An auction-based approach for prebooked urban logistics facilities

Journal Article

Author(s):
Yang, Kaidi; Roca-Riu, Mireia; Menendez, Monica

Publication date:
2019-12

Permanent link:
https://doi.org/10.3929/ethz-b-000296225

Originally published in:
Omega 89, https://doi.org/10.1016/j.omega.2018.10.005

Working Paper
Original Manuscript

An Auction-based Approach for Prebooked Urban Logistics Facilities

Kaidi Yang, Mireia Roca-Riu, Mónica Menéndez
Institute for Transport Planning and Systems (IVT),
{kaidi.yang, mireia.roca-riu, monica.menendez}@ivt.baug.ethz.ch

The proper management of urban loading/unloading facilities for logistic carriers is crucial for the efficiency of the service. In this paper, we propose a comprehensive and unified approach based on an Auction approach to optimize the use of prebooked loading and unloading facilities. These facilities are now possible due to the advances in technological systems.

The new proposed approach assigns parking spots to companies and computes the price of the service, to maximize the social welfare of the system. It has the following advantages: i) it motivates carriers to participate as they become active members on the auction, giving them freedom to express their preferences; ii) it maximizes the social welfare, resulting in a fair system; and iii) it guarantees truthful participation of the carriers, i.e. the best strategy is to bid based on their real valuation.

An extended set of test instances is solved with an enhanced branch-and-bound algorithm that is able to provide optimal solutions in a reasonable computational time. The solutions where demand is uniformly distributed over the time horizon present better properties (in terms of lower prices and higher assigned valuations, and therefore, higher utility) than test instances where demand is concentrated following a peak hour behaviour.

Key words: Parking Slot Assignment Problem, Combinatorial Auctions, Citylogistics, Mechanism Design, Urban facilities

1. Introduction and Literature Review

Delivery operations in urban areas are crucial for logistic carriers (Ma 2001). Urban logistic facilities are widely used in order to provide a space for efficient delivery operations and to avoid illegal double parking (Beziat 2015). Thanks to the evolution of information and communication technologies, new and intelligent systems have been recently developed to efficiently manage delivery areas through the use of prebooked systems (Munuzuri et al. 2006) and/or dynamic delivery parking spots (Roca-Riu et al. 2017). In these systems, carriers need to express in advance their requests for using the facility, and then each carrier is assigned a starting time following a first-come first-served criteria. The optimal assignment of carrier requests to starting times is known as the Parking Slot Assignment Problem and has been described and studied in (Roca-Riu et al. 2015).

The Parking Slot Assignment Problem for prebooked systems consists of an assignment from requests to parking spots and time slots. During a given time period, e.g. a day or a morning/afternoon time period, a logistics facility offers a given number of parking spots for loading and unloading operations. A set of carriers would like to use this facility at different times, and the problem lies in how to optimally assign requests to time slots in the different parking spots. Each request can be assigned to start at any time instant in any of the possible parking spots. Each request might have a different time preference and a given duration. The assignment solution cannot exceed the capacity of the facility.

Similar problems arise with the parking reservation problem in urban traffic systems (Ayala et al. 2011, 2012, Geng and Cassandras 2011, Mackowski et al. 2015, Lei and Ouyang 2017, Shao et al. 2016, Guo et al. 2016, Du and Gong 2016), e.g. parking reservation for parking lots. In such problems, each driver can reserve a parking spot from a parking lot or a parking agency, either in real-time or in advance. (Ayala et al. 2011) studied a centralized model and a distributed model for the parking spot competition within drivers. In the centralized model, the parking spots were allocated minimizing the total distance from the drivers to the parking spots, whereas the distributed model was based on game theory where the selfish drivers compete for the parking spots. A pricing scheme was further applied in (Ayala et al. 2012) to stimulate the selfish drivers to act in a favorable way for the entire system. However, those works did not consider the time dimension, i.e. they ignored the parking durations and assumed a constant parking demand over time. (Geng and Cassandras 2011) presented a dynamic model for the parking reservation problem based on a moving horizon scheme. Drivers send requests and receive an assigned spot from the parking agency. The drivers can then decide whether to take this assigned spot or not. If the drivers decline the assigned spot, they will compete for parking spots in the next decision step. However, the parking prices are assumed as fixed and given in this work. (Mackowski et al. 2015) and (Lei and Ouyang 2017) proposed a reservation system where the drivers can dynamically reserve a parking spot from a parking agency and the agency can change the parking price over time. The problem was formulated as a bi-level multi-period mathematical program with equilibrium constraints. The upper level was the pricing problem of the agency, which maximized the total economic surplus and, at the same time, minimized the occupancy imbalance penalty. The lower level considered the impact of the pricing on the parking demand. (Mackowski et al. 2015) assumed that the drivers only send requests when they are close to the parking space. (Lei and Ouyang 2017) relaxed this assumption and adopted an approximate dynamic programming algorithm to solve the problem.

Another type of problem reflects the concept of the shared parking systems that enable the private parking spots to be used by public drivers. (Shao et al. 2016) and (Guo et al. 2016) considered a parking space sharing problem where an agency temporarily manages the residential private parking spots from the owners (e.g. during working hours when the owners go out for work), and rent them to the public users to fully utilize the limited parking resources. In such a system, drivers send the requests to the platform in advance (e.g. one day before), and the platform allocates the parking spots to the drivers. A binary integer programming model to optimize the revenue of the agency is formulated in (Shao et al. 2016). Then, (Guo et al. 2016) considered the risk that the parking spots might still be occupied when the owners need them, and applied a simulation optimization method to solve the problem.

Although looking similar, there are some essential differences between the parking reservation problem for urban traffic systems and the Parking Slot Assignment Problem for logistics facilities. First, for urban traffic systems, it is impossible and unnecessary to find the global optimal solution, as the parking reservation problem in urban traffic systems is usually a large scale optimization problem, even with a moving time horizon. Parking facilities need to respond to a large number of parking requests within a short period of time. However, the Parking Slot Assignment Problem for logistics facilities presented in this paper assumes that the parking requests are made one day in advance. Additionally, there are usually a smaller number of parking spaces and a smaller number of requests for logistic deliveries. Therefore, it is possible, and thus more desirable, to find the global optimal allocation of requests in the Parking Slot Assignment Problem. Second, in urban systems, either the parking lots, online platforms or parking agencies aim to maximize their total revenue. However, the parking spots for logistic services are normally public parking spots, so that efficiency and fairness are more important factors to consider. To address the special requirements in the logistic applications, (Roca-Riu et al. 2015) studied the Parking Slot Assignment Problem and formulated it as a mixed integer linear program. Each carrier reports a single desired time window of start and a parking duration. If the parking spaces cannot satisfy all the carriers, (Roca-Riu et al. 2015) proposed a few different penalty functions, such as the number of unallocated requests, the total earliness or tardiness, etc. However, (Roca-Riu et al. 2015) relies on two important assumptions: 1) all the carriers are homogeneous in terms of penalty functions; 2) the time window that all the carriers request reflect their true preferences, i.e. they will not try to get a better assignment by requesting a larger or shorter time window and/or parking duration than their real preference.

This paper proposes a unifed approach from an auction's perspective to relax the two aforementioned assumptions of (Roca-Riu et al. 2015). One advantage of an auction system is that it takes the user heterogeneity into account. Truthfulness can also be satisfied by many of the auction systems.

Thanks to the good properties of the auction systems, they have been attracting increasing attention in the traffic research community, for solving problems like intersection control (Carlino et al. 2013) and highway reservation (Su and Park 2015). A few papers also proposed auction-based parking management systems for urban traffic (Hashimoto et al. 2013, Ayala et al. 2012, Chen et al. 2015, Zou et al. 2015). (Hashimoto et al. 2013) proposed an on-line auction system for parking reservation for electric vehicles, considering electricity trading. In the parking spot competition problem studied by (Ayala et al. 2012), a vehicle-based auction was proposed to set incentive prices such that the Nash equilibrium defines the system optimum. However, both (Hashimoto et al. 2013) and (Ayala et al. 2012) cannot ensure truthful reporting from the drivers. A widely applied mechanism that guarantees truthful reporting is the VickreyClarkeGroves (VCG) mechanism (Nisan et al. 2007) that achieves system optimum. (Chen et al. 2015) proposed an online VCG auction system where the whole time horizon was divided into a few time intervals, and the VCG scheme was applied to each time interval to myopically optimize the system performance. It was also shown that the drivers have incentives to reserve as early as possible. However, (Chen et al. 2015) did not consider the parking durations, and thus considered each time slot separately. Therefore, the optimal allocation within the time horizon of a whole day could not be guaranteed. The static and dynamic auction systems for parking reservation were studied in (Zou et al. 2015). In the static auction system, the drivers were assumed to arrive and request the time slot at the same time. In the dynamic auction system, drivers were assumed to send parking requests when they were close to the parking facility. The requests included their reported arrival time, latest possible waiting time for a parking spot, departure time, and slot valuation. The auction system then assigned the time slot to the drivers and charged a certain price that enabled truth telling. However, the auction system proposed in (Zou et al. 2015) can not be used for prebooked logistic facilities. First, the auction system does not allow prebooking, which is essential for logistic carriers to organize their routes. Looking for a parking spot spontaneously is not very beneficial for logistic carriers. Second, the static auction did not consider the parking durations, and each time interval was solved independently, which is not applicable in logistic facilities as the capacity is normally binding. Third, none of the two systems considered the different valuation functions for starting at different times from the logistic companies.

In order to address the particularities of the logistic parking problem, this paper introduces a new approach from an auction's perspective. An auction is proposed where carriers can bid for time slots. The approach aims at maximizing social welfare and ensures truthful participation. The companies will participate in an auction, where they will bid according to their preferences for different starting times. Each carrier is interested in using the parking facility and has different valuation over this time horizon. Then, the auction will be resolved, providing the optimal assignment: the starting times for companies and the associated prices. We will later prove that under some conditions, the auctions approach is a generalization of the existing penalties approach proposed in (Roca-Riu et al. 2015).

