Optimal Scheduling of Electric Vehicle Charging

Zhanwei Yu and Di Yuan

Department of Information Technology, Uppsala University, Sweden

Contents

1 Introduction

In the recent years, there is a great amount of efforts in moving toward electrical buses in Sweden [1–3] . To this end, many system components, ranging from new technical solutions to business models, have to be developed, tested, and deployed.

This is a deliverable within research and innovation project Electrified Passenger Transport in Smart Cities (Swedish: Elektrifierade persontransporter - en del av den smarta staden, [4]). The deliverable focuses on scheduling of battery charging of electrical buses at a bus depot. The charging schedule of a single bus specifies when and how much to charge the battery, for the time period when the bus is at the depot (i.e., the time since its arrival to the depot and its scheduled departure time for next service). As there are many buses and they share the charging infrastructure at the depot subject to electricity power limit, optimization algorithms become necessary for scheduling. Using optimization, the schedule ensures that the individual buses are charged such that there is a sufficient amount of battery level for service, and the schedule is optimal in efficiency. Here, the efficiency can be represented by the time before the charging is complete for all buses, the charging cost (if the electricity price varies over time), and the (total) power limit that is required at the depot of the bus operator.

It should be remarked that optimal scheduling for electrical buses falls into the domain of discrete optimization, or more specifically, combinatorial optimization [5]. There are a couple of sources behind this. First, typically the total power available does not allow all buses to be charged simultaneously. Hence, at each time moment we fact the discrete choice of selecting which buses to charge. Moreover, some charging station types can only provide a few, discrete power levels for charging. These, together with that the problem elements are strongly intertwined, make this optimization problem challenging. In this deliverable, we present problem modeling to enable solving small-scale scenarios rapidly using integer programming, and notions that are designed for large-scale scenarios. We also use two case studies to illustrate the optimization outcome.

2 Problem Description

2.1 Preliminaries

In system modeling, time is sliced into time slots. The charging solution (i.e., which buses are being changed and at what power level for each individual bus) does not change within a time slot. Hence, the duration of a time slot represents the granularity in scheduling optimization. Making the duration shorter potentially enables a better schedule, though the problem size will grow. We use 15 minutes as the baseline, however the optimization approaches are not dependent on time slot length. Later on, $\mathcal{T}$ is used to denote the set of time slots.

The buses may have to share charging stations. For example, a station may be able to accommodate two buses with two separate charges sharing the power available at the station. Thus which buses are paired at a station has impact on the optimal schedule. Moreover, a depot may have restriction on the flexibility in parking. Specifically, due to the layout of the depot, the buses may have to park in columns, one after another. Another aspect is the assignment of service routes to the buses. As the service routes differ in distance, the required state of charge (SoC) is dependent on this assignment. Note that with column parking, there is a further restriction in the assignment, namely the departure times of the buses parked in a line have to be in ascending order.

In the optimal scheduling problem we address, parking as well as route assignment are considered as problem input. There are a couple of good reasons for doing so. First, having parking and route assignment as part of the scheduling problem makes the optimization more complex. Second, parking layout varies by depot. If necessary, parking optimization can be carried out as a separation task.

In its basic form, the optimization task is subject to the following constraints, some of which are discussed further later.

- A bus can be charged only after its arrival.
- Each bus reaches its required SoC before the departure time.
- The power limit of each individual charging station, if present, is adhered to, and the total power available for charging at the depot is not exceeded.
- At any time and bus, the power used for charging (if charging at all) is one of the levels permitted by the charging station, and, if buses share a charging station, together the power may not exceed the limit of the station.

Remark 1. *The deliverable focuses on optimal charging schedule for buses at the depot overnight. That is, the buses subject to charging arrive in the late afternoon or evening, and depart in the next morning. In reality, some buses will also need charging in daytime. However, the number of such buses is usually rather few and hence the charging operation is much less challenging.*

Remark 2. *For our case studies, we will use two data sets from Svealandstrafiken and Transdev, respectively. However, throughout the discussing of system model and mathematical optimization,* *we use the input setting from Svealandstrafiken, because this case has more restrictions and constraints.*

2.2 Buses

The basic problem input consists in a given set of electrical buses; we use $\mathcal{B}$ to denote its index set. Each bus has an initial state of charge (IRoC) upon arrival to the depot, and a target state of charge (TRoC) before departure for service. The departure time corresponds to the deadline of reaching the TRoC. The buses vary in arrival time. As stated earlier, time is sliced, hence in the problem the arrival and departure times are converted to time slot indices. For bus $b \in \mathcal{B}$, these are denoted by a_b and d_b, respectively.

Figure 1 shows an example of the arrival and departure times of a bus and how these are mapped to time slots. In the example, the time period for scheduling is from 18.00 to 7.00 next morning. Note that the time slots are set to be on the conservative side to ensure the schedule is indeed time-feasible. The bus in question arrives at 18:12 and departs at 7:02. Thus the arrival time is in the first time slot (with a slot length of 15 minutes). However, if optimization selects (the entire) time slot one for charging, in reality the bus will be under-charged. Therefore, time slot two is considered the arrival time slot. For the same reason, the time slot for departure is moved earlier, from time slot 53 to time slot 52.

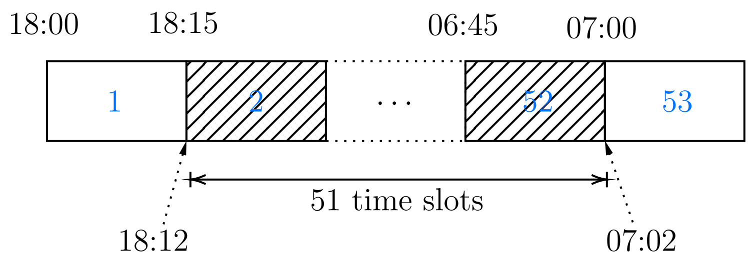

Figure 1: An example of arrival and departure time of a bus and the corresponding time slots.

An example of a 17-bus instance is given in Table 1. For each bus, the table shows the ending time of the service; this is the arrival time to the depot. The start time is the time for departure. Charging may take place between the arrival and departure times, and the corresponding time slots are shown within parentheses. The ISoC and TSoC are specified in percentage with respect to full charge. The distance is the total distance of the routes to be taken by the bus.

Assuming C is the average energy consumption per km, R is the energy consumption rate per km, and C is the battery capacity, the TSoC is calculated as follows. Here, $\eta > 1$ is a parameter to provide some margin. In our case study for Svealandstrafiken, $\eta = 1.2$.

$$\text{TSoC} = \frac{\text{Distance} \times R}{C} \times \eta. \tag{1}$$

The values of battery capacity C and unit energy consumption are given in Table 2. These values again are specific for the electric buses of Svealandstrafiken.

Table 1: Example scenario of 17 buses.

