

Parking place demand and offer assignment

Abstract

This paper proposes a novel binary optimization model to address the parking place demand and offer assignment problem. Although the problem has some similarities with timetabling and assignment problems, some particular extensions have been made. A new formulation for consecutiveness constraints is proposed. This new model takes into account in a multiobjective setting the preferences of the users to park in an already known parking place, the distance between demanded and offered parking places and the number of total assignments made. The model has a daily scope to be executed on an hourly basis, with the possibility to modify the assignments already made for those clients that have not yet confirmed their assignment. Using randomly generated parameters, several large-scale case studies are presented that show the strength of the current approach.

1. Introduction

Nowadays, the authorities are more concerned than ever about the greenhouse effect gases, and transportation is one of the major contributors to these gases (see [1]). Several studies have appeared to look for solutions to this problem, most of them related with parking management. In [2], the authors propose an algorithm to optimize the location of a public parking. Marshall shows in [3] that there are too many parking in small cities, and very often underused. He proposes to switch from the conventional town-planning policies to a mixed-use activity center. That means that residential areas and commercial ones should not be separated as they use to be. That way, the parking are used during the whole day, instead of using it just in the day for commercial zones and the nights in the residential ones. Knoflachner, [1], suggests that public transportation must be encouraged and private car discouraged, by making parking places less accessible than public transport stops. Other works focuses on public transportation improvement, such as [4] in Tagus Valley (Portugal) and [5] in Vilnius (Lithuania), showing the increasing interest in all these related matters.

Not only are the authorities concerned with this problem, but also users and companies are more and more aware, making new parking solutions emerge. In [6] it is shown how some companies have developed a system of car pooling to reduce the number of cars traveling from or to the company site, proposing a solution to match cars and destinations. Other users seem willing to share their cars to split trip costs. In [7], the authors present an integrated system for the organization of a car pooling service, where the specific routing problem is solved heuristically.

One of the latest alternatives involves renting one's parking place when it is not used, for example while being at work. This could reduce parking problem, as well as decrease the number of cars running during rush hours, since time to find parking places would be dramatically reduced. Although there is not yet any reference in the

technical literature, a system with these features has already been put in operation in several Spanish cities (see [8]), using a previous version of the solution proposed here.

This paper proposes a novel formulation to solve the problem of parking place assignment that can be considered as a kind of assignment or scheduling problems. Assignment problems consist in assigning people to jobs, matching one to one. Different variations of assignment problems can be found in [9]. Toroslu proposed in [10] two kinds of job assignment problems of: satisfying all the job demands, verifying as many as possible or satisfying the constraints, maximizing the number of jobs finished. The latter is similar to the parking place assignment problem. Timetabling problems, used to match participants with each other and with the resources available, are equivalent to the cars and parking places. Classical timetabling problems include employee, university, exams, lectures or sports timetabling, see [11] and [12] for real-world applications.

Problem description

Sharing parking places involves demand and offer assignment. The assignment must consider parameters such as distance between car park offers and demand addresses, parking place and car dimensions, maximum matching time requirements, etc.

In addition there are several considerations to be taken into account. Firstly, a user may prefer to park in a known place. Therefore, if a person parks every day in the same parking place, despite the availability of a free place nearer his final destination, the user may probably prefer the same customary place. However, if the usual parking place is unavailable, then the client could use the next nearer available place. Frequent use of this new place could make the client to become used to it, not wanting to go back to the old one.

Secondly, cars or other types of vehicles (such as motorbikes or vans) can only be parked in places larger than its own size.

Parking a car means keeping the car in the same parking place for the whole period of time demanded. Of course, once the period ends, and if the offer is still available, the same place can be assigned to a different demand. Therefore, different vehicles can be assigned to the same parking place in the same day if the assignments do not overlap.

Clients should be allowed to limit their acceptable walking distance from the potential assignment to the requested location, since it may be better not to assign a place than to provide it too far away. The proposed algorithm tries to minimize this walking distance, and its bound is considered as a hard constraint.

Finally, it must be taken into account that offers and demands vary dynamically with time. Indeed at any time new users may access the system to find or to offer new parking places. In addition, existing clients may change their needs along the week. For example, from Monday to Thursday, a client may demand a place from 9 to 18 h, while

on Fridays from 8 to 15 h, and nothing on Saturdays and Sundays. This means that the assignments must be computed in real time many times per day.