The remainder of the paper is structured as follows. In Section 2 the problem is stated in a general form, and Section 3 proposes the new formulation with mechanism design approach. In Section 4 the algorithm used to solve the formulation is described, which uses an enhanced initial solution and lazy constraints. The equivalence of the new proposed approach with the penalty approach from (Roca-Riu et al. 2015) is shown in Section 5. Section 6 presents the results of an extensive set of experiments and analyzes the features of the solutions provided with the new approach. Finally some conclusions are drawn in Section 7.

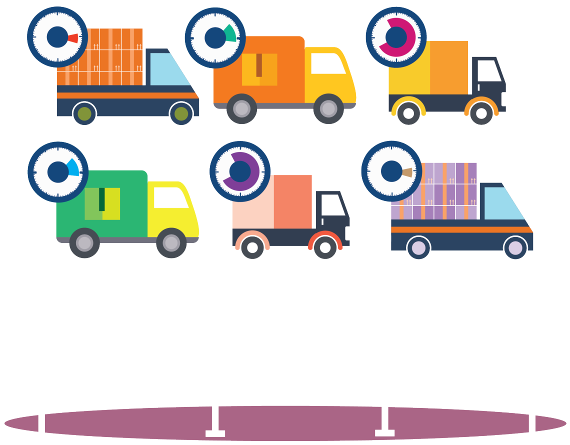
2. Problem

We have a set N of n companies that are interested in using a single loading and unloading facility. This facility offers c parking spots during a time horizon (See Figure 1a). The time horizon can be either one day, or part of a day. Each company $i \in N$ needs to use the parking facility for a duration, s_i, starting within this time horizon. It is assumed that each company has its own time preference on when to start using the parking facility. Such preference can be expressed in different forms, for example, with a valuation vector. It is also assumed that each company has only one such request, and for simplicitity we will use indistinctively the term request or company to refer to an individual request from a company to use the facility. Cases where a company has multiple independent requests for the same facility could be treated as if the requests were coming from multiple companies.

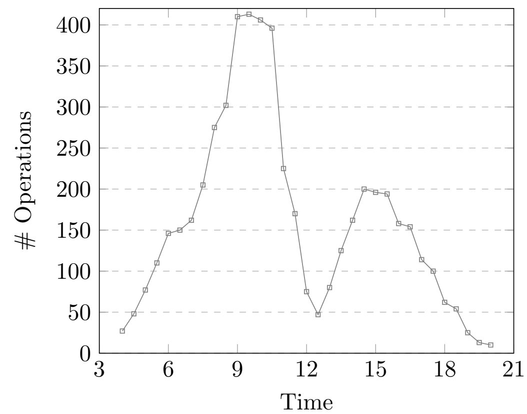
In practice, these loading/unloading areas can serve many companies with different parking durations along the day. As an example, Figure 1b presents an estimation of the number of loading/unloading operations that take place every 30 minutes in an area in the center of Lyon, France. The durations of these activities are generally short; with 90% lasting less than 30 minutes in some cities, like Barcelona, Paris, Valencia, Lyon or Amsterdam (Barcelona City Council 2014, Dablanc and Beziat 2015, Figliozzi 2007).

The problem consists in finding a feasible prebooking allocation of the multiple requests to available time slots, taking into account the time preferences and the service durations of the different companies. The objective is to find the best possible assignment, and maximize the social welfare for the whole group of companies. Thus, we aim to optimize the system performance of all the companies, while providing a fair assignment.

The time horizon $[0, T]$ is considered as a discrete set $\Pi_r = \{t_0, t_1, t_2, \dots, t_r\}$ of $r+1$ homogeneously distributed time instants $(t_0 = 0, t_r = T)$ where the requests may start. This discretization is made for two reasons. First, for operational purposes, the time can not be treated as a continuous variable. In



(a) Vehicles interested in usng a given loading/unloading facility



(b) Estimation of deliveries and pick-ups at the center of Lyon (France)(David and Armetta 2013)

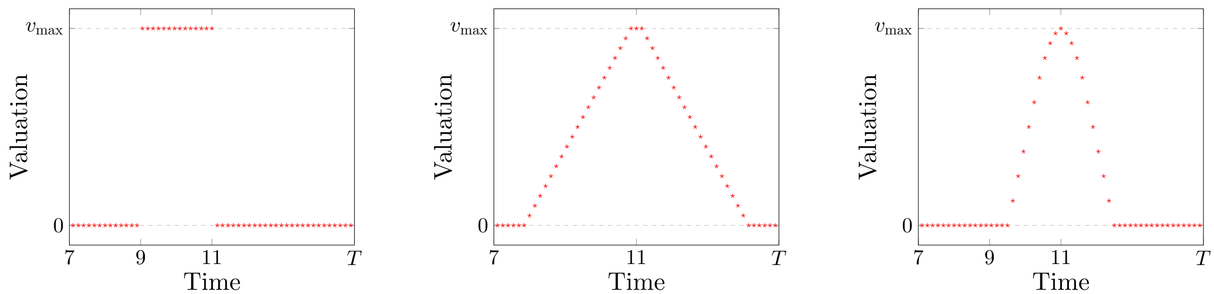
Figure 1 Loading and unloading facilities problem

reality, at most, a minute precision can be feasible. Second, the formulation with time discretization has proved to work better in practice (Roca-Riu 2015), providing enough accurate solutions while requiring less computation time. For notation convenience, we denote $J_r = \{0, ..., r\}$ as the index set of the discrete time horizon Π_r. In the end, the problem is to assign n requests to Π_r time instants in a maximum of c different parking places simultaneously, taking into account the duration of the loading/unloading services. The problem is then a combinatorial optimization problem.

3. An Auction Formulation to prebook Logistics Facilities

In order to solve the optimal assignment of parking requests to time slots while taking companies' time preferences into account, we propose an auction formulation where companies bid for time slots.

We first model the time preference of company i regarding the possible starting times as a non-negative valuation vector $v_i = [v_{i0}, \cdots, v_{ir}]$. Each element $v_{ij} \geq 0$ represents the value of company i for starting to use a parking spot at time instant t_j, $t_j \in \Pi_r$. The companies do not need to reveal the valuation vector in the auction, the model just assumes that this vector exists and that each company can value the different time instants with it. In other words, the parking facility does not need to know the valuation vectors in order to resolve the auction. In addition, the valuation vector is independent of the parking spot being used, since we assume that they are all equal. Instead, it is correlated with the benefits that the company obtains when the request is satisfied at the desired starting time. For example, if a company is not interested in starting parking at a certain time instant at all, then the valuation for this time instant is 0. Note that this valuation vector accomodates any kind of valuation function shape for the companies. Figure 2 shows several valuation vectors with different shapes: Figure 2a is the valuation vector of one company that is only interested in starting the service between 9 and 11 in the morning, then its valuation function will be constant and equal to $v_{\max}$ within this time period, and zero otherwise; Figure 2b represents a company that has the maximum interest at 11h, and the valuation decreasing linearly before and after that time; and Figure 2c represents a company whose preferred time is also at 11h, but the valuation decreases quadratically over time.



(a) Constant (b) Linear (c) Quadratic

Figure 2 Examples of valuation vectors

The proposed auction consists of two steps. In the first step, each company $i \in N$ bids for a parking slot before the considered time horizon (e.g. one day before they need to park). Specifically, the company i reports a vector of bids $b_i = [b_{i0}, \cdots, b_{ir}]$ where each element b_{ij} represents the bid of starting parking at time t_j, and a time duration s_i in minutes when it needs to use the facility. Note that the company could strategically have a different bid vector from its valuation vector. However, as we will later discuss it is desirable for each company to bid truthfully (i.e. $b_{ij} = v_{ij}$).

In the second step, two problems need to be solved: 1) an allocation problem which allocates each request to a time slot; 2) a pricing problem which defines how much each company has to pay for getting the assigned slot. Denote the assignment as a set of binary variables $x = [x_{ij}, i \in N, j \in J_r]$, where $x_{ij} = 1$ if the request of company i is assigned to the starting time t_j, and $x_{ij} = 0$ otherwise. Denote the pricing for each company i as p_i, which usually depends on the allocation x.

The utility for a company is its valuation for getting the assigned starting time minus the price they have to pay for the given assignment. Specifically, the following function describes the utility function for company i that depends on the assignment x:

$$u_i(x) = \sum_{j \in J_r} v_{ij} x_{ij} - p_i(x) \tag{1}$$

where p_i is the price to be paid by company i. Each company is interested in maximizing its individual utility function, $u_i(x)$.

3.1. Design of the auction

The aim is to design an auction (i.e. an allocation rule and a pricing rule) that fulfills the following desired properties.

- The allocation rule maximizes social welfare. Here, the social welfare is the sum of the utilities obtained from all the companies with the given assignment, which represents the total utility of all the companies.
- The auction is dominant strategy incentive compatible or, alternatively named, truthful. So the best strategy for each company is to bid its real valuation, i.e. to provide the true valuations ($b_{ij} = v_{ij}$) and the true duration.
- Under the allocation rule and the pricing rule, the utility of each company is non-negative. In other words, the companies do not suffer any loss by participating in this auction system, i.e. $p_i \leq \sum_{j\in J_r} v_{ij}x_{ij}$

Fortunately, the VickreyClarkeGroves (VCG) Mechanism for combinatorial auctions (Nisan et al. 2007) satisfies the above properties, and aids on the design of the desired auction.

The allocation rule should maximize the social welfare, which is the sum of all the valuations obtained for each of the companies with a given assignment. However, as the true valuations are not known, the allocation problem instead optimizes the assignment according to the bids submitted. The optimal allocation x^* is obtained solving the following integer program:

$$\max_x \sum_{i\in N}\sum_{j\in J_r} b_{ij}x_{ij} \tag{2}$$

$$\sum_{j\in J_r} x_{ij} \leq 1, \quad i = 1, ..., N \tag{3}$$

$$\sum_{i\in N}\sum_{k\in \mathcal{T}_{ij}} x_{ik} \leq c, \quad j \in J_r \tag{4}$$

where $\mathcal{T}_{ij} \subseteq J_r$, with $j \in J_r$ and $i \in N$, such that $\mathcal{T}_{ij} = \{k \in J_r | t_k \geq 0, t_k < t_j, t_k > t_j - s_i\}$.