No.	**Start** (d_b)	**End** (a_b)	**Distance** (km)	**TSoC**	**ISoC**
1	07:15 (53)	18:29 (3)	178.62	68.41%	11.40%
2	07:02 (52)	18:12 (2)	182.96	70.07%	11.68%
3	06:01 (48)	18:42 (4)	188.28	72.11%	12.02%
4	05:54 (47)	18:06 (2)	196.96	75.43%	12.57%
5	06:17 (49)	18:20 (3)	196.96	75.43%	12.57%
6	06:07 (48)	18:20 (3)	197.05	75.47%	12.58%
7	05:56 (47)	18:10 (2)	197.41	75.60%	12.60%
8	05:33 (46)	18:26 (3)	199.66	76.47%	12.74%
9	06:03 (48)	18:56 (5)	199.66	76.47%	12.74%
10	05:51 (47)	18:09 (2)	201.65	77.23%	12.87%
11	05:56 (47)	18:17 (2)	203.23	77.83%	12.97%
12	06:16 (49)	18:46 (5)	203.23	77.83%	12.97%
13	04:48 (43)	18:27 (3)	206.70	79.16%	13.19%
14	04:54 (43)	18:26 (3)	225.47	86.35%	14.39%
15	04:48 (43)	18:59 (5)	225.63	86.41%	14.40%
16	06:06 (48)	20:00 (9)	228.55	87.53%	14.59%
17	06:36 (49)	20:30 (11)	228.55	87.53%	14.59%
		Average:	203.56		

Table 2: Battery and energy consumptions parameters.

Parameter	**Value**
The battery capacity (C)	564 (kWh)
Energy consumption rate (R)	1.8 (kWh/km)

2.3 Depot and Charging Station

In an ideal scenario, the depot is constructed such that all electric buses can be parked without any order restriction, and they can be charged fully independently except for the total power limit at the depot. Some depots however have some restrictions, such as the one of Svealandstrafiken to be presented below.

The depot layout is shown in Figure 2. There are three parking lines for electric buses, with six, six, and five parking lots, respectively. The parking lots are indexed from one to 17. The first two lines are organized in form of five pairs of parking lots. Each pair of buses, such as buses one and seven, share one charging station.

As an additional restriction, the charging rates are discrete. In the scenario of Svealandstrafiken, the possible rates are $\{0, 50, 150\}$ kW. However, the maximum output of a charging station is 150 kW. Hence, for a pair of buses sharing a station, the following combinations of charging power are possible: $\{0, 0\}$, $\{0, 50\}$, $\{50, 0\}$, $\{50, 50\}$, $\{0, 150\}$, and $\{150, 0\}$. Thus, if both buses are

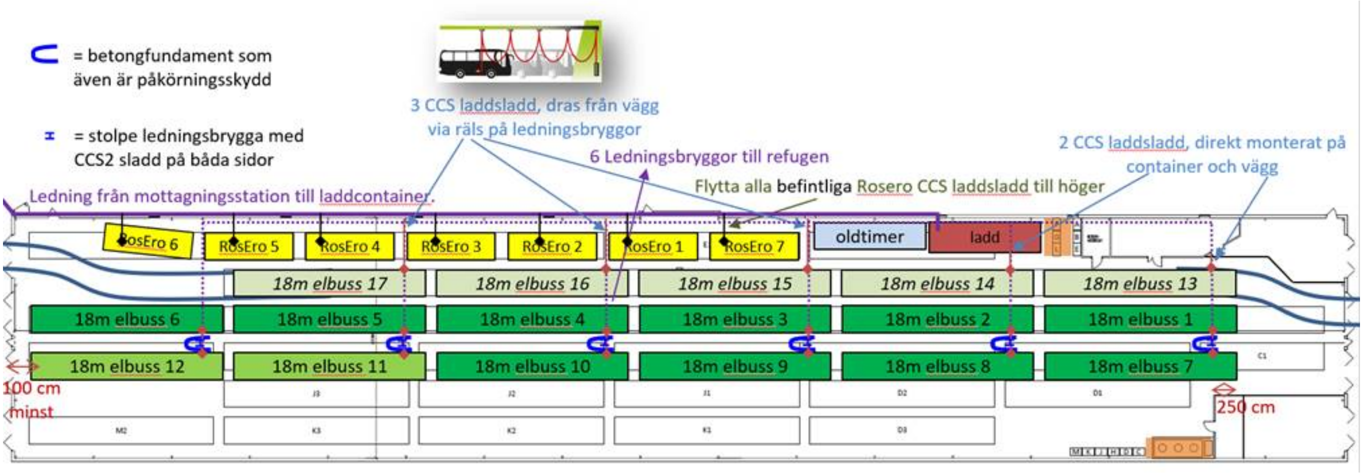

Figure 2: The layout of a depot of Svealandstrafiken,

charged, the only possibility is to use 50 kW.

The depot has a total power limit, denoted by P_{dep}. The value may vary in time. For the depot illustrated in Figure 2, P_{dep} is 700 kVA from 6:00 to 22:00, and 1400 kVA from 22:00 to 6:00. In addition, there is a base load P_{base} of 500 kW; this is the constant power consumption for other functions of the depot. Figure 3 shows the available power for charging the electrical buses over time. One can note that until 22:00, the power available for charging is limited to 200 kW only, meaning that at most four buses can be charged simultaneously with positive rate. As we will see, as a result, to reach TRoC for all buses before their respective departure times is quite challenging.

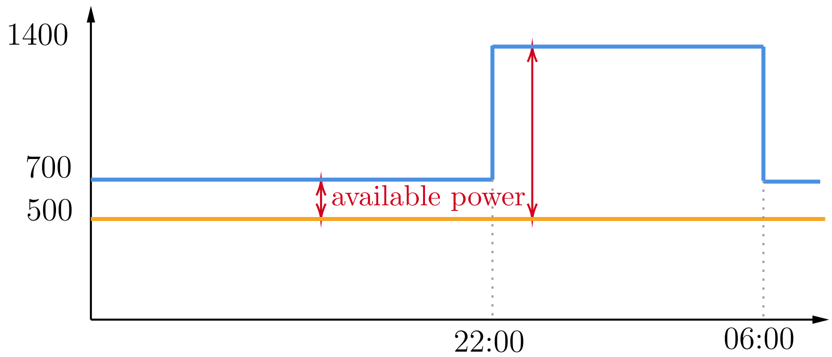

Figure 3: The power limit (blue line) and the base load (orange line) over time, with the difference being the amount of available power for bus charging.

2.4 Charging Rate

A simple model for charging rate is to assume the SoC increases linearly in the charging power. In a more realistic model, however, the charging rate is is non-linear. In particular, the rate declines somewhat for higher SoC. Denote the charging power for bus b by P_b. The charging rate, denoted by P'_b, is scaled by parameter α as follows.

$$P_b' = \alpha P_b, \tag{2}$$

The value of α depends on the current SoC. An example of values is shown in Table 3. In this cases, there are three intervals with decreasing values of α.

Table 3: An example of non-linear charging coefficient.

Battery level (SoC)	$\boldsymbol{\alpha}$
0 – 70%	1.0
70 – 80%	0.8
80 – 100%	0.6

A non-linear charging rate potentially introduces more complexity in problem-solving. However, for the charging power structure such as the one specified above, we can in fact eliminate the non-linearity via pre-processing. Note that the charging power of 150 kw is a multiple of 50 kw. Thus, given the IRoC and TRoC, and the non-linear charging rate function, we can perform pre-processing to obtain the number of time slots needed for each individual bus if it is charged with 50 kw power. Denote this number by r_b for bus b. If the bus is charged with 150 kw in some time slot, then that time slot in effect completes three units of r_b. Therefore, instead of specifying IRoC and TRoC, the problem can be equivalently be stated using integer parameter r_b for each $b \in \mathcal{B}$.

2.5 Optimization Objectives

An optimization problem is always defined with some objective function to be maximized or minimized. For electric bus charging, several objectives, as discussed below, may become relevance.