In fact the model must be able to be run at least once per hour in order to make the algorithm viable in a real environment. The natural rolling time scope should be 24 hours, due to the daily cycle of the demand requirements. Any time before the assignments take place, the affected clients (demands and offers subject to potential assignment) are asked to confirm their previous demand or offer. This is a key point of the system, since theoretically available places or demands can be discarded in real time when no explicit confirmation is obtained. This allows occasional routine modification, such as for example when a sick client doesn't use the car.

In the sequel the paper analyses analogous job assignment problems, and describes the mathematical formulation of the parking place sharing algorithm proposed. In section IV some case studies are presented and their results analyzed to confirm the validity of the proposed approach. Finally, some conclusions are extracted and further developments are suggested.

Mathematical model

As has already been seen, the problem of parking place assignment is related to some other classical assignment or scheduling problems such as assignment problem or timetabling.

These problems have a combinatorial nature and the number of constraints and variables increases exponentially as in the parking place assignment problem. However, the current problem has some additional characteristics (which will be defined in the following mathematical formulation) that make it different. The mathematical formulation of the *job assignment problem*, as explained in [13], is:

$$\begin{aligned} \min & \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \\ \sum_{i=1}^n x_{ij} &= 1 \quad \forall j \\ \sum_{j=1}^n x_{ij} &= 1 \quad \forall i \end{aligned}$$

where i are jobs and j persons, and n the total number of jobs and persons, which in this model coincide. C_{ij} represents the assignment cost of job i to person j . The job assignment is a variable defined as

$$x_{ij} = \begin{cases} 1 & \text{job } i \text{ assigned to person } j \\ 0 & \text{otherwise} \end{cases}$$

The objective function minimizes the total assignment cost. The first constraint guaranties that each person performs one and only one job, and the second one that each job is done by one and only one person. This problem is a particular case of the

transportation problem, which has an optimal solution with binary variables. This problem is also called Linear Assignment Problem (LAP), [14].

The parking place assignment problem extends the assignment problem across time. The time scope of the model spans over the next 24 hours with a time unit of 1 hour. It may also be possible to consider a shorter time window with a time unit of 30 minutes or a 2-day scope with a 2-hour time unit. The assignment variable can be redefined to include a new dimension k , corresponding to hours:

$$x_{ijk} = \begin{cases} 1 & \text{demand } i \text{ supplied with offer } j \text{ in hour } k \\ 0 & \text{otherwise} \end{cases}$$

where demands are cars to be parked, and offers are the available parking places.

Another important modification is the introduction of a multiobjective function, see [15], that directs the problem towards more realistic solutions. On the one hand it is important to maximize the number of attained assignments as a primary objective of a matchmaker company. On the other hand a second objective is client satisfaction that is achieved by minimizing the distance between each demand and offer, and maximizing also the assignment familiarity, which is 1 when the same place has already been assigned to the same demand in the recent past, and 0 otherwise:

$$F_{ij} = \begin{cases} 1 & \text{demand } i \text{ supplied with offer } j \text{ in previous days} \\ 0 & \text{otherwise} \end{cases}$$

[16] introduces the concepts of priority and seniority, to prioritize some persons over others. Although it is similar to the familiarity, the latter is a dynamic attribute, since it depends on the previously offers and demands matched. In addition, familiarity is not a constraint but a term in the objective function, which gives more flexibility to the solution. In addition, priorities can be assigned to clients based on a price discrimination scheme. The more they pay the better chances to be assigned.

Both distance and familiarity have been normalized. The distance has been normalized by dividing it by the maximum acceptable distance D established by each client. Familiarity has been normalized by dividing by the number of days taken into consideration. Since the algorithm has been conceived to maximize the number of assignments, normalized distance and familiarity have been weighted by parameters α and β . These weights must be small enough to make both factors negligible with respect to the number of assignments, so that these objectives are only used to select from solutions having the same number of assignments. Nevertheless, another cases giving priority to client satisfaction are compatible with the model proposed.