The allocation rule is defined by the solution of the integer programming problem (2–4). The starting times are assigned with the objective of maximizing total social welfare, while fulfilling the capacity restrictions of the parking spots. Constraints (3) impose that for every request at most one assignment is active for the whole time horizon. Constraints (4) guarantee that at a given time instant t, at most c requests (one per parking spot) are being served simultaneously. When we assign a request to a given starting time, that does block the parking spot for the duration of the service to the assigned request. For that reason, Constraints (4) check how many assignments are active at each time instant t_j. In order to count the active assignments, it is not enough to add all assignment variables at time index $j \in J_r$ for every request $i \in N$, but also the assignment variables from previous time indices k such that $t_k < t_j$ and $t_k > t_j - s_i$, that could still be active as the assignments last for a duration of s_i. The previous features of the allocation rule show the combinatorial nature of the auction. The formulation simplifies the assignment to the starting time. However, this assignment is only valid if it stays active during the whole duration of the loading/unloading activity. Constraining the possibility of simultaneously using more than c parking spots converts the assignment in a combinatorial problem.

Once the allocation rule has been defined, we define a payment rule such that it guarantees truthful bidding from carriers. As we will see, to guarantee truthful bidding, we ensure that each request pays its externality to the system. That is achieved by solving the following problem for each request $i \in N$:

$$p_i(x) = \sum_{h \neq i} \sum_{j \in J_r} b_{hj} y^*_{hj} - \sum_{h \neq i} \sum_{j \in J_r} b_{hj} x^*_{hj} \tag{5}$$

where x^* is the optimal solution of the problem of the allocation rule, and y^* is the solution to the following problem, which is the same problem solved previously in the allocation rule, but without company i:

$$\max_y \sum_{\substack{h \in N \\ h \neq i}} \sum_{j \in J_r} b_{hj} y_{hj} \tag{6}$$

subject to

$$\sum_{j \in J_r} y_{hj} \leq 1 \qquad h = 1, \ldots, i-1, i+1, \ldots, N \tag{7}$$

$$\sum_{\substack{h \in N \\ h \neq i}} \sum_{k \in \mathcal{T}_{hj}} y_{hk} \leq c \qquad j \in J_r \tag{8}$$

where, as before, $\mathcal{T}_{hj} \subseteq J_r$ and $h \in N$, with $j \in J_r$, such that $\mathcal{T}_{hj} = \{k \in J_r | t_k \geq 0, t_k < t_j, t_k > t_j - s_h\}$. Problem (6–8) gives the solution to the maximum social welfare from parking slot allocation when i does not participate, and the other drivers' welfare is maximized. Clearly, the solution to this problem does not depend on company's i valuation v_{ij}, bid b_{ij}, and neither on its request duration s_i.

Finally, the utility received by the company i is:

$$u_i(x^*) = \sum_{i \in N} \sum_{j \in J_r} b_{ij}(x^*) x^*_{ij} - \sum_{h \neq i} \sum_{j \in J_r} b_{hj}(y^*) y^*_{hj} \tag{9}$$

where the first term is the maximum social welfare which is achieved under allocation x^*, and the second term is independent of valuation v_{ij}, and bidding b_{ij}. Then, the payment scheme aligns each company's utility maximization with the system welfare maximization objective. Note that this expression is obtained from Eq. (1), with the optimal solutions (x^*, y^*) from problems (2)–(4) and (6)–(8).

The VCG mechanism (the allocation rule and pricing rule) ensures truthful bidding and non-negative utilities for all companies. The general proof can be found in (Nisan et al. 2007).

The concept is intuitive in a single-item auction, where the payment rule ensures that companies can not get better utility by bidding different from their valuation. As already mentioned, the payment rule defined by the VCG mechanism is that each company pays the externality created to the system, i.e., the difference on social welfare triggered by the company participating on the auction. The allocation rule assigns the item to the company with the highest bid. The winning company has to pay a price equivalent to the difference in the social welfare of the rest of the companies comparing two situations: either the company is present in the auction or it is not. Note that the social welfare of the other companies when this company is present is 0. The social welfare when the company is not present is then the second highest bid, so this is the price that the winning company has to pay.

Let us analyze the case from a given company perspective ($i \in N$), with v_i utility valuation, b_i the bid and $\max_{h \neq i} b_h$ as the highest bid of all the companies except the company being analyzed. If the bid of the company (b_i) is lower than the highest bid of the rest ($b_i < \max_{h \neq i} b_h$) the company loses the auction and receives 0 utility. If the bid is higher ($b_i > \max_{h \neq i} b_h$), company i wins and receives as utility its valuation minus its price, which is equal to the highest bid of the rest ($v_i - \max_{h \neq i} b_h$). Let us see that this value is the highest possible when the company bids its real valution, i.e. when $b_i = v_i$. On one hand, if the real valuation of the company is smaller than the highest bid of the rest of the companies ($v_i < \max_{h \neq i} b_h$) the maximum utility that the company can get is zero. Therefore, there is no reason why the company could offer a different bid from its real valuation, as that would result in a higher utility. On the other hand, if the valuation is higher than the highest bid of the rest of the companies ($v_i > \max_{h \neq i} b_h$), the utility that the company gets is ($v_i - \max_{h \neq i} b_h$), which is achieved when company sends the real valuation as his bid ($b_i = v_i$). In this case, the company bids truthfully and wins.

4. Improved Solution Algorithm

According to (Lehmann et al. 2002) there is no polynomial time algorithm for the allocation problem. Moreover, we need to solve the pricing problem once for each company, i.e. n times for solving all the prices. Note that each of the pricing problems has the same structure as the allocation problem, with only one less variable. Hence, if the number of requests is large, the computation complexity of solving both problems is very high. Fortunately, we can exploit the similarity between the allocation and pricing problems to reduce the computation time. In this section, we propose to improve the general solution algorithm for the mixed integer linear program based on the Branch and Bound (B&B) by exploiting two properties of the problem. In Section 4.1, we use the solution to the allocation problem as an initial solution to each of the pricing problems. In Section 4.2, we relax some constraints on the initial formulation and add them whenever they are needed within the solving procedure as lazy constraints, i.e. they are only active when the current solution violates some of these constraints.

4.1. Initial solution

The optimal solution to the allocation problem $x^*_{hj},\ h \in N,\ j \in J_r$ provides obviously a feasible solution to the pricing problems. For the pricing problem of request i, it is obvious that $x^*_{hj},\ h \neq i,\ j \in J_r$ is one feasible solution. That is to say, the solution obtained in the allocation problem when all companies participate in the auction, can serve as an initial solution to the n pricing problems that have to be solved. The effectivity of this strategy will be later evaluated in Section 6.4.

4.2. Capacity constraints as lazy constraints

When a formulation is solved with lazy constraints, this set of constraints is not generated on the inital formulation, but only later, in case the obtained solution violates some of them, they are added to the formulation. Lazy constraints have proven to be specially useful when solving problems with a large set of constraints, e.g. the subtour elimination constraints of exponential size for the travelling salesman problem (Laporte 1992). Although the number of constraints of both the proposed allocation problem and the pricing problems are not large compared to the number of assignment variables, treating the capacity constraints (4) and (8), from the allocation and the pricing problem respectively, as lazy constraints can still be useful to reduce the computation time. The reasons are two-fold. First, the companies are expected to have the highest time preferences in certain time periods, e.g. mornings or afternoons. In the other time periods, such as early morning or late night, the time preferences are usually quite low. As a result, the capacity constraints may only be active for a small number of time slots. Second, as the pricing problem has a similar structure to the allocation problem, it is very likely that the pricing problems have a similar set of active capacity constraints. Hence, the set of capacity constraints for the pricing problem can be initiated as the resulting set of active capacity constraints obtained when solving the allocation problem. Thanks to this, the pricing problems are expected to be solved faster.

The application of the lazy constraints in this paper is summarized in Algorithm 1 (detailed below), which is used to solve both the allocation and the pricing problems. For the allocation problem, the initial set of capacity constraints can be either empty or user defined according to the structure of the valuation vectors. For example, if the valuation vectors express a high demand of service in a subset of time instants, the capacity constraints of these time instants can be included in the initial set. After solving the allocation problem, we use the output set of capacity constraints to initialize each pricing problem. Moreover, in order to improve the performance of the algorithm, at each algorithm iteration we may also add more constraints besides the single violated constraint. Let $\bar{L}$ be the whole set of capacity constraints of the given problem ((4) or (8)), and we use the notation of $l_j \in \bar{L}$, $j \in J_r$ for one capacity constraint, as they are indexed in subset J_r. One possible way is to add all constraints between l_j and l_k as long as t_j and t_k are close enough, e.g. $0 < t_j - t_k < \kappa$ (κ is a positive parameter of this algorithm). We do this because if one constraint is active in the optimal solution, it is possible that the nearby constraints are also active. At each iteration a small subset of constraints can be added to the formulation, which might avoid the need to solve the problem once for every new violated constraint found.

Algorithm 1 Algorithm with lazy constraints formulation

Input: An initial set of capacity constraints, $\tilde{L} \subset \bar{L}$, where $\bar{L}$ is the whole set of capacity constraints ((4), (8)). Note that $l_j \in \bar{L}$, $j \in J_r$ is one capacity constraint, which is indexed in subset J_r.

Output: The optimal solution x^*.

1: Initialize the current set of capacity constraints as $L \leftarrow \tilde{L}$;
2: Solve the problem with L and obtain the optimal solution $\hat{x}$ of the relaxed problem.
3: **while** $\exists\ l_j \in \bar{L} \setminus \tilde{L} \mid l_j$ is violated in $\hat{x}$ **do**
4: Add l_j to the constraint set, i.e. $L \leftarrow L \cup \{l_j\}$
5: Add $l_k \in \bar{L} \setminus \tilde{L} \mid 0 < t_j - t_k < \kappa$ to the constraint set, i.e. $L \leftarrow L \cup \{l_k\}$
6: Solve the problem with L and obtain the optimal solution $\hat{x}$ of the relaxed problem.
7: $x^* := \hat{x}$ The optimal solution of the relaxed problem $\hat{x}$ is the optimal solution.

In this paper, we will evaluate the following four versions of the solution algorithm.