- One common objective in optimal scheduling is to accomplish all tasks as early as possible. In optimization, this is also known as the makespan [6]. Minimizing the makespan means to make a schedule with the earliest possible time point by which all buses reach their respective TRoC levels. (Note that most buses will have their charging complete before this time point.)

- For charging power, another objective is load balancing, namely to find a schedule such that the total charging power is evenly distributed. A special case is to minimize the maximum charging power over time. Solving this min-max problem will let us know the total power that the bus operator would need, in the worse case, from the grid.

- Another interesting objective is to minimize the electricity cost (if it varies over the scheduling period), or maximize the potential profit via bidding in electrical markets.

- Finally, it is of interest to electrify as many buses as possible. For a given scenario, we can electrify as many buses as the number of parking lots. However, even in this case, the objective remains interesting (assuming more parking lots are available), because it can tell the potential of expansion.

3 Optimization Approaches

3.1 Problem Complexity

The complexity of a combinatorial optimization is largely problem-dependent. Some combinatorial optimization problems, such as the shortest path problem [7], admits algorithms that guarantees global optimum time efficiently. Other problems, for example the traveling salesman problem [8], are known to be hard in terms of approaching global optimum time efficiently when the instance scale grows.

In combinatorial optimization, a problem is considered easy, if there exists some algorithm that finds the optimum solution in an amount of time that is bounded by a polynomial in input size. The notion of NP-hardness is generally used to classify difficult problems. For an NP-hard problem, there is no known polynomial-time and exact algorithm. Thus, in the worse case, an exact algorithm is facing the entire solution space (i.e., all combinations of discrete decisions) in problem solving. Electric bus scheduling (and in fact most scheduling optimization problems) is NP-hard [9], in terms of complexity.

Remark 3. *A common misconception is that that an NP-hard problem cannot be solved to optimum. This is not true. The notion of NP-hardness implies, in theory, that if we insist on finding the global optimum, then there is no existing algorithm that can guarantee to do so and at the same time scale well. In practice, NP-hard problems can often be solved fast to optimum, in particular if the instance size is not very large.*

3.2 Solution Algorithm: Preliminaries

Generally speaking, for an NP-hard combinatorial optimization problem, we have three types of solution algorithms.

- *Exact algorithm* that guarantees optimum. To this end, one tool at hand is integer linear programming (ILP) [10], where the discrete choices are represented using integer (with binary as a special case) variables. Solving an ILP leads to the use of the branch-and-bound (B&B) algorithm. In short, the algorithm solves a relaxed problem by removing the integrality requirement (thus giving a bound to the optimal value), and explores the solution space by selecting variables and setting their (integer) values. The algorithm also uses cutting planes to make the relaxed problem closer to the original one. A portion of the solution space will be cut off, if via the bound one can tell that the optimum is not located in that part of the space. To be practical, the algorithm would need an ILP that is an effective representation of the optimization problem.

- *ILP-driven heuristics.* The use of ILP does not necessarily imply an exact algorithm. In fact, we can use ILP to construct some heuristic solution. The simplest idea would be to perform some type of rounding to the solution obtained from the relaxed problem. Such an approach performs well, again if the ILP model is an effective problem representation.

- *Heuristics* that are designed to run fast but do not leave guarantee of solution quality with certainty. There is an array of notions for search heuristics, such as local search [11], simulated annealing [12], tabu search [13], evolutionary algorithms [14], etc. Generally speaking, such algorithms explore part of the solution space, with some mechanism (e.g., randomness and memory) for diversifying the search, and often also some mechanism (e.g., selection based on solution fitness) for intensifying the search.

For an optimization problem, if we compute a solution that may not be the optimal one, it is highly desirable to know potentially how much room is left for improvement, i.e., some bound as an estimate of how far the solution can be from the optimum. Heuristics such as local search cannot provide such a bound. For this reason, we chose to use ILP and an ILP-based heuristic.

3.3 Integer Linear Programming

Combinatorial optimization problems can be modeled using ILP. There are a couple of advantages of an ILP approach. First, using ILP, we can compute the exact, global optimum, rather than a local optimum (that is found using heuristic search algorithms). Second, ILPs are well studied and there are off-the-shelf optimization tools (e.g., [15, 16]), some being open source (e.g., [17]). In addition, ILP models are often easy to adapt to variations of a problem (i.e., different objectives), whereas for a pure heuristic designed for a specific problem setting, adding a constraint or changing the objective function may render the heuristic ineffective or even infeasible and hence extra development effort is required for adaption.

Taking an ILP optimization approach, the challenge is to formulate the problem mathematically with linear functions. Here, we demonstrate how this can be done for optimal charging schedule for two of the objective functions discussed earlier, with the restrictions and constraints that are present for the Svealandstrafiken's depot.

Let us consider the minimum makespan problem. We begin by defining the optimization variables. They consist of three sets of binary variables, and one single continuous variable. Also, we use $\mathcal{C}$ to denote the set containing pairs of buses sharing charging stations.

$$
\begin{aligned}
x_{bt}^{50} = \quad & \begin{cases} 1 & \text{if bus } b \text{ is charged with 50 kw power in time slot } t, \\ 0 & \text{otherwise.} \end{cases} \\
x_{bt}^{150} = \quad & \begin{cases} 1 & \text{if bus } b \text{ is charged with 150 kw power in time slot } t, \\ 0 & \text{otherwise.} \end{cases} \\
y_t = \quad & \begin{cases} 1 & \text{if any bus is being charged in time slot } y, \\ 0 & \text{otherwise.} \end{cases} \\
q = \quad & \text{Makespan}
\end{aligned}
$$

$$\min_{\boldsymbol{x},\boldsymbol{y}\in\{0,1\},q\geq 0} \quad q \tag{3a}$$

$$\text{s.t.} \quad x_{bt}^{50} + x_{bt}^{150} \leq 1, \forall b \in \mathcal{B}, t \in \mathcal{T} \tag{3b}$$

$$\sum_{b\in\mathcal{B}}(50x_{bt}^{50}+150x_{bt}^{150})\leq\hat{P}_t,\forall t\in\mathcal{T} \tag{3c}$$

$$\sum_{t=a_b}^{d_b}x_{bt}^{50}+3x_{bt}^{150}\geq r_b,\forall b\in\mathcal{B} \tag{3d}$$

$$50x_{bt}^{50}+150x_{bt}^{150}+50x_{\bar{b}t}^{50}+150x_{\bar{b}t}^{50}\leq 150,\forall(b,\bar{b})\in\mathcal{C} \tag{3e}$$

$$x_{bt}^{50}+x_{bt}^{150}\leq y_t,\forall t\in\mathcal{T},b\in\mathcal{B} \tag{3f}$$

$$q\geq ty_t,\forall t\in\mathcal{T} \tag{3g}$$

The meaning of the constraints are as follows.

- By equation (3b), a bus cannot be charged with both power levels in any time slot.
- In (3c), the sum is taken over all buses (for each time slot), and this sum is the total power consumption that is bounded by the available power $\hat{P}_t$.
- By constraint (3d), each bus has to be charged with a sufficient number of energy, in unit defined by charging with 50 kw for one time slot, in order to reach the TRoC.
- For the next set of constraints, (3e), there is one defined for each time slot and pair of bus sharing a charging station, ensuring that these two buses use up to 150 kw power.
- By constraint (3f), if any bus is being charged (i.e., the left-hand side is one), y_t is one. Finally, by (3g), if y_t is one, then q, the makespan, is at least t.

