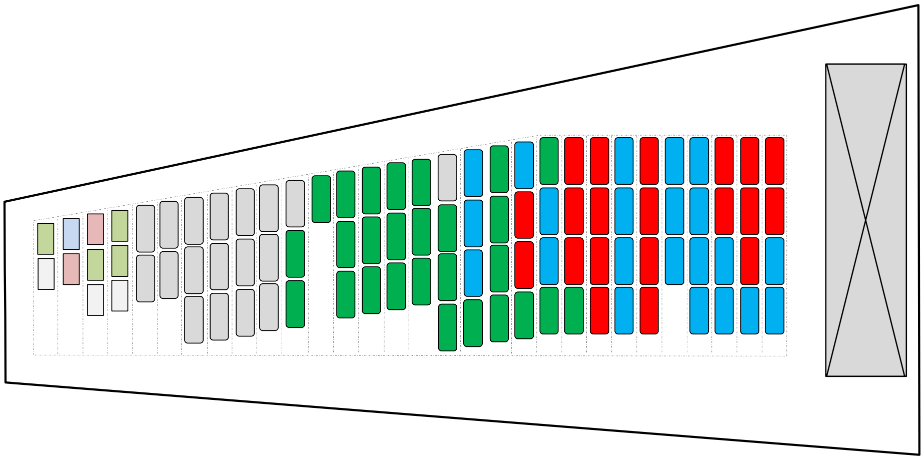
Table 2 reports all the quality differences for these two plans. The criteria evaluated by the logistics manager are exactly the hierarchical objectives mentioned in the previous section.

For the first objective, the manager was not able to manually group some schedules in the same series. This does not occur for the plan that is generated by the software.

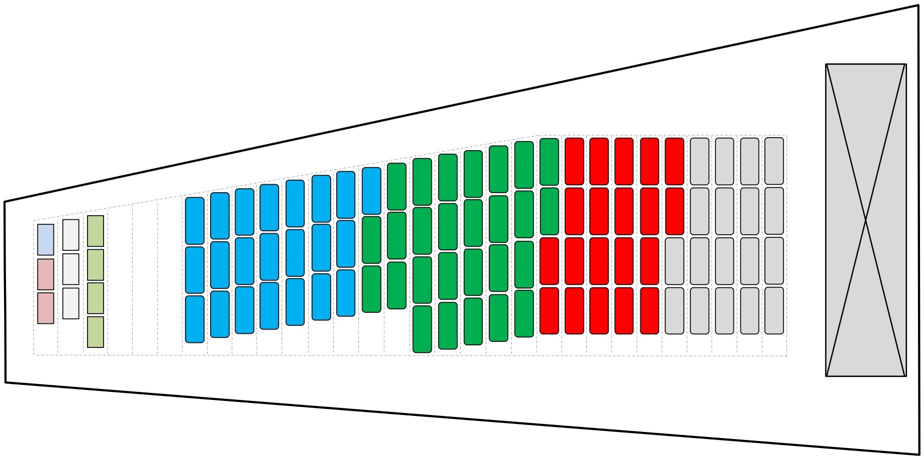
For the second objective, the software obtains more free lanes with more space left: 12 free lanes with a total of 414 meters left instead of 8 lanes with 166 meters.

The two other measurements taken into consideration are the number of schedules of the same vehicle series non-grouped by schedule types and the number of schedules that do not respect the minimal time period between two departures. Since the software proceeds in two phases, good practices for the schedule types grouping and time periods cannot be entirely fulfilled. However, the resulting parking plan is much better than the manually designed plan.

Last but not least, the program is able to automatically generate the plan in a couple of minutes whereas the logistics manager needed a couple of days to design a plan. This is a true success story according to the company.



(a) Plan manually designed by the logistics manager: mixed schedule types and occupied space not fully optimized.



(b) Plan automatically generated by the application: grouped schedule types and free lanes at disposal.

Figure 6: Parking plans for the DJ depot for the most constrained delivery in year 2011. Only half of the depot is illustrated here.

Allowing the software to run of a couple of hours might improve the last hierarchical objective (i.e. respecting minimal time periods) but benefits are quite marginal.

Table 2: Measurements of the quality of a parking plan manually designed by the logistics manager with one automatically generated by the application.