- **Branch and Bound (B&B)**: the standard B&B algorithm as implemented in `CPLEX`. The allocation problem and the pricing problems are solved independently without considering the relation between them.
- **B&B with Initial solution**: the B&B algorithm implemented in `CPLEX` providing initial solutions to the pricing problems. The initial solutions are derived from the optimal solution to the allocation problem, as explained in Section 4.1
- **B&B with Lazy Constraints**: the B&B algorithm implemented in `CPLEX` where capacity constraints are handled as lazy constraints. The initial set of lazy constraints of the pricing problem is derived from the allocation problem, as explained in Section 4.2.

- **B&B Combined**: the B&B algorithm implemented in `CPLEX` with both the initial solution and the lazy constraints.

5. Comparison of Penalty and Auction's Formulations

The auction formulation proposed in Section 3 is based on a valuation vector that each company has for the time of service. From another perspective, (Roca-Riu et al. 2015) assumed that companies express their valuations of the different times through the use of time windows. In this section, we show that the proposed auction formulation provides a unified approach that generalizes the formulation of (Roca-Riu et al. 2015).

In the time-window formulation (Roca-Riu et al. 2015), each company is interested in using the parking facilities during a time window ($[a_i, d_i]$), and during this time window their valuation is the same. In other words, the benefits that each company obtains are equal as long as the assigned starting time fulfills the time window. However, when the assigned starting time is outside the requested time window, a penalty is incurred. This penalty expresses somehow the inconvenience that the assignment represents for the company. Several objective functions are proposed based on a *fairness* idea, by means of different penalties. Each penalty is defined to account for different possible perceptions of the real inconvenience caused. As a generalization of the penalty concept, we can use a penalty formulation as follows:

$$\phi_{ij} = \begin{cases} 0 & t_j \in [a_i, d_i] \\ e_{ij} > 0 & t_j \notin [a_i, d_i] \end{cases} \tag{10}$$

where the penalty ϕ_{ij} is the inconvenience that company i experiences with starting time t_j, and the values e_{ij} vary according to the penalty definition, but are always positive. In this context, the overall penalities are minimized in the proposed formulation. Recall that x_{ij} is 1 when company i is assigned the starting time t_j, then we can write the objective function for minimizing penalties as follows:

$$\Phi(x, \phi) = \min \sum_{i \in N} \sum_{j \in J_r} \phi_{ij} x_{ij} \tag{11}$$

In the auction formulation, the objective function aims to maximize the social welfare, and the valuation is expressed through the companies' bids. Since we have designed a truthful mechanism, we know that companies bids are equal to their valuation, then the objective function is the maximization of the overall valuation, as follows:

$$\mathcal{U}(x, v) = \max \sum_{i \in N} \sum_{j \in J_r} v_{ij} x_{ij} \tag{12}$$

We aim to provide the valuation vectors v_{ij} such that Eq.(11) and Eq.(12) are equivalent. To this end, we define the valuation v_{ij} such that

$$v_{ij} = v_{\max} - \alpha\phi_{ij} \tag{13}$$

where $v_{\max}$ is a large constant value, and α is a factor parameter. Note that in order to have a meaningful v_{ij}, the values of $v_{\max}$ and α should be properly chosen such that $v_{ij} \geq 0$, $\forall i \in N, j \in J_r$. Such requirement results from the desired property that the companies do not suffer any loss by participating in the system.

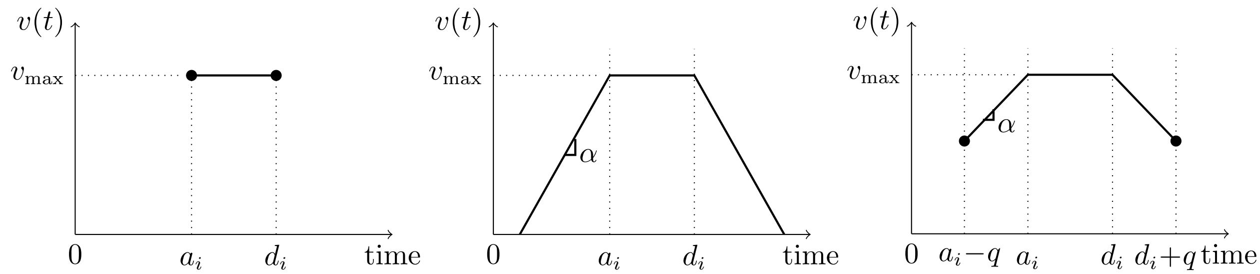
With Eq.(13), the equivalence holds. First, the assignment fulfills Constraint (3) such that for each company at most one time interval is assigned. Second, the objective functions are equivalent. Since $v_{\max}$ is constant, the corresponding term can be removed from the objective function. Similarly, α is only a factor that can also be removed from the optimization.

$$\mathcal{U}(x,v) = \max \sum_{i\in N} \sum_{j \in J_r} v_{ij}x_{ij} \iff \max \sum_{i\in N} \sum_{j \in J_r} (v_{\max} - \alpha\phi_{ij})x_{ij} \iff$$
$$\max \sum_{i\in N} \sum_{j \in J_r} v_{\max}x_{ij} + \max - \Big[\sum_{i\in N} \sum_{j \in J_r} \alpha\phi_{ij}x_{ij}\Big] \iff \min \sum_{i\in N} \sum_{j \in J_r} \phi_{ij}x_{ij} = \Phi(x,\phi) \tag{14}$$

In summary, we have proven that the formulation proposed in Section 3 is a generalization of the one proposed in (Roca-Riu et al. 2015). The new formulation has a utility perspective instead of a penalty perspective. However, the penalties can also be interpreted from the utility perspective, when the valuation that each company has for the time instants is defined as a constant positive valuation minus the penalties incurred. Then, the companies give a constant positive valuation during the time window, and the valuation decreases outside the time window as the penalties increase. In particular, each penalty formulation corresponds to a specific company valuation. In the Appendix A, this equivalence will be shown for the original formulations designed on (Roca-Riu et al. 2015), which correspond to some basic companies valuation, see Figure 3. Note that the valuation functions described there are only a particular example, but the formulation is valid for any shape of these valuation functions, as discussed with Figure 2.

6. Results

In this section, we present the results obtained in a series of computational experiments to analyze and compare the different models studied in this paper. First, the two sets of test instances ($S1$, $S2$) that will be used in the experiments are described. Then, we compare the new auction's formulation and the penalty formulation from (Roca-Riu et al. 2015) experimentally in Section 6.2. Next, the sensitivity of the algorithm to the level of discretization of the time horizon is studied in 6.3. In Section 6.4 different solution approaches are compared in terms of efficiency and computational time. Finally, Section 6.5 analyzes some interesting features of the solutions provided with the auction system.



(a) Binary (b) Trapezoid (c) Truncated Trapezoid

Figure 3 Basic companies valuation

6.1. Test instances

As mentioned in the previous sections, a System for Prebooked Urban Logistics Facilities was studied in (Roca-Riu et al. 2015) with the Parking Slot Assignment Problem approach. The set of test instances proposed there will be used here and referred to as the $S1$ set, to perform some of the experiments of this section. The set has 60 test instances (numbered from 1...60) and are based on patterns observed in an experimental study in the city of Barcelona, Spain. The number of parking places ranges from 2 to 8, and requests are expressed as time windows. The requests are distributed according to a triangular pattern around a peak hour that is located either in the center or at the beginning of the morning or afternoon interval. All details can be found in (Roca-Riu et al. 2015), and the original instances can be downloaded from http://mrocariu.github.io/code/.

In order to evaluate the new features that the unified approach offers, a new test instance set ($S2$) has been created. This set has been specifically designed to perform a sensitivity analysis later in Section 6.5. With the new approach, each company has its valuation vector over the different time intervals. If we combine the valuation vectors of all the companies together, we can observe the average valuation per time instant throughout the day. In this way, one can get a general idea of the demand for using a loading/unloading area at each time instant.

All individual valuation vectors have the same shape with vectors of the following form.

$$v_{ij} = v_{\max} e^{-\frac{(t_j - \tau_i)^2}{2\sigma^2}} \tag{15}$$

where τ_i is the time instant with the highest valuation, σ is the shape parameter of the valuation vector, and $v_{\max}$ is the maximum valuation. In this paper, σ is equal for all requests and instances, and takes value of 60 min. In order to account the heterogeneity of the companies, $v_{\max}$ can take the value of either 10, 15 or 20. The combination of τ_i and $v_{\max}$ is determined such that the average valuation per instance adjusts to one of the two types of the average valuation patterns (uniform or peak). They are shown in Figure 4. The uniform pattern has a uniform valuation between 8:00-14:00, whereas the peak pattern has a peak between 10:00-12:00.

For each average valuation pattern, we evaluate the system with different number of parking spots and different parking durations. As already mentioned, there are two average valuation patterns: uniform and peak. The number of parking spots is either 3 or 6; and the parking duration can be 10 min, 20 min or 30 min. The instances with 3 parking spots have 60 requests and the instances with 6 parking spots have 120 requests. For each possible combination of the previous three features, 100 instances are randomly generated, so in total $S2$ has 1200 new instances. In particular, we will call Uniform and Peak the sets of instances with uniform and peak valuation patterns respectively, and number them from 1001...1600, and from 2001...2600.

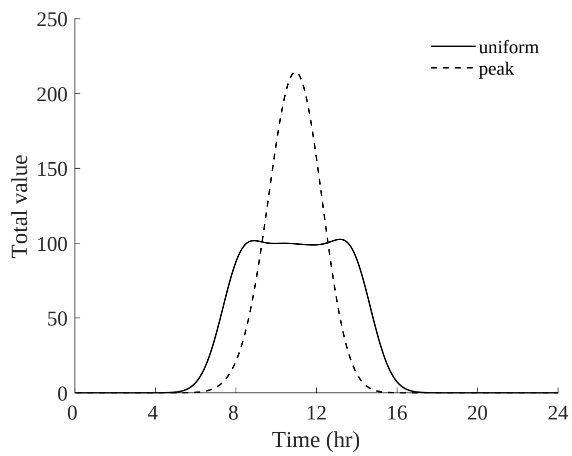


Figure 4 **Average valuation pattern**

6.2. Experimental comparison of formulations

In this section, we will prove that the results obtained from the new formulation and the valuation functions defined in Section 5 are equivalent to the previous results in (Roca-Riu et al. 2015). Particularly, the instance set $S1$ will be solved with both formulations (penalty and auction), and the equivalence between the obtained optimal objective values will be checked.