As can be seen, all functions defining the constraints of the variables are linear. At optimum, the schedule, represented by the $\boldsymbol{x}$-variables, is the one with minimum possible makespan. Next, we show how the ILP can be adapted to minimize the maximum power over the time slots. To this end, we use a continuous variable u for the maximum power consumption over all time slots.

$$\min_{\boldsymbol{x}\in\{0,1\},u\geq 0}\quad u \tag{4a}$$

$$\text{s.t.}\quad (3\text{b}),(3\text{d})-(3\text{e}) \tag{4b}$$

$$\sum_{b\in\mathcal{B}}50x_{bt}^{50}+150x_{bt}^{150}\leq u,\forall t\in\mathcal{T} \tag{4c}$$

That u indeed takes the value of largest power consumption over time slots is represented by (4c). Note that we do not need to include (3d), because u is minimized. In fact, if the minimum u is above $\hat{P}$, it is not possible to complete the charging in time with the power available at the depot. Hence this ILP can be used to check if there is a feasible schedule at all or not, and, in the latter case, provide information of how much additional power has to be made available.

3.4 Column Generation

3.4.1 Preliminaries

Since an ILP may or may not scale well in problem size, we design another approach that is based on solving pure linear programming problems. This is done by considering a problem reformulation, utilizing the fact that a combinatorial optimization problem typically admits more than one mathematical formulation.

From the problem description, for one bus, the solution consists in a sequence of decisions over the time slots. For each time slot, the decision can be either charging or not charging, and in the former case at what power level. An example of a sequence (or charging schedule) of an individual bus is illustrated in Figure 4. For each time slot, the schedule tells the charging power.

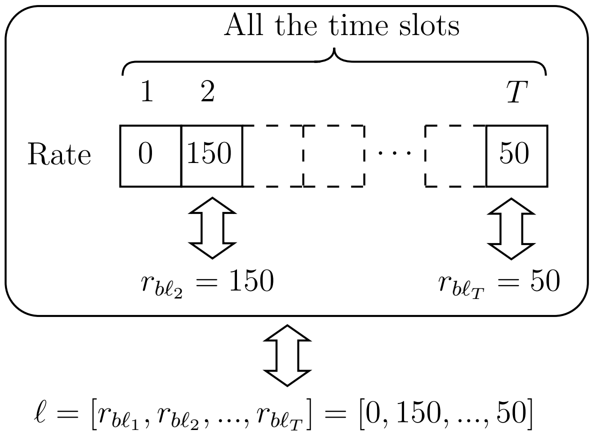

Figure 4: An example showing how a column ℓ of bus b represents a specific charging schedule.

The number of sequences of a bus is finite (though exponential in the number of time slots). For each individual bus, the sequence is a feasible one (which can be modeled as a sequence with zero cost), if and only if its charging decisions meet the TSoC. A charging schedule, in fact, amounts to selecting a sequence of each bus, such that they together satisfy the common restrictions, in particular the total power available and that of a charging station shared by two buses.

Taking the above view of problem solution, we define set $\mathcal{L}_b$ as the index set of all sequences of bus b. Note that this is now part of input. We define the following new parameters and variables.

- $\Delta_{b\ell}$: cost associated with sequence ℓ. This cost is zero if the bus is charged to its TSoC, and positive otherwise. Specifically, the value of $\Delta_{b\ell}$ is the TSoC minus the reached SoC by using sequence ℓ.
- $r_{b\ell t}$: is the charging rate in schedule ℓ used for recharging bus b in time slot t. We know that the charging rate can only be either 0, or 50, or 150, for the case of Svealandstrafiken.

$$\chi_{b\ell} = \begin{cases} 1 & \text{if bus } b \text{ uses sequence (charging schedule) } \ell \in \mathcal{L}_b, \\ 0 & \text{otherwise.} \end{cases}$$

3.4.2 Reformulation for makespan

Based on the definitions above, the reformulation is as follows.

$$
\begin{aligned}
\min_{\boldsymbol{\chi}\in\{0,1\}} \quad & \sum_{b\in\mathcal{B}}\sum_{\ell\in\mathcal{B}} \Delta_{b\ell}\chi_{b\ell} && \text{(5a)}\\
\text{s.t.} \quad & \sum_{\ell\in\mathcal{L}_b} \chi_{b\ell} = 1, \forall b \in \mathcal{B} && \text{(5b)}\\
& \sum_{\substack{\ell\in\mathcal{L}_b:\\ r_{b\ell t}=150}} \chi_{b\ell} + \sum_{\substack{\ell\in\mathcal{L}_{\bar{b}}:\\ r_{\bar{b}\ell t}=50 \text{ or } 150}} \chi_{\bar{b}\ell} \leq 1, \forall t \in \mathcal{T}, \forall (b,\bar{b}) \in \mathcal{C} && \text{(5c)}\\
& \sum_{\substack{\ell\in\mathcal{L}_b:\\ r_{b\ell t}=50}} \chi_{b\ell} + \sum_{\substack{\ell\in\mathcal{L}_{\bar{b}}:\\ r_{\bar{b}\ell t}=150}} \chi_{\bar{b}\ell} \leq 1, \forall t \in \mathcal{T}, \forall (b,\bar{b}) \in \mathcal{C} && \text{(5d)}\\
& \sum_{b\in\mathcal{B}}\sum_{\ell\in\mathcal{L}_b} r_{b\ell t}\chi_{b\ell} \leq \hat{P}_t, \forall t \in \mathcal{T} && \text{(5e)}
\end{aligned}
$$

- By constraint (5b), for each bus, exactly one sequence (charging schedule) of index set $\ell \in \mathcal{L}_b$ is to be selected.
- The next two sets of constraints apply to pairs of buses sharing a charging station. For each of such pairs, if one bus is charged at power level of 150 kW in a time slot (i.e., any χ-variables for such choices is one), then the other bus may not be charged at all.

For the above reformulation, there is an exponential number of variables $\boldsymbol{\chi}$, each corresponding to a charging schedule of one bus. Since the coefficient vector of a variable in an optimization formulation is a column in the constraint matrix, what we have is an exponential number of columns. However, at optimum, only one column/variable is selected per bus. Thus, we consider an algorithm that generates columns that are promising for defining the optimum. This is called the column generation algorithm. The algorithm starts with a very few number of columns and then iteratively adds the ones that are likely to improve the solution. To this end, we consider the continuous approximation (known as the linear programming relaxation, or LP relaxation in short) of the reformulation, such that the generation of columns can be down systematically using LP duality. In summary, the algorithm carries out the following steps.

- Consider a restricted subset of $\mathcal{L}_b, b \in \mathcal{B}$.
- Solve the LP relaxation, that is, binary variables $\boldsymbol{\chi}$ is relaxed to be continuous.
- For each bus $b \in \mathcal{B}$, identify if adding a variable (and its coefficient column) will yield improvement. This can be done by solving a shortest path problem per bus, formulated via LP duality. (Note, since this step decomposes by bus, it can be implemented to run in parallel).
- Add the identified variables and repeat.

Upon termination, the optimal objective value is a lower bound on the makespan. That is, the integer optimum will not have a lower makespan than that reported by the column generation algorithm.