	Parking plan 2011	
	manually designed	automatically generated
# schedules in depot DJ	198	
# vehicle series	9	
# schedule types	4	
# schedules involved in non-grouped series	10 5.0%	0 0%
# free lanes in depot DJ	8 (166m)	12 (414m)
# schedules non-grouped by schedule types	56 26.7%	2 1.0%
# schedules not respecting time between departures	51 25.8%	8 4.0%
# time	few days	∼ 4 minutes

5.3. Quality measurements for reference parking plans

To assess the performance of our application, six instances were selected. They concern morning schedules, which are the hardest problems. We intentionally decided not to report about noon, evening and weekend deliveries since they are relatively small and non-challenging instances that can be solved rapidly in an exact manner.

Four instances for year 2014 and two for year 2015 are considered. These instances were chosen because they correspond to the most constrained reference parking plans that are regularly used as a basis to generate thousands of other plans for years.

They were directly extracted from the *HASTUS* system that the transport company uses. More details about this commercial software are related in [13].

Table 3 reports results obtained for different instances involving all depots.

The execution time for one run (solving the problem for all depots) lasts less than 15 minutes in any cases.

First are the number of schedules which differs depending on the period of the year (e.g. normal periods, holidays and special events). For example, during the international motor show event (2014-auto), with up to 130,000 visitors a day, many more vehicles are required. This number is basically an indicator of the difficulty of the instance to solve.

Then are the number of different vehicle series and schedule types that are considered for each instance. Although they do not vary that much, a small increase makes the resolution process much harder.

The second part of Table 3 depicts the hierarchical objectives aforementioned. A cutting phase occurs for 4 instances when inseparable blocks of vehicle series cannot be obtained, especially when there are too many topological and blocking constraints, interdictions of series on lanes and tramway direction to respect. However, the percentage of schedules involved in cut series is always less than 8% of the total number of schedules.

Schedule types grouping and time periods cannot be entirely fulfilled since the software aims to maximize in priority the number of free lanes during the group positioning phase.

However, the resulting parking plans still entirely satisfy the company.

In very few situations, the program might not be able to generate a complete parking plan with all schedules positioned in the depot. It occurs for the 2014-auto event instance with the second depot (DJ), where there are 9 schedules (162 meters) not positioned among the 238 schedules, and no free lane left (the total length of the depot is 4649 meters). Such a situation happens when the problem is infeasible because there are more vehicles to position than available length in the lanes.

6. Conclusions

This article has introduced an industrial problem that consists of creating parking plans for a public transportation company. The resulting software is

Table 3: Measure of the quality of generated parking plans for different instances. An identical lane configuration was used for all the instances.

	Instance					
	2014-1	2014-2	2014-3	2014-auto	2015-1	2015-2
# scheds	323	320	336	355	324	324
# scheds in DB	49	51	53	49	53	53
# scheds in DJ	206	202	214	238	202	202
# scheds in DC	34	33	35	34	35	35
# scheds in DK	4	4	4	4	4	4
# scheds in DV	30	30	30	30	30	30
# vehicle series	16	17	17	16	17	17
# schedule types	6	7	7	6	7	7
# scheds involved in non-grouped series	0	14 4.3%	24 7.1%	0	24 7.3%	25 7.7%
# free lanes in DB	4 (408 m)	2 (204 m)	2 (204 m)	4 (408 m)	2 (204 m)	2 (204 m)
# free lanes in DJ	12 (792 m)	13 (858 m)	10 (660 m)	0	13 (858 m)	13 (858 m)
# free lanes in DC	9 (207 m)	9 (207 m)	9 (207 m)	9 (207 m)	8 (184 m)	8 (184 m)
# free lanes in DK	5 (335 m)	5 (335 m)	5 (335 m)	5 (335 m)	5 (335 m)	5 (335 m)
# free lanes in DV	0	0	0	0	0	0
# scheds non-grouped by schedule types	16 4.9%	14 4.3%	21 6.2%	20 5.6%	22 6.8%	30 9.2%
# scheds not respecting time between departures	15 4.6%	15 4.7%	17 5.1%	14 3.9%	20 6.2%	16 4.9%
# scheds not positioned automatically	0	0	0	9 (162 m) 2.5%	0	0
Computational time [s]	450	645	790	495	750	825

able to produce solutions that meet all the expectations of the company for all deliveries of the week with a limited computational time.

A methodology to solve this complex problem is presented. The key to the success is a right decomposition of the problem into sub-problems that can be efficiently solved with exact methods, in most cases.