Using the formulation developed in Section 5, we can compare the values of the two objective functions, the valuation approach and the penalty approach.

$$\begin{aligned}\mathcal{U}(x,v) &= \max\sum_{i\in N}\sum_{j\in J_r} v_{ij}x_{ij} = \max\sum_{i\in N}\sum_{j\in J_r}(v_{\max}-\alpha\phi_{ij})x_{ij}\\ &= \max\sum_{i\in N}\sum_{j\in J_r} v_{\max}x_{ij} + \max - \sum_{i\in N}\sum_{j\in J_r}\alpha\phi_{ij}x_{ij}\\ &= nv_{\max} - \alpha\min\sum_{i\in N}\sum_{j\in J_r}\phi_{ij}x_{ij} = nv_{\max} - \alpha\Phi(x,\phi) \end{aligned} \tag{16}$$

After some preliminary tests, $v_{\max} = 100$ and $\alpha = 0.1$ are used for the experiments in order to guarantee that all the valuations are positive with the trapezoid and truncated trapezoid valuation functions. Another consideration is that $v_{\max}$ should not be too large in order to avoid numerical issues. In the case of the binary valuation, $v_{\max} = 1$ and $\alpha = 1$ are sufficient as there is only two possibilities for penalty and valuation, either zero or one. Note that the choice of these two variables do not represent the valuations in reality. They are only adopted here to show the equivalence between formulations as a proof of concept. It is shown in Table 5 in Appendix B that the proposed formulation is equivalent to the formulations in (Roca-Riu et al. 2015).

Table 5 presents the equivalence of the formulations with the three corresponding valuation functions analyzed in section 5: Binary Valuation, Trapezoid Valuation and Truncated Trapezoid Valuation. For each valuation function, $\Phi(x, \phi)$, $\mathcal{U}(x, v)$ are evaluated, and column named (16) contains a zero if (16) is fulfilled. In all instances, the two different optimal values obtained with each model are equivalent, i.e. they give the same optimal value expressed in a different objective function.

6.3. Sensitivity to level of discretization

In this section, we study the impact of different levels of discretization of the time horizon on the algorithm solutions. To this end, we test four discretization levels, i.e. 1 minute, 2 minutes, 5 minutes and 10 minutes, in the instance set $S1$ that are solved to optimality with the general solution algorithm. For presentation simplicity, the results for a subset of $S1$ are shown (instances 3, 16, 33, 36, 38 and 41). The results for the other instances follow a similar trend. All the information on the request, i.e. parking durations and bids, need to be adapted to the time instant discretization. Particularly, the valuation is converted by sampling. An example of the conversion is given by Figure 5. The valuation of the illustrated instance reaches its maximum within time window [420, 432]. In the 10 minutes discretization, there are only positive valuations at 410, 420, 430 and 440. The valuations at these time instants remain the same despite of the discretization time.

The parking duration is adapted as the minimum integer number of time intervals that provides a larger duration than required. Denote s_i^m as the parking duration for a discretization precision of m minutes.

$$s_i^m = \lceil s_i^1 / m \rceil \tag{17}$$

The conversion of the parking duration can lead to waste of parking capacity, as the assigned duration is longer than required. Similarly, a coarse level of discretization provides less flexibility in the assignment problem (as it reduces the number of time instants). These could result in a slightly worse social welfare.

The social welfare value (objective function) and the computation times are compared in Table 1 and Table 2, respectively, for the different possible valuations (trapezoid, truncated trapezoid

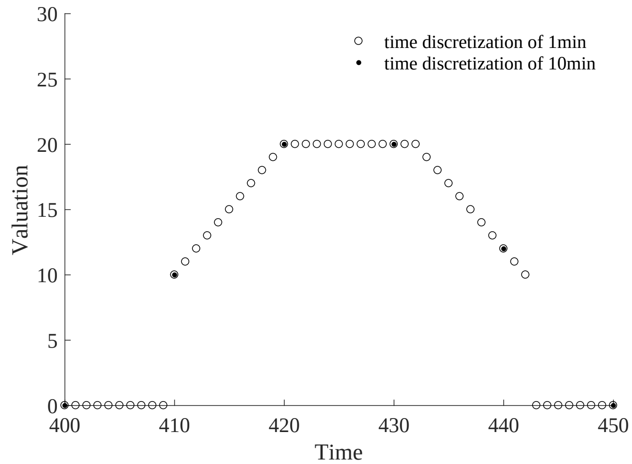


Figure 5 Illustration of the conversion between different levels of time discretization

and binary). The last rows of Table 1 present the average objective function, and the absolute difference to the 1 min interval due to the discretization. The last row of Table 2 presents the percentage average computation time savings compared to the 1 min interval.

Table 1 Impact of the level of time discretization on social welfare

Inst.	Trapezoid				Truncated Trapezoid				Binary			
	1 min	2 min	5 min	10 min	1 min	2 min	5 min	10 min	1 min	2 min	5 min	10 min
3	5993.6	5988.6	5973.5	5939	5993.6	5988.6	5973	5936	5800	5700	5500	5200
16	8691.3	8685.8	8663.5	8623	8691.3	8685.8	8663.5	8623	8400	8300	7900	7500
33	4970.6	4967.8	4960	4944	4970.1	4967.2	4957.5	4937	4400	4400	4200	4200
36	6138.9	6131	6103.5	6060	6138.7	6130.8	6101	5875	4800	4800	4600	4200
38	13693.8	13690.6	13675	13643	13693.8	13690.6	13675	13643	13400	13400	13100	12700
41	5467	5464	5446.5	5421	5465.6	5462.8	5442	5327	4800	4700	4500	4300
avg.	7492.5	7488.0	7470.3	7438.3	7492.2	7487.6	7468.7	7390.1	6930.3	6883.3	6633.3	6350.0
	-	(-4.5)	(-22.2)	(-54.2)	-	(-4.6)	(-23.5)	(-97.5)	-	(47)	(297)	(680.3)

Table 2 Impact of the level of time discretization on the computation time (seconds)

Inst.	Trapezoid				Truncated Trapezoid				Binary			
	1 min	2 min	5 min	10 min	1 min	2 min	5 min	10 min	1 min	2 min	5 min	10 min
3	1643	160	31	17	1188	57	16	8	87	30	9	5
16	3044	376	64	29	1444	167	27	13	221	66	19	12
33	444	85	25	9	123	32	13	3	52	18	6	3
36	1458	245	47	14	620	78	27	13	205	43	8	5
38	3058	914	157	96	841	300	63	32	445	139	43	21
41	533	170	32	10	448	44	15	6	68	25	7	4
avg. saving	-	80%	96%	98%	-	83%	95%	98%	-	71%	92%	95%

As expected, it can be seen that the social welfare (objective function) decreases as the discretization level becomes coarser, i.e. when time intervals are longer. However, the decrease is relatively small when changing the precision from 1 minute to 2 minutes, while the computation time drops drastically (71-83%). The rest of the experiments on this section are performed with a discretization level of 2 minutes, which offers a reasonable balance between the loss in precision in the objective function and the reduction of the computation time.

6.4. Efficiency of solution improvements

In this section, the four different versions of the solution algorithm developed in Section 4 will be evaluated and quantified. Recall that we test the four following versions: B&B, B&B with initial solution (Initial), B&B with lazy constraints (Lazy) and the combined one (Com.). Parameter κ is chosen as 10 for the Initial and Combined algorithms.

Table 3 Computation time of the four algorithms (seconds)

Inst	Trapezoid				Truncated Trapezoid				Binary			
	B&B	Initial	Lazy	Com.	B&B	Initial	Lazy	Com.	B&B	Initial	Lazy	Com.
1	1392	1111	676	649	413	366	383	281	117	132	106	103
2	714	482	357	298	581	537	616	555	50	51	54	48
3	160	115	59	48	57	48	53	43	30	32	26	26
5	1711	1185	763	628	792	624	774	516	146	124	130	104
6	1194	762	714	508	440	305	454	436	62	64	59	57
7	854	465	309	227	281	222	250	203	88	85	72	76
12	624	356	252	201	222	169	238	190	59	54	49	48
15	14	12	5	5	7	6	5	4	5	5	4	4
16	376	280	161	116	166	130	135	110	66	63	62	50
18	1622	1213	478	464	1238	983	1168	920	212	202	192	173
23	1965	1243	669	510	753	571	672	653	166	163	144	143
24	376	286	249	230	182	137	171	142	28	27	23	23
26	706	467	264	220	209	173	225	168	68	68	63	59
28	78	72	33	31	34	30	28	26	20	16	17	14
30	9	8	3	3	4	4	3	3	3	3	3	3
32	1300	1638	934	608	487	365	470	344	172	161	167	171
33	85	56	44	34	32	26	29	25	18	19	18	18
35	1027	799	472	391	435	323	358	285	172	191	170	159
36	245	162	130	108	78	74	85	64	43	42	43	45
38	914	535	348	268	300	229	248	194	139	158	134	136
39	851	613	478	359	814	641	850	616	181	200	219	212
40	2180	1281	1055	715	756	520	676	502	247	238	258	238
41	170	83	89	51	44	34	48	38	25	24	26	23
42	1934	1057	739	506	660	453	416	352	227	251	224	217
43	413	301	169	156	126	109	114	100	80	84	77	77
44	11851	8226	6404	6014	5148	3920	6205	3485	25811	51084	13809	27153
45	29409	51088	13377	12371	4908	2991	4840	2965	47475	51102	16333	6748
Median	851	482	348	268	300	229	250	203	80	84	72	76

The four solution algorithms are tested with the $S1$ instance set. Table 3 summarizes the computation time of the different algorithms with the three valuation possibilities (trapezoid, truncated trapezoid, and binary). As mentioned in the previous section, the discretization interval is set to 2 minutes. The instances where all the requests were satisfied within the maximum valuation time were excluded from the table, as these instances are easy to solve, and no pricing problem needs to be solved. The rest of the instances are solved in less than one hour, except for instances 244 and 245. Also the allocation problem is always solved in less than 20 minutes, and each pricing problem takes less than 10 minutes. The computation time is the total time for solving both the allocation problem and all the pricing problems. The fastest algorithm for each instance and valuation is highlighted with shadings. The optimal solution of all problems (allocation and pricing) is obtained by all of the four solution algorithms in the test set $S1$.