The solution at this stage may be fractional-valued. To obtain a feasible integer solution, we propose a tailored type of rounding. Namely, for a bus, the most likely optimal charging decisions over the time slots are identified, based on the variable values. Next, for a bus b, one time slot is chosen and its charging decision (i.e., no charging, charging at 50 kW, or changing at 150 kW) is set. In effect, we have reduced the space of the set $\mathcal{L}_b$. We then apply column generation again, and repeat the process until an integer solution is reached.

For the other objective of power minimization, it is rather straightforward to adapt (5). The corresponding column generation algorithm works in a similar way.

4 Case Study: Svealandstrafiken

The data of Svealandstrafiken consists in 17 buses. The buses are delivered by March 2023 [18]. We use the depot and bus data discussed earlier for our case study of Svealandstrafiken. First, we perform charging scheduling optimization for the 17 buses (see Table 1), and then perform some what-if analysis (e.g., what happens to the optimization result if the total power available becomes larger than today). For the case study, the ILP approach is used and hence the answers we obtain are globally optimal.

4.1 Result of Minimum Makespan

Figure 5 illustrates the result of the minimum makespan schedule for the 17 buses. In the figure, the buses are sorted in ascending order of distance of its service to be performed the next day. Because of this sorting, the time margin between reaching the TRoC and the deadline (i.e., departure time) becomes smaller with respect to the x-axis. For each bus, the dot shows the time of reaching the TRoC, and the arc shows the remaining time before departure.

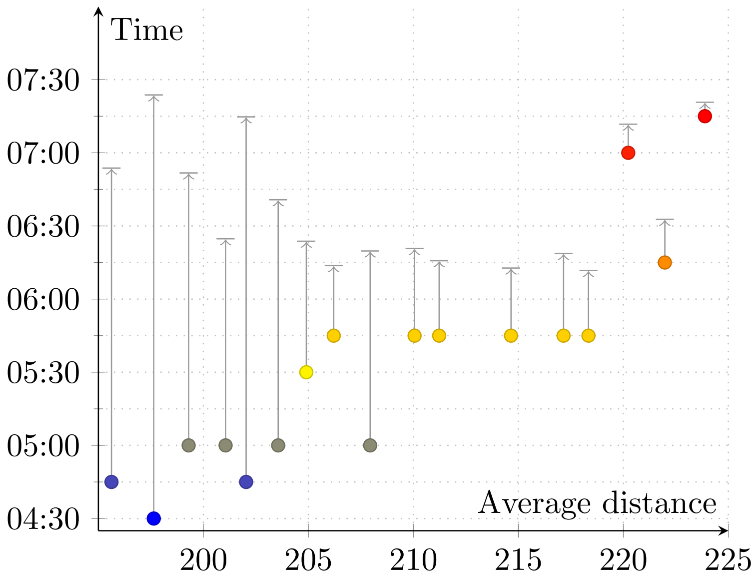

Figure 5: Result of minimizing makespan for 17 buses. The dots represent the completion times, and the arrows represent the margins.

As can be seen from the figure, the minimum makespan is 7:15 in the morning. There is one bus that finishes charging at this time point. There are two other buses reaching their TRoC at 7:00 and 6:15, respectively. For all the other 14 buses, charging become complete no later than 5:45. A similar observation is made for the time margin, namely, most buses have a large amount of time margin before departure, though for a few buses the time is tight.

In terms of the completion time, one can clearly see that the distance (and hence the amount of SoC needed) plays a major role, namely longer distance directly translates into later completion time. The time margin is directly related to distance as well.

4.2 One More Charger: From 17 to 18

Let us assume that one more charger is available and it is exclusive (i.e., not of shared type). The result is shown in Figure 6. In the figure, the triangle marker above a dot on the same vertical line is the new completion time of the bus (if 18 buses need to be charged).

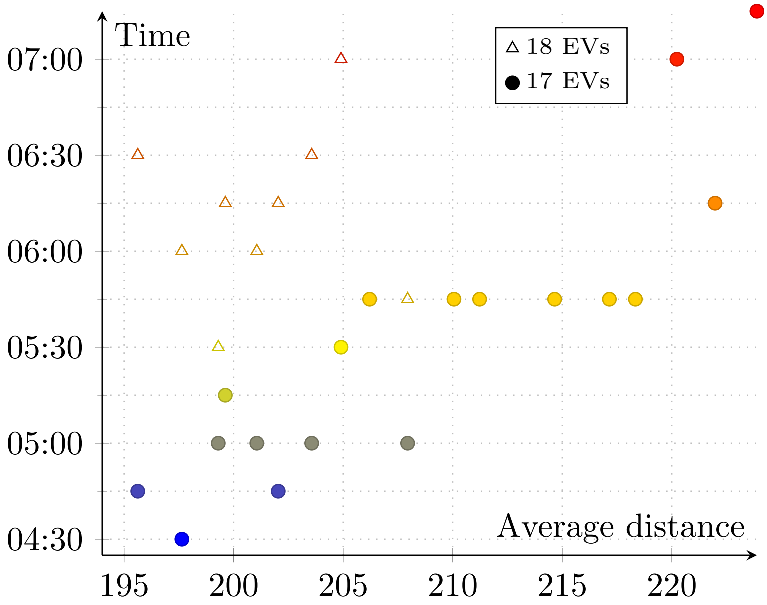

Figure 6: Charging 18 buses.

From the figure, one can see that, in order to support 18 buses, the completion time of many buses becomes considerably postponed. Thus, the result indicates that, given the current power limit, there is not much room left of electrifying more buses.

4.3 If More Power Is available

From the previous result, one can ask the question: What happens if the power limit is higher) We consider two scenarios to this end.

- **Scenario 1**: There is an additional power of 150 kVA available from 22:00 to 06:00. Figure 7 shows the available power for charging the electrical buses over time in this scenario.
- **Scenario 2**: There is an additional power of 150 kVA available all the time. Figure 8 shows this scenario.

For the 17-bus instance shown in Table 1, we study the benefits of having more power available. The results are given in Table 4. As we can observe, if we increase the power limit by 10% during the "busy hours of charging" (Scenario 1), all the 17 buses have charging completed by 3:45; this is significantly earlier than before. If the power limit is higher also for the other hours (Scenario 2), we gain another 30 minutes. As a conclusion, having more power available during the crucial hours has a clear benefit.

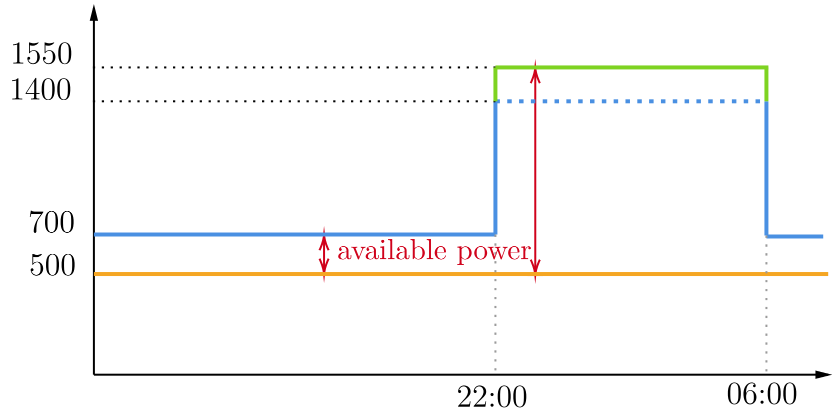

Figure 7: The green line represents the new power limit under Scenario 1.