Since the solution of a sub-problem may over-constrain the sub-problem at a further stage, several sub-optimal solutions are kept during the process, increasing the probability that one of them leads to a feasible solution for a subsequent stage.

The algorithm provides a solution in every case, sometimes a partial one if the problem is infeasible. This occurs, for example, if the total length of the vehicles is higher than the capacity of the parking lanes available in the depots. In such a case, some vehicles are discarded, one after the other, until a feasible partial solution is found. The depot manager may then manually find a solution such as temporarily storing the vehicle outside the depot or in a repair area.

The software was used daily with satisfaction by the company for several years. Since 2015, the company has produced thousands of departure schedules with the software.

The reader interested by detailed problem formulation and algorithms descriptions can refer to [12].

We focus more on the reproducibility of methods during the group positioning and schedule assignment phases and additionally we have published instances that were solved[2].

For the large instances that the company faces, we have tried to highlight the difficulty of the problem by showing a general optimization software is not able to deliver a solution in a computational time compatible with production requirements.

---

[2] http://mistic.heig-vd.ch/taillard/problemes.dir/problemes.html

Acknowledgement

The authors thank the Geneva Public Transportation (TPG) for providing problem instance and their feedback about the use of the software. This work was partially supported by the Engineering and Business School of the State of Vaud (HEIG-VD), Project HES-SO 38782 VPM.

[1] U. Blasum, M. R. Bussieck, W. Hochstättler, C. Moll, H.-H. Scheel, T. Winter, Scheduling trams in the morning, Math. Meth. of OR 49 1 (1999) 137–148.

[2] T. Winter, U. T. Zimmermann, Real-time dispatch of trams in storage yards, Annals of Operations Research 96 (1-4) (2000) 287–315. `doi:10.1023/A:1018907720194`.

[3] G. Gallo, F. D. Miele, Dispatching buses in parking depots, Transportation Science 35 3 (2001) 322–330.

[4] M. Hamdouni, G. Desaulniers, O. Marcotte, F. Soumis, M. van Putten, Dispatching buses in a depot using block patterns., Transportation Science 40 3 (2006) 364–377.

[5] R. Freling, R. Lentink, L. Kroon, D. Huisman, Shunting of passenger train units in a railway station., Transportation Science39 2 (2005) 261–272.

[6] L. G. Kroon, R. M. Lentink, A. Schrijver, Shunting of passenger train units: An integrated approach, Transportation Science 42 4 (2008) 436–449.

[7] P. Toth, D. Vigo, Heuristic algorithms for the handicapped persons transportation problem, Transportation Science 31 (1) (1997) 60–71. `doi:10.1287/trsc.31.1.60`.
URL `http://dx.doi.org/10.1287/trsc.31.1.60`

[8] A. Caprara, M. Fischetti, P. Toth, Modeling and solving the train timetabling problem, Operations Research 50 (5) (2002) 851–861. `doi:10.1287/opre.50.5.851.362`.
URL `http://dx.doi.org/10.1287/opre.50.5.851.362`

[9] A. Caprara, P. Toth, D. Vigo, M. Fischetti, Modeling and solving the crew rostering problem, Operations Research 46 (6) (1998) 820–830. `doi:10.1287/opre.46.6.820`.
URL `http://dx.doi.org/10.1287/opre.46.6.820`

[10] S. de Groot, D. Huisman, Vehicle and crew scheduling: Solving large real-world instances with an integrat
in: M. Hickman, P. Mirchandani, S. Voß(Eds.), Computer-aided Systems in Public Transport, Vol. 600 of Lecture Notes in Economics and Mathematical Systems, Springer Berlin Heidelberg, 2008, pp. 43–56. `doi:10.1007/978-3-540-73312-6_3`.
URL `http://dx.doi.org/10.1007/978-3-540-73312-6_3`

[11] V. Cacchiani, A. Caprara, P. Toth, Solving a real-world train-unit assignment problem, Math. Program. 124 (1-2) (2010) 207–231. `doi:10.1007/s10107-010-0361-y`.
URL `http://dx.doi.org/10.1007/s10107-010-0361-y`

[12] Luong, Taillard, Formulation and algorithms for creating parking plans in a public transportation company, Tech. rep., HEIG-VD (2019).

[13] J.-M. Rousseau, J.-Y. Blais, Hastus: An interactive system for buses and crew scheduling., Computer Scheduling of Public Transport–2 (1985) 45–60.