The computation times presented are sufficient for a real application of a prebooking system, as we require the companies to submit the requests one day ahead. The maximum number of requests handled is 235 (Instance 245). It is expected that instances of larger size can also be solved if we allow longer computational times.

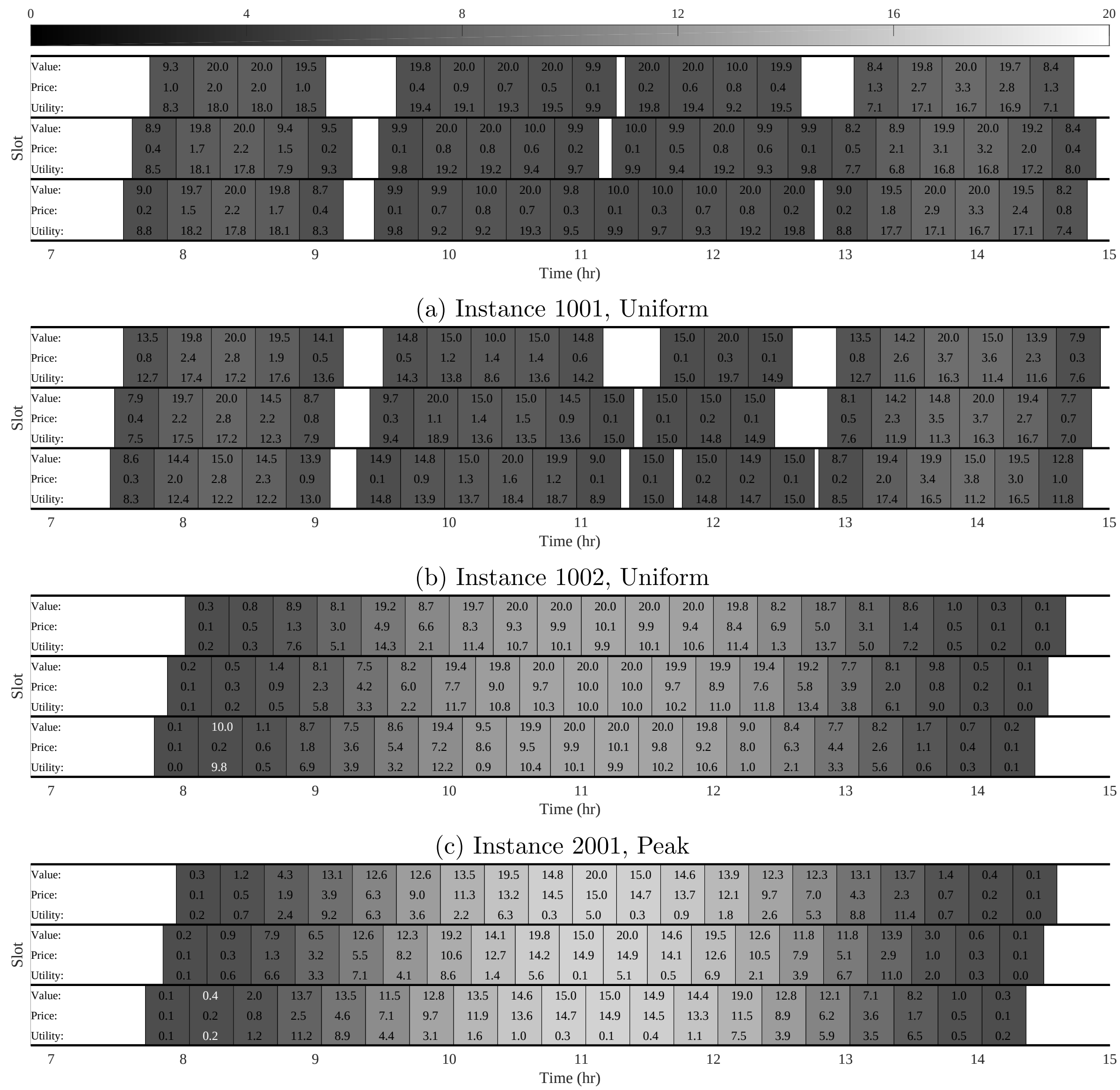
It can be seen from Table 3 that the three modified B&B versions outperform the standard B&B algorithm in most scenarios tested. However, for the binary valuation, it seems that the standard B&B algorithm performs rather good. One possible explanation is that the binary valuation functions enable a sparse structure of both the allocation and pricing problems, which is handled well with `CPLEX` software. For trapezoid and truncated trapezoid valuations, there is not much sparsity to exploit.

Table 3 shows that the combined solution algorithm provides the optimal solution fastest in most of the scenarios. This is particularly true for trapezoid valuation. For some instances with truncated trapezoid or the binary valuation, applying only the initial solution or lazy constraints might outperform the combined one. This shows that the combined approach might not be faster in all instances. Nevertheless, the combined approach still works the best overall in terms of computation time, which is illustrated in the last row of Table 3 with the median of the computation times.

6.5. Analysis of the auction system

This subsection discusses the properties of the auction system. We use the allocation and pricing solutions of the new instances to evaluate the impact of the different instance features: the companies' valuation vectors, the number of parking spots, and the parking duration. Particularly, we study how the values of the requests at the assigned time instant (assigned value), the prices charged from the requests, and the utilities of the requests change with the companies' valuation, the number of parking spots, and the parking duration. As an example, we illustrate the aforementioned properties in the new set of instances $S2$. First, an analysis of the individual instances is conducted in Section 6.5.1. Second, an aggregated analysis is presented with the entire results in Section 6.5.2.

6.5.1. Individual instance analysis. Figure 6 illustrates the allocation and pricing solutions of four instances, i.e. Instances 1001, 1002, 2001 and 2002, the first two of Uniform and the first two of Peak. Instances 1001 and 2001 only have a $v_{\max}$ of 10 and 20, whereas 60% of the requests in Instances 1002 and 2002 have a $v_{\max}$ 15, and the rest 40% have a $v_{\max}$ 10 and 20. It can be seen



(a) Instance 1001, Uniform

(b) Instance 1002, Uniform

(c) Instance 2001, Peak

(d) Instance 2002, Peak

Figure 6 Plot of assigned individual instances (3 parking spots, parking durations are 20 minutes). Each colored rectangle represents a request. The heatmap colors are determined based on the price of each request. The color changes from black to white as the price increases. The numbers on each request represent the assigned value, the price, and the utility, respectively.

by comparing Figure 6a and Figure 6b (also Figure 6c and Figure 6d) that despite the difference in valuation functions, the allocation and pricing solutions of different instances are overall similar if they have the same valuation pattern. This is particularly true for the prices. This is expected, as the valuation patterns determine how the valuation function of the requests distributes over the time horizon. For the same valuation pattern, the distributions of requests for different instances are very similar, which leads to the similar allocation and pricing solutions. A more general discussion will be given in Section 6.5.2. Nevertheless, apart from the difference in valuation functions in general, we can observe some randomness for the values of the requests, e.g. the second leftmost requests in the third slot for the peak demand pattern (Figure 6c and Figure 6d, marked in white). These two requests are assigned at a similar time slot, but with very different assigned values and utilities. This is due to the randomness of the requests. Although for peak demand pattern, most companies prefer the slots in the peak hour. Still with a low probability, there are some companies with the highest value in the non-peak hour (the marked request in Figure 6c). This request can easily be allocated at the time slot with the highest value and charged with a low price, as the competition at this time slot is not intense.

It is also shown that the allocation and pricing solutions for different valuation patterns are essentially different. For the peak valuation patterns, the requests concentrate in a short period of time (10:00-12:00). In this case, almost all companies would like to use the facility during 10:00-12:00, creating an intense competition. Hence, the prices charged from the assigned requests tend to be higher during this period. On the other hand, the values of the requests are generally lower outside this period, as they are not assigned at their preferred time instants. For the uniform valuation patterns, the requests spread uniformly over a long period of time (7:00-15:00). As a result, the prices charged from the requests do not change much with the time, and the values of the requests are also distributed uniformly over the entire time horizon. Overall, the utilities of requests in the uniform demand pattern are higher than the utilities in the peak demand pattern. This shows that the concentrated time preference of the companies deteriorates the performance of the system.

6.5.2. Aggregated instance analysis. Figure 7 shows the empirical probability density function (PDF) of the assigned value, the price and the utility for different valuation patterns, number of parking spots, and parking durations. The empirical probability density distributions are derived from all the prices in all the instances in each case, as the distributions are the same for each instance in the same case. A quantitative comparison with the average values can be found in Table 4.

Solid lines correspond to 3 parking spots, while dashed lines correspond to 6 parking spots. Recall that we have the same average demand per parking spot despite the number of parking spots. Comparing the dash versus solid lines in all the graphs in Figure 7, we observe that the number of spots does not influence the empirical probability distribution as long as the average demand per parking spot is the same. This holds for the prices, the assigned values and the utilities.

This is expected, since the valuation per parking spot determines the distribution of requests over the time horizon and the competition intensity of the requests. Hence, the allocation and pricing solutions are also expected to be similar. It can also be observed from Fig. 7a and Fig.7c that there are always peaks at 10, 15 and 20 for both utilities and values, especially at parking duration of 10 min. This is because the maximum of the valuation function $v_{\max}$ is either 10, 15 or 20.

Comparing the different valuation patterns, we can see from Figure 7c that the uniform valuation pattern always leads to a higher utility than the peak valuation pattern, as the empirical PDFs of the uniform valuation pattern concentrates more to the right (higher utilities). The same property can be observed for the assigned values in Figure 7a, where companies receive higher assigned values in the scenarios with a uniform valuation pattern. This is expected, as the competition for the time instants is more intense for the peak demand pattern. Therefore, the requests assigned outside the preferred time period receive low values, and the requests assigned in the preferred time period suffer from high prices, as they have more externalities than in the scenarios with a uniform valuation pattern. This confirms that the concentrated requests tend to make the solution worse, as already seen in Section 6.5.1. However, for the prices in Figure 7b, there is no clear relation between the two valuation patterns. For the parking duration of 10 min and 20 min, the uniform valuation pattern renders a lower price than the peak demand pattern. However, for the parking duration of 30 min, the price for the uniform demand valuation is higher than for the peak demand pattern. This can be explained due to the impact of parking duration, which is detailed in the next paragraph. The above observation is consistent with the results in Table 4.