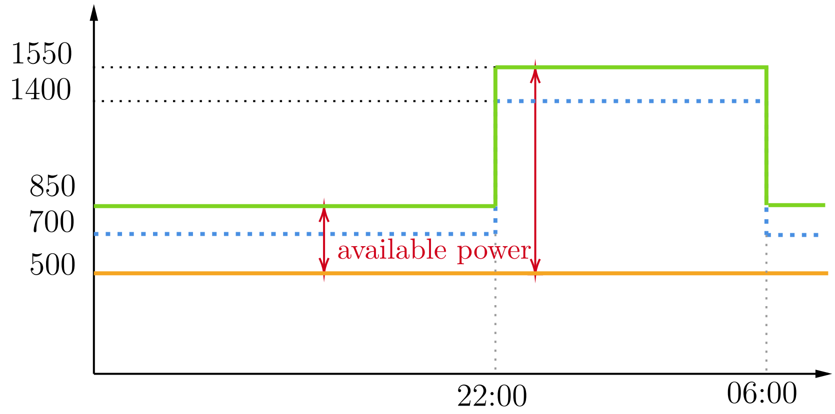

Figure 8: The green line represents the new power limit under Scenario 2.

Table 4: The results for the 17-bus instance assuming more power available.

	Completion time (makespan)
Original	5:00
Scenario 1	3:45
Scenario 2	3:15

For the two scenarios, we next consider many buses can be electrified (assuming that the depot is expended with more parking lots and charges). The results are shown in Table 5. From this table, one can see that it is feasible to electrify four additional buses; an increase of being close to 25%. To achieve this, it is sufficient to having more power during the crucial hours of charging, even though having more power throughout the night leads to smaller makespan.

4.4 Removing Battery Margin

All the previous results assume that there is a 20% margin (backup energy) in the batter level when a bus finishes its service the next day. Suppose we set ISoC to be zero, i.e., we remove this

Table 5: The results of electrifying more buses assuming more power available.

	Completion time	
Buses	Scenario 1	Scenario 2
17	03:45	03:15
18	04:45	04:45
19	05:15	05:00
20	05:30	05:15
21	05:45	05:15
22	No solution	No solution

margin. The corresponding TRoC values are adjusted accordingly, and the specific values are given in Table 6.

Table 6: Example scenario of 17 buses with revised TSoC and ISoC.

No.	**Start** (d_b)	**End** (a_b)	**Distance** (km)	**TSoC**	**ISoC**
1	07:15 (53)	18:29 (3)	178.62	57.01%	0
2	07:02 (52)	18:12 (2)	182.96	58.39%	0
3	06:01 (48)	18:42 (4)	188.28	60.09%	0
4	05:54 (47)	18:06 (2)	196.96	62.86%	0
5	06:17 (49)	18:20 (3)	196.96	62.86%	0
6	06:07 (48)	18:20 (3)	197.05	62.89%	0
7	05:56 (47)	18:10 (2)	197.41	63.00%	0
8	05:33 (46)	18:26 (3)	199.66	63.72%	0
9	06:03 (48)	18:56 (5)	199.66	63.72%	0
10	05:51 (47)	18:09 (2)	201.65	64.36%	0
11	05:56 (47)	18:17 (2)	203.23	64.86%	0
12	06:16 (49)	18:46 (5)	203.23	64.86%	0
13	04:48 (43)	18:27 (3)	206.70	65.97%	0
14	04:54 (43)	18:26 (3)	225.47	71.96%	0
15	04:48 (43)	18:59 (5)	225.63	72.01%	0
16	06:06 (48)	20:00 (9)	228.55	72.94%	0
17	06:36 (49)	20:30 (11)	228.55	72.94%	0
		Average:	203.56		

The results show that this revised instance has a makespan of 04:45; this is only 15 minutes better than the case of 20% backup energy. Interestingly, in terms of the number of buses, the maximum remains 18. Hence having no margin of battery energy does little help.

4.5 Summary

From the results presented, for Svealandstrafiken, the data represents is a rather saturated case. That is, with the current capacity and power limit, having 17 buses is close to the capacity. However, if more power is made available, even if the increase is moderate amount there is a clear improvement.

5 Case Study: Transdev

The case parameters are listed in Table 7, and the bus information of Transdev is in Table 8 that is selected from the original bus information (see the appendix Table 9). We would like to minimize the maximum power over the time slots for this case.

Table 7: Case parameters.

Parameter	**Value**
The number of buses	16
Charging power	60 (kW)
The battery capacity	396 (kWh)

Table 8: Buses information.

Block	**Start**	**End**	**RSoC (%)**	**Block**	**Start**	**End**	**RSoC (%)**
13101	3:59	21:25	52.38	13212	5:20	24:53	31.51
13102	4:19	25:33	30.21	13213	5:30	8:08	16.40
13103	4:27	18:57	38.79	13213	9:19	17:11	39.02
13104	4:39	23:30	50.94	13214	5:39	24:20	16.06
13105	4:47	26:03	21.09	13215	5:45	24:01	34.12
13106	4:59	19:24	28.17	13216	5:57	18:10	41.01
13109	5:43	18:38	37.16	13217	6:12	8:20	31.00
13110	5:51	25:49	34.20	13217	12:38	16:25	25.30
13111	6:03	18:09	31.28	13217	16:38	26:00	9.85
13112	6:09	25:22	72.73	13218	6:15	8:43	25.78
13113	6:19	25:03	32.51	13218	15:38	25:30	52.38
13114	6:21	23:50	47.20	13219	6:26	8:53	34.36
13115	6:38	23:54	46.62	13219	15:19	17:54	40.97
13201	4:30	18:59	40.95	13220	6:36	9:07	23.33
13202	4:31	24:20	43.98	13220	15:23	17:55	37.23
13203	4:34	8:28	52.64	13221	6:41	9:13	40.38
13203	15:53	25:57	27.35	13221	15:08	25:54	26.50
13204	4:44	17:45	28.07	13222	6:56	9:21	26.46
13205	4:52	25:41	45.24	13222	12:55	18:15	46.25
13206	5:00	24:33	28.26	13223	6:59	9:01	26.68
13207	5:03	17:40	44.81	13223	10:59	25:52	20.57
13208	5:08	19:09	41.23	13224	7:06	9:35	26.38
13209	5:14	18:45	45.99	13224	14:59	25:44	31.65
13210	5:15	19:13	23.26	13225	7:11	18:05	33.26
13211	5:16	9:05	35.33	13226	7:26	17:45	49.99
13211	14:31	19:05	42.01				

5.1 Integer Linear Programming

The ILP formulation can be given by

$$
\begin{aligned}
\min_{\boldsymbol{x}\in\{0,1\},u\geq 0} \quad & u && \text{(6a)}\\
\text{s.t.} \quad & \sum_{t=a_b}^{d_b} x_{bt} \geq r_b, \forall b \in \mathcal{B} && \text{(6b)}\\
& \sum_{b\in\mathcal{B}} 60x_{bt} \leq u, \forall t \in \mathcal{T} && \text{(6c)}
\end{aligned}
$$

$$
\begin{aligned}
x_{bt} = & \begin{cases} 1 & \text{if bus } b \text{ is charged (with 60 kW power) in time slot } t, \\ 0 & \text{otherwise.} \end{cases}\\
r_b = & \text{ The RoC of bus } b \text{ divided by 60}\\
u = & \text{ The maximum power over the time slots}
\end{aligned}
$$

- By constraint (6b), each bus has to be charged with a sufficient number of energy (i.e., RSoC), in unit defined by charging with 60 kw for one time slot, in order to reach 100%.
- By constraint (6b), that u takes the value of largest power consumption over time slots.

5.2 Reformulation for column generation

$$
\begin{aligned}
\min_{\boldsymbol{\chi}\in\{0,1\},u>0} \quad & u_{\max} && \text{(7a)}\\
\text{s.t.} \quad & \sum_{\ell\in\mathcal{L}_b} \chi_{b\ell} = 1, \forall b \in \mathcal{B} && \text{(7b)}\\
& \sum_{b\in\mathcal{B}} \sum_{\ell\in\mathcal{L}_b} u_{b\ell t}\chi_{b\ell} \leq u_t, \forall t \in \mathcal{T} && \text{(7c)}\\
& u_t \leq u_{\max}, \forall t \in \mathcal{T} && \text{(7d)}
\end{aligned}
$$

$$
\begin{aligned}
\chi_{b\ell} = & \begin{cases} 1 & \text{if bus } b \text{ uses sequence (charging schedule) } \ell \in \mathcal{L}_b, \\ 0 & \text{otherwise.} \end{cases}\\
u_{b\ell t} = & \begin{cases} 1 & \text{if bus } b \text{ is charged in time slot } t \text{ in sequence (charging schedule) } \ell \in \mathcal{L}_b, \\ 0 & \text{otherwise.} \end{cases}\\
u_t = & \text{ The maximum power in time slot } t\\
u_{\max} = & \text{ The maximum power over the time slots}
\end{aligned}
$$

- By constraint (7b), for each bus, exactly one sequence (charging schedule) of index set $\ell \in \mathcal{L}_b$ is to be selected.

- The left-hand side of constraint (7c) is the sum of total power in time slot t from the selected sequences for the total buses, and it should be less than the maximum power in time slot t, i.e., u_t.
- By constraint (7d) impose that $u_{\max}$ is greater than all the u_t for any time slot. In addition, our objective function is to minimize $u_{\max}$, so we can obtain the minimized maximum power over the time slots after we solve the model.

5.3 The result of Transdev

We present the night part (18:30 - 24:00) of the result in Table 9, where yellow and blue blocks represent that the bus is in service and being charged in those time slots, respectively. The result shows that a maximum of six buses are charged simultaneously, i.e., the minimized maximum power over the time slots is $6 \times 60 = 360$ kW.

Figure 9: The night part of schedule for Transdev

6 Conclusions

References

[1] A.-C. Lundström, M. N. Holmström, E.Torstensson, and M. Eriksson, "Electric buses for Swedish public transport services," Trafikverket, Sweden, Tech. Rep., 2019.

[2] "Varannan ny buss en elbuss," https://www.bussmagasinet.se/2022/08/varannan-ny-buss-en-elbuss/, 2022.

[3] "Elbussmarknaden ökade med 26 procent under 2022," https://www.energinyheter.se/20230216/28548/elbussmarknaden-okade-med-26-procent-under-2022, 2023.

[4] "Elektrifierade persontransporter - en del av den smarta staden," https://www.vinnova.se/p/elektrifierade-persontransporter---en-del-av-den-smarta-staden2/, 2021.

[5] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 6th ed. Springer, 2018.

[6] H. Xiong, S. Shi, D. Ren, and J. Hu, "A survey of job shop scheduling problem: The types and models," *Computers & Operations Research*, vol. 142, 2022, https://doi.org/10.1016/j.cor.2022.105731.

[7] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[8] D. L. Applegate, R. E. Bixby, V. Chv'atal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007.

[9] O. Sassi and A. Oulamara, "Electric vehicle scheduling and optimal charging problem: Complexity, exact and heuristic approaches," *International Journal of Production Research*, vol. 55, pp. 519–535, 2014.

[10] L. Wolsey, *Integer Programming*, 2nd ed. Wiley, 2020.

[11] W. Michiels, J. Korst, and E. Aarts, *Theoretical Aspects of Local Search*. Springer, 2007.

[12] P. Salamon, P. Sibani, and R. Frost, *Facts, Conjectures, and Improvements for Simulated Annealing*. SIAM, 2002.

[13] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1997.

[14] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer, 2010.

[15] "IBM ILOG CPLEX optimixer," https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer, 2023.

[16] "GUROBI optimizer," https://www.gurobi.com/solutions/gurobi-optimizer/, 2023.

[17] "HiGHs - high performance software for linear optimization," https://highs.dev/, 2023.

[18] "Samtliga 17 elbussar i Västerås är nu redo för trafik," https://www.svealandstrafiken.se/se/nyheter/#/pressreleases/samtliga-17-elbussar-i-vaesteraas-aer-nu-redo-foer-trafik-3239366, 2023.

Table 9: Appendix: The original buses information of Trasndev.