The impact of parking duration can also be observed in Figure 7. Figure 7a shows that all empirical PDFs of values move to the left as the parking duration increases. This is because a higher parking duration makes the instances more saturated. Hence, with the increase of the parking duration, less requests can be allocated to the same time instant. It can also be observed from Table 4 that the average assigned values decrease with the increase of the parking duration. The same applies for the utility in Figure 7c. The increase of the parking duration reduces the utilities of the companies. In Figure 7b, the price distribution moves to the right (i.e. the price increases) with the increase of the parking duration for the uniform valuation pattern. However, for the peak valuation pattern, the prices first increase, and then decrease. For short parking durations, especially in scenarios with the uniform valuation, the instances are not very saturated. Therefore, most requests can be assigned at preferred time instants. Hence, the externality of each request is generally small. As the parking duration increases, the system becomes saturated. Then there are two cases. In the first case, the requests can still be assigned at preferred time instants, but they have more impact on the other requests. In this case, the assigned values of the requests are still quite high, and the externalities, i.e. the prices, increase. This corresponds to the scenarios with the uniform valuation pattern and the parking duration of 20 min and 30 min, and the case with the peak valuation pattern and the parking duration of 20 min. In the second case, most of the requests cannot be assigned at preferred time instants. Therefore, the assigned values of most of the requests are quite low. In this very saturated case, the oversaturation is caused by the joint influence of many requests. Hence, the externalities of any request cannot be high, as the removal of a single request cannot impose much improvement on the solution. This corresponds to the scenario with the peak valuation pattern and the parking duration of 30 min.

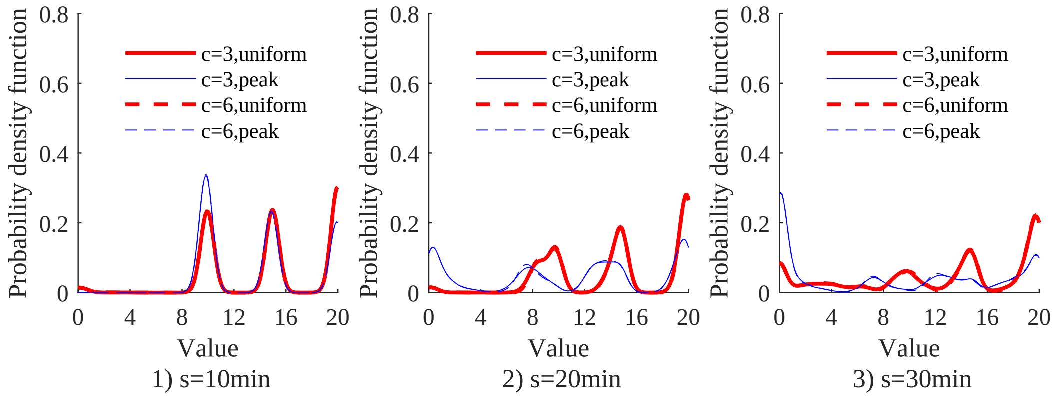
Table 4 **Average assigned values, prices and utilities for all set of instances**

	Value				Price				Utility			
Parking Spots	c=3		c=6		c=3		c=6		c=3		c=6	
Valuation pattern	U	P	U	P	U	P	U	P	U	P	U	P
s = 10 min	9.95	9.34	9.94	9.29	0.14	2.72	0.16	2.51	9.82	6.62	9.78	6.78
s = 20 min	9.57	6.13	9.57	6.33	1.09	4.65	1.05	4.44	8.49	1.47	8.52	1.89
s = 30 min	7.89	4.45	7.97	4.76	6.48	3.51	6.47	3.31	1.41	0.95	1.51	1.45

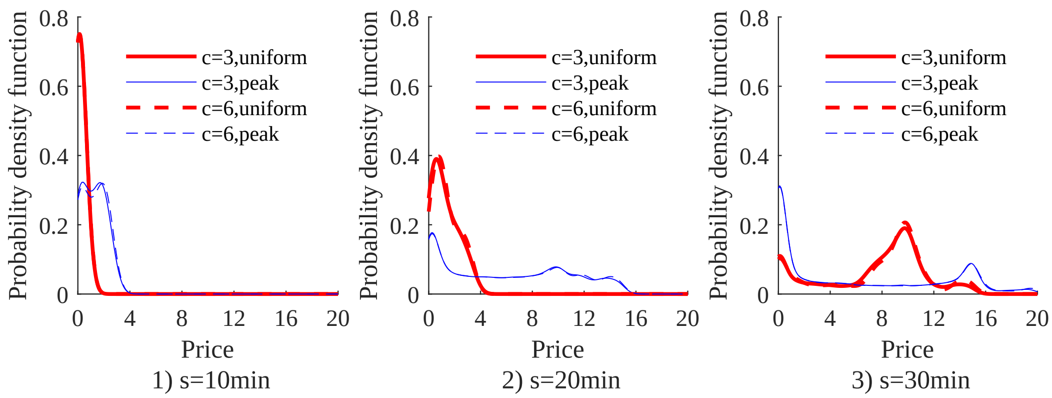
7. Conclusions

In this paper we proposed a new approach for the optimal management of urban loading/unloading facilities for logistic carriers. The approach, which is based on Combinatorial Auctions, optimizes the use of prebooked facilities with the goal to maximize the social welfare of the system. The new formulation of the problem enhances participation, provides a fair system, and guarantees that the carriers will express their true valuations. We proved that the flexibility of the new approach allows carriers to use any valuation function to express their time preferences over the offered spots, and it is valid to generalize a previous approach.

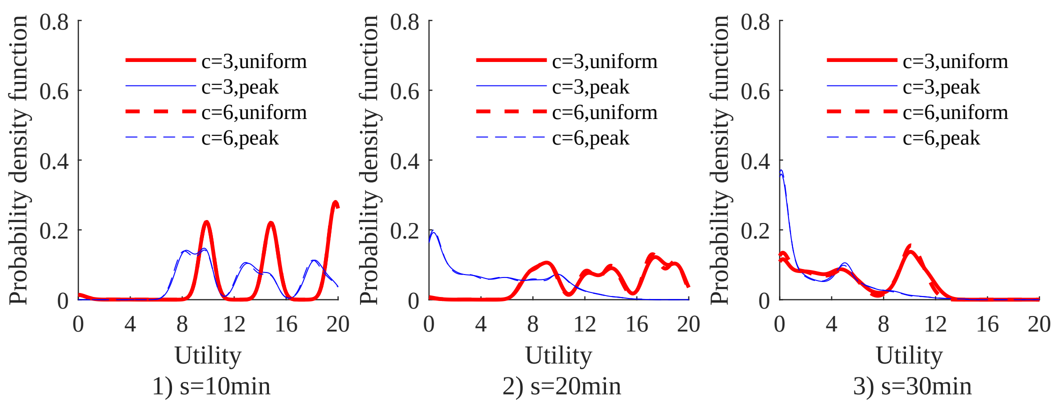
We developed different improvements to the standard solution algorithm, including the provision of an initial solution and the use of lazy constraints for the capacity constraints. The formulation and the different solution algorithms were tested in an extended set of test instances. The sensitivity to the level of the time discretization was specifically studied, and the results showed that using a 2 minute interval provides enough precision while reducing significantly the computation time. Regarding the different solution algorithm improvements, the results showed that the combined approached with an initial solution and the use of lazy constraints for the capacity constraints provides the optimal solution the fastest in most scenarios. Finally, the features of the solutions were also analyzed, individually and at the system level. The solutions for the problems where demand is uniformly distributed over the time horizon have better properties (higher valuation and utilities, and lower prices) than the ones where demand is concentrated in time. In any case, the optimal solution provides the maximum social benefit. This paper shows the benefits of using combinatorial auctions to optimally manage the prebooked urban loading/unloading facilities, that may become very popular with the introduction of technological and communication systems to improve the efficiency in urban areas.



(a) Aggregated probability distribution function of values



(b) Aggregated probability distribution function of prices



(c) Aggregated probability distribution function of utilities

Figure 7 Probability density distribution of the assigned value, the price, and the utility

Acknowledgements

This work was supported by the Hasler Foundation with the CALog project.

Appendix A: Comparison of formulations: Valuation functions

A.1. Binary valuation

Binary valuation is the simplest form to value different time instants, and can be represented as follows.

$$v_{ij} = \begin{cases} v_{\max}, & t_j \in [a_i, d_i] \\ 0, & t_j \notin [a_i, d_i] \end{cases} \tag{18}$$

where $v_{\max} > 0$ represents the maximum valuation.

In this case, companies only give two possible valuations across the whole time horizon. The company has a positive valuation during the desired time window and zero otherwise. An illustrative example is shown in Figure 3a.

It is easy to see that this valuation is equivalent to MOD4 in (Roca-Riu et al. 2015), where the number of requests scheduled outside the time window is minimized. When the assigned starting time is within the time window, no penalties are incurred, and the maximum valuation is achieved by the company. When the assigned starting time is outside the time window, the company suffers from a fixed penalty and has a zero valuation.

A.2. Piecewise trapezoid valuation

We consider a piecewise trapezoid valuation as a more complex valuation (see Figure 3b).

$$v_{ij} = \begin{cases} v_{\max} - \alpha(a_i - t_j), & t_j < a_i,\ j \in J_r \\ v_{\max}, & a_i \le t_j \le d_i,\ j \in J_r \\ v_{\max} - \alpha(t_j - d_i), & t_j > d_i,\ j \in J_r \end{cases} \tag{19}$$

where $v_{\max}$ is the maximum valuation and α is a factor ensuring $v_{ij} \ge 0,\ i \in N,\ t_j \in \Pi_r$.

Companies with such valuation functions have a preferred starting time inside the time window $[a_i, d_i]$ where valuation is maximum. Then, outside this time window the valuation decreases linearly. This valuation is equivalent to earliness/tardiness minimization (MOD1 in (Roca-Riu et al. 2015)) from the penalty formulation. Inside the time window, the maximum valuation is equivalent to zero penalties. When the starting time happens outside the window, minimizing the penalty incurred is equivalent to maximizing the corresponding valuations.

In this case, we need to guarantee that $v_{\max}$ is large enough so that the valuation is still positive with the highest possible penalty, i.e. $v_{ij} > 0$ for the given values of $v_{\max}$ and α. The earliness/tardiness minimizes the time deviation between the starting time and the desired time windows. For that reason, if we fix $v_{\max} = T$, i.e. equal to the time horizon, we guarantee that the valuation will still remain positive. The maximum time deviation cannot be higher than the time horizon.

A.3. Piecewise truncated trapezoid valuation

In this case, the valuation is exactly as in the previous section except that valuation is only positive inside the interval $[a_i - q, d_i + q]$, and drops to zero outside (see Figure 3c). This reflects the behaviour of some companies that can have a preferred interval, and also a tolerance valuation close to that interval, but their valuation is not positive after a given time distance from the desired interval. This valuation is equivalent to the penalites formulation of the earliness tardiness minimization subject to maximum displacement (MOD3 in (Roca-Riu et al. 2015)). In particular, the maximum displacement q is equivalent to the distance from the interval where the company valuation drops to zero. In this case, the maximum penalty is equal to q, and for that reason $v_{\max} = q$ is enough to guarantee positive valuations.

Appendix B: Experimental comparison of formulations

Table 5 Experimental comparison of formulations

		Valuations								
		Trapezoid			Truncated Trapezoid			Binary		
Inst.	n	$\mathcal{U}(x,v)$	$\Phi(x,\phi)$	(16)	$\mathcal{U}(x,v)$	$\Phi(x,\phi)$	(16)	$\mathcal{U}(x,v)$	$\Phi(x,\phi)$	(16)
1	111	11044.9	551	0	11044.9	551	0	100	11	0
2	75	7404.9	951	0	7400.1	999	0	63	12	0
3	60	5993.6	64	0	5993.6	64	0	58	2	0
4	58	5800.0	0	0	5800.0	0	0	58	0	0
5	119	11743.5	1565	0	11731.4	1686	0	98	21	0
6	82	8062.0	1380	0	8050.6	1494	0	67	15	0
7	92	9181.9	181	0	9181.7	183	0	89	3	0
8	42	4200.0	0	0	4200.0	0	0	42	0	0
9	48	4800.0	0	0	4800.0	0	0	48	0	0
10	73	7300.0	0	0	7300.0	0	0	73	0	0
11	66	6600.0	0	0	6600.0	0	0	66	0	0
12	74	7320.6	794	0	7320.3	797	0	62	12	0
13	64	6400.0	0	0	6400.0	0	0	64	0	0
14	36	3600.0	0	0	3600.0	0	0	36	0	0
15	24	2398.7	13	0	2398.7	13	0	23	1	0
16	87	8691.3	87	0	8691.3	87	0	84	3	0
17	104	10400.0	0	0	10400.0	0	0	104	0	0
18	146	14515.8	842	0	14514.8	852	0	131	15	0
19	76	7600.0	0	0	7600.0	0	0	76	0	0
20	30	3000.0	0	0	3000.0	0	0	30	0	0
21	69	6900.0	0	0	6900.0	0	0	69	0	0
22	76	7600.0	0	0	7600.0	0	0	76	0	0
23	136	13556.8	432	0	13556.8	432	0	127	9	0
24	54	5357.8	422	0	5356.8	432	0	47	7	0
25	57	5700.0	0	0	5700.0	0	0	57	0	0
26	82	8144.7	553	0	8144.3	557	0	71	11	0
27	75	7500.0	0	0	7500.0	0	0	75	0	0
28	45	4499.1	9	0	4499.1	9	0	44	1	0
29	98	9800.0	0	0	9800.0	0	0	98	0	0
30	21	2099.3	7	0	2099.3	7	0	20	1	0
31	72	7200.0	0	0	7200.0	0	0	72	0	0
32	109	10857.3	427	0	10857.0	430	0	98	11	0
33	50	4970.6	294	0	4970.1	299	0	44	6	0
34	106	10600.0	0	0	10600.0	0	0	106	0	0
35	129	12857.8	422	0	12857.8	422	0	118	11	0
36	62	6138.9	611	0	6138.7	613	0	48	14	0
37	51	5100.0	0	0	5100.0	0	0	51	0	0
38	137	13693.8	62	0	13693.8	62	0	134	3	0
39	97	9588.2	1118	0	9583.5	1165	0	79	18	0
40	135	13428.5	715	0	13428.5	715	0	120	15	0
41	55	5467	330	0	5465.6	344	0	48	7	0
42	170	16977.6	224	0	16977.6	224	0	164	6	0
43	100	9984.7	153	0	9984.7	153	0	96	4	0
44	220	21826.7	1733	0	21817.8	1822	0	190	30	0
45	237	23566.2	1338	0	23565.5	1345	0	212	25	0
46	77	7700.0	0	0	7700.0	0	0	77	0	0
47	46	4600.0	0	0	4600.0	0	0	46	0	0
48	22	2200.0	0	0	2200.0	0	0	22	0	0
49	23	2300.0	0	0	2300.0	0	0	23	0	0
50	32	3200.0	0	0	3200.0	0	0	32	0	0
51	31	3100.0	0	0	3100.0	0	0	31	0	0
52	29	2900.0	0	0	2900.0	0	0	29	0	0
53	23	2300.0	0	0	2300.0	0	0	23	0	0
54	60	6000.0	0	0	6000.0	0	0	60	0	0
55	41	4100.0	0	0	4100.0	0	0	41	0	0
56	116	11600.0	0	0	11600.0	0	0	116	0	0
57	103	10300.0	0	0	10300.0	0	0	103	0	0
58	66	6600.0	0	0	6600.0	0	0	66	0	0
59	73	7300.0	0	0	7300.0	0	0	73	0	0
60	151	15100.0	0	0	15100.0	0	0	151	0	0

References

Ayala, D., O. Wolfson, B. Xu, B. Dasgupta, J. Lin. 2011. Parking slot assignment games. *Proceedings of the 19th International Conference on Advances in Geographic Information Systems*. ACM, 299–308.

Ayala, D., O. Wolfson, B. Xu, B. Dasupta, . Lin. 2012. Pricing of parking for congestion reduction. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 43–51.

Barcelona City Council. 2014. Urban mobility plan of Barcelona 2013-2018 .

Beziat, A. 2015. Parking for freight vehicles in dense urban centers: The issue of delivery areas in paris. *Transportation Research Board 94th Annual Meeting*. 15-2078.

Carlino, D., S. D. Boyles, P. Stone. 2013. Auction-based autonomous intersection management. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 529–534.

Chen, Z., Y. Yin, F. He, J. L. Lin. 2015. Parking reservation for managing downtown curbside parking. *Transportation Research Record: Journal of the Transportation Research Board* (2498) 12–18.

Dablanc, L., A. Beziat. 2015. Parking for freight vehicles in dense urban centers - The issue of delivery areas in Paris. *Metro Freight Volvo Center of Excellence* 1–16.

David, B., F. Armetta. 2013. Logistique et transport de marchandises, les aires de livraison du futur. *Le Temps du Bilan* .

Du, L., S. Gong. 2016. Stochastic poisson game for an online decentralized and coordinated parking mechanism. *Transportation Research Part B: Methodological* **87** 44–63.

Figliozzi, M. A. 2007. Analysis of the efficiency of urban commercial vehicle tours: Data collection, methodology, and policy implications. *Transportation Research Part B: Methodological* **41**(9) 1014–1032.

Geng, Y., C. G. Cassandras. 2011. Dynamic resource allocation in urban settings: A smart parking approach. *IEEE International Symposium on Computer-Aided Control System Design (CACSD)*. IEEE, 1–6.

Guo, W., Y. Zhang, M. Xu, Z. Zhang, L. Li. 2016. Parking spaces repurchase strategy design via simulation optimization. *Journal of Intelligent Transportation Systems* **20**(3) 255–269.

Hashimoto, S., R. Kanamori, T. Ito. 2013. Auction-based parking reservation system with electricity trading. *15th IEEE Conference on Business Informatics (CBI)*. IEEE, 33–40.

Laporte, G. 1992. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* **59**(2) 231 – 247.

Lehmann, D., L. I. Oćallaghan, Y. Shoham. 2002. Truth revelation in approximately efficient combinatorial auctions. *J. ACM* **49**(5) 577–602.

Lei, C., Y. Ouyang. 2017. Dynamic pricing and reservation for intelligent urban parking management. *Transportation Research Part C: Emerging Technologies* **77** 226–244.

Ma, L. 2001. Analysis of unloading goods in urban streets and required data. *City Logistics II* 367–379.

Mackowski, D., Y. Bai, Y. Ouyang. 2015. Parking space management via dynamic performance-based pricing. *Transportation Research Procedia* **7** 170–191.

Munuzuri, J., J. Larraneta, J. Ibanez, G. Montero. 2006. Pilot demonstration of web-based loading zone reservation system. *The 4th International Conference on City Logistics, Recent Advances in City Logistics.*.

Nisan, N., T. Roughgarden, E. Tardos, V. V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA.

Roca-Riu, M. 2015. Improving urban deliveries via collaboration. Ph.D. thesis, Barcelona-Tech, UPC.

Roca-Riu, M., J. Cao, I. Dakic, M. Menéndez. 2017. Designing dynamic delivery parking sports in urban areas to reduce traffic disruptions. *Journal of Advanced Transportation* **In press**.

Roca-Riu, M., E. Fernández, M. Estrada. 2015. Parking slot assignment for urban distribution: Models and formulations. *Omega* 157 – 175.

Shao, C., H. Yang, Y. Zhang, J. Ke. 2016. A simple reservation and allocation model of shared parking lots. *Transportation Research Part C: Emerging Technologies* **71** 303–312.

Su, P., B. B. Park. 2015. Auction-based highway reservation system an agent-based simulation study. *Transportation Research Part C: Emerging Technologies* **60** 211–226.

Zou, B., N. Kafle, O. Wolfson, J. Lin. 2015. A mechanism design based approach to solving parking slot assignment in the information era. *Transportation Research Part B: Methodological* **81**(Part 2) 631 – 653.