Block	Start	End	Distance	Total kWh	Identifier	Start1	End1	SoC after
13101	3:59	21:25	299.947	630.120361	Prep-out	3:59	4:05	100
13101	3:59	21:25	299.947	630.120361	Prep-in	21:22	21:25	47.617878
13102	4:19	25:33	386.779	812.3927	Prep-out	4:19	4:25	100
13102	4:19	25:33	386.779	812.3927	Prep-in	25:30	25:33	69.7873383
13103	4:27	18:57	268.296	563.439941	Prep-out	4:27	4:33	100
13103	4:27	18:57	268.296	563.439941	Prep-in	18:54	18:57	61.2085571
13104	4:39	23:30	331.712	696.860291	Prep-out	4:39	4:45	100
13104	4:39	23:30	331.712	696.860291	Prep-in	23:27	23:30	49.0562592
13105	4:47	26:03	355.128	746.052185	Prep-out	4:47	4:53	100
13105	4:47	26:03	355.128	746.052185	Prep-in	26:00	26:03	78.9011993
13106	4:59	22:15	331.712	696.672058	Prep-out	4:59	5:05	100
13108	5:16	19:24	256.531	538.811768	Prep-in	19:21	19:24	71.8233261
13109	5:43	18:38	224.88	472.351288	Prep-out	5:43	5:49	100
13109	5:43	18:38	224.88	472.351288	Prep-in	18:35	18:38	62.8389015
13110	5:51	25:49	375.128	787.877258	Prep-out	5:51	5:57	100
13110	5:51	25:49	375.128	787.877258	Prep-in	25:46	25:49	65.79319
13111	6:03	18:09	224.68	471.861298	Prep-out	6:03	6:09	100
13111	6:03	18:09	224.68	471.861298	Prep-in	18:06	18:09	68.7153778
13112	6:09	25:22	331.912	697.317017	Prep-out	6:09	6:15	100
13112	6:09	25:22	331.912	697.317017	Prep-in	25:19	25:22	27.2621632
13113	6:19	25:03	343.363	721.182373	Prep-out	6:19	6:25	100
13113	6:19	25:03	343.363	721.182373	Prep-in	25:00	25:03	67.484314
13114	6:21	23:50	300.061	630.359802	Prep-out	6:21	6:27	100
13114	6:21	23:50	300.061	630.359802	Prep-in	23:47	23:50	52.7924614
13115	6:38	23:54	311.712	654.695129	Prep-out	6:38	6:44	100
13115	6:38	23:54	311.712	654.695129	Prep-in	23:51	23:54	53.3706322
13201	4:30	18:59	274.918	701.145813	Prep-out	4:30	4:36	100
13201	4:30	18:59	274.918	701.145813	Prep-in	18:56	18:59	59.0470009
13202	4:31	8:56	100.405	256.081055	Prep-out	4:31	4:37	100
13202	14:59	24:20	181.025	461.713806	Prep-in	24:17	24:20	56.0199013
13203	4:34	8:28	81.725	208.46875	Prep-out	4:34	4:40	100
13203	4:34	8:28	81.725	208.46875	Prep-in	8:25	8:28	47.356369
13203	15:53	25:57	211.971	540.599487	Prep-out	15:53	15:59	100
13203	15:53	25:57	211.971	540.599487	Prep-in	25:54	25:57	72.6493073
13204	4:44	17:45	249.665	636.755859	Prep-out	4:44	4:50	100
13204	4:44	17:45	249.665	636.755859	Prep-in	17:42	17:45	71.9256668
13205	4:52	25:41	394.235	1005.49921	Prep-out	4:52	4:58	100
13205	4:52	25:41	394.235	1005.49921	Prep-in	25:38	25:41	54.7515678
13206	5:00	15:45	204.958	522.72467	Prep-out	5:00	5:06	100
13206	16:08	24:33	165.286	421.552643	Prep-in	24:30	24:33	71.7380371
13207	5:03	17:40	230.496	587.884766	Prep-out	5:03	5:09	100
13207	5:03	17:40	230.496	587.884766	Prep-in	17:37	17:40	55.1845245
13208	5:08	11:15	113.252	288.86264	Prep-out	5:08	5:14	100
13208	14:46	19:09	89.602	228.518417	Prep-in	19:06	19:09	58.7675323
13209	5:14	16:05	208.893	532.768921	Prep-out	5:14	5:20	100
13209	16:19	18:45	53.567	136.605835	Prep-in	18:42	18:45	54.0059166
13210	5:15	19:13	266.927	680.770508	Prep-out	5:15	5:21	100
13210	5:15	19:13	266.927	680.770508	Prep-in	19:10	19:13	76.7313232
13211	5:16	9:05	79.837	203.622681	Prep-out	5:16	5:22	100
13211	5:16	9:05	79.837	203.622681	Prep-in	9:02	9:05	64.661232
13211	14:31	19:05	94.151	240.116714	Prep-out	14:31	14:37	100
13211	14:31	19:05	94.151	240.116714	Prep-in	19:02	19:05	57.9828491
13212	5:20	16:35	199.909	509.861389	Prep-out	5:20	5:26	100
13212	16:59	24:53	154.537	394.141052	Prep-in	24:50	24:53	68.4878616

13213	5:30	8:08	53.56	136.587997	Prep-out	5:30	5:36	100
13213	5:30	8:08	53.56	136.587997	Prep-in	8:05	8:08	83.5992966
13213	9:19	17:11	148.808	379.55542	Prep-out	9:19	9:25	100
13213	9:19	17:11	148.808	379.55542	Prep-in	17:08	17:11	60.9716835
13214	5:39	24:20	344.222	877.917908	Prep-out	5:39	5:45	100
13214	5:39	24:20	344.222	877.917908	Prep-in	24:17	24:20	83.9389114
13215	5:45	24:01	348.056	887.706116	Prep-out	5:45	5:51	100
13215	5:45	24:01	348.056	887.706116	Prep-in	23:58	24:01	65.8709869
13216	5:57	18:10	226.593	577.932129	Prep-out	5:57	6:03	100
13216	5:57	18:10	226.593	577.932129	Prep-in	18:07	18:10	58.9899368
13217	6:12	8:20	48.133	122.789146	Prep-out	6:12	6:18	100
13217	6:12	8:20	48.133	122.789146	Prep-in	8:17	8:20	68.9926376
13217	12:38	16:25	70.843	180.667969	Prep-out	12:38	12:44	100
13217	12:38	16:25	70.843	180.667969	Prep-in	16:22	16:25	74.6911621
13217	16:38	26:00	189.294	482.776367	Prep-out	16:38	16:44	77.9739914
13217	16:38	26:00	189.294	482.776367	Prep-in	25:57	26:00	68.1185074
13218	6:15	8:43	55.167	140.689178	Prep-out	6:15	6:21	100
13218	6:15	8:43	55.167	140.689178	Prep-in	8:40	8:43	74.2187881
13218	15:38	25:30	192.822	491.832794	Prep-out	15:38	15:44	100
13218	15:38	25:30	192.822	491.832794	Prep-in	25:27	25:30	47.6196747
13219	6:26	8:53	59.749	152.376617	Prep-out	6:26	6:32	100
13219	6:26	8:53	59.749	152.376617	Prep-in	8:50	8:53	65.6319656
13219	15:19	17:54	63.619	162.248444	Prep-out	15:19	15:25	100
13219	15:19	17:54	63.619	162.248444	Prep-in	17:51	17:54	59.0281792
13220	6:36	9:07	52.077	132.816345	Prep-out	6:36	6:42	100
13220	6:36	9:07	52.077	132.816345	Prep-in	9:04	9:07	76.6624374
13220	15:23	17:55	63.747	162.566513	Prep-out	15:23	15:29	100
13220	15:23	17:55	63.747	162.566513	Prep-in	17:52	17:55	62.7632141
13221	6:41	9:13	72.239	184.242783	Prep-out	6:41	6:47	100
13221	6:41	9:13	72.239	184.242783	Prep-in	9:10	9:13	59.6134109
13221	15:08	25:54	221.93	566.02832	Prep-out	15:08	15:14	100
13221	15:08	25:54	221.93	566.02832	Prep-in	25:51	25:54	73.4965439
13222	6:56	9:21	57.636	146.990128	Prep-out	6:56	7:02	100
13222	6:56	9:21	57.636	146.990128	Prep-in	9:18	9:21	73.536972
13222	12:55	18:15	106.189	270.813629	Prep-out	12:55	13:01	100
13222	12:55	18:15	106.189	270.813629	Prep-in	18:12	18:15	53.7404366
13223	6:59	9:01	41.44	105.686996	Prep-out	6:59	7:05	100
13223	6:59	9:01	41.44	105.686996	Prep-in	8:58	9:01	73.3113556
13223	10:59	25:52	286.066	729.610107	Prep-out	10:59	11:05	100
13223	10:59	25:52	286.066	729.610107	Prep-in	25:49	25:52	79.4217758
13224	7:06	9:35	43.177	110.129692	Prep-out	7:06	7:12	100
13224	7:06	9:35	43.177	110.129692	Prep-in	9:32	9:35	73.6111832
13224	14:59	25:44	206.562	526.836487	Prep-out	14:59	15:05	100
13224	14:59	25:44	206.562	526.836487	Prep-in	25:41	25:44	68.3438263
13225	7:11	18:05	211.371	539.08606	Prep-out	7:11	7:17	100
13225	7:11	18:05	211.371	539.08606	Prep-in	18:02	18:05	66.732048
13226	7:26	17:45	193.635	493.842682	Prep-out	7:26	7:32	100
13226	7:26	17:45	193.635	493.842682	Prep-in	17:42	17:45	50.0090179