The parking place assignment problem can now be stated as:

$$\begin{aligned}
& \max \sum_{ij} x'_{ij} (1 - \alpha D_{ij} + \beta F_{ij}) \\
& \sum_i x_{ijk} \leq 1 \quad \forall jk \\
& \sum_j x_{ijk} \leq 1 \quad \forall ik \\
& \sum_k x_{ijk} = H_i x'_{ij} \quad \forall ij \\
& y_{ijk} = z_{ijk+H_i} \quad \forall ijk \\
& x_{ijk-1} - x_{ijk} + y_{ijk} - z_{ijk} = 0 \quad \forall ijk \\
& \sum_k (y_{ijk} + z_{ijk}) \leq 2 \quad \forall ij \\
& x'_{ij}, x_{ijk} \in \{0,1\}, ijk \in \Phi \\
& 0 \leq y_{ijk}, z_{ijk} \leq 1
\end{aligned}$$

where parameter D_{ij} is the normalized distance between the demand i and offer j , and H_i is the duration of the parking demand. x'_{ij} defines the assignment of a demand to an offer at any time

$$x'_{ij} = \begin{cases} 1 & \text{demand } i \text{ assigned to offer } j \text{ in any hour} \\ 0 & \text{otherwise} \end{cases}$$

The first two constraints correspond to non simultaneousness hypothesis. A car is parked at most in only one place and a place is used by at most only one car at any given hour. The third constraint represents the relation between the hourly assignment variable x_{ijk} and the parking status of the car x'_{ij} . A car is considered parked if and only if the car remains stopped at the same parking place during all the demanded hours H_i . The new variables y_{ijk} and z_{ijk} correspond to the beginning and end of the assignment.

In the last three constraints we propose a new formulation for establishing the consecutiveness of the parking hours by introducing the variables y_{ijk} and z_{ijk} for detecting the beginning and end of the assignment, respectively.

$$\begin{aligned}
& y_{ijk} = z_{ijk+H_i} \quad \forall ijk \\
& x_{ijk-1} - x_{ijk} + y_{ijk} - z_{ijk} = 0 \quad \forall ijk \\
& \sum_k (y_{ijk} + z_{ijk}) \leq 2 \quad \forall ij
\end{aligned}$$

The first one expresses that the end of the parking time occurs H_i hours after its beginning. The second one states the relation between hourly parking assignment variables and the beginning and ending variables. Observing the second equation, if the car is not parked at hour $k-1$, $x_{ijk-1} = 0$, and is parked at hour k , $x_{ijk} = 1$, then the parking period begins at k and necessarily $y_{ijk} = 1$ and $z_{ijk} = 0$. On the contrary, if the car is parked at hour $k-1$, $x_{ijk-1} = 1$, and is not parked at hour k , $x_{ijk} = 0$, then the parking period ends at k , and automatically $z_{ijk} = 1$ and $y_{ijk} = 0$. The last equation establishes that at most one beginning and one end can be set.

An additional condition $ijk \in \Phi$ is introduced by allowing the tuple ijk to be set only when parking place is available, a car is demanding it, the car fits into the parking place, and any other convenient condition holds such as the maximum allowed walking distance D .

As these constraints are the most difficult to solve, an alternative formulation for the consecutiveness constraints has been tested, suppressing the variables Z_{ijk} corresponding to the end of the assignment

$$\begin{aligned} \sum_{k'=k}^{k+H_i-1} x_{ijk'} &\geq H_i y_{ijk} & \forall ijk \\ x_{ijk-1} - x_{ijk} + y_{ijk} &\leq 0 & \forall ijk \\ \sum_k y_{ijk} &\leq 1 & \forall ij \end{aligned}$$

The first constraint states that if the parking period begins at hour k , $y_{ijk}=1$ and the last H_i x_{ijk} variables have to be equal to 1. The second one is similar to the corresponding one in the earlier formulation but without using the variable Z_{ijk} , corresponding to the end of the assignment. It changes from an equality to an inequality, eliminating Z_{ijk} . The third one says that at most one beginning of assignment is allowed. This leads to the following complete second formulation:

$$\begin{aligned} \max \sum_{ij} x'_{ij} (1 - \alpha D_{ij} + \beta F_{ij}) \\ \sum_i x_{ijk} &\leq 1 & \forall jk \\ \sum_j x_{ijk} &\leq 1 & \forall ik \\ \sum_k x_{ijk} &= H_i x'_{ij} & \forall ij \\ \sum_{k'=k}^{k+H_i-1} x_{ijk'} &\geq H_i y_{ijk} & \forall ijk \\ x_{ijk-1} - x_{ijk} + y_{ijk} &\leq 0 & \forall ijk \\ \sum_k y_{ijk} &\leq 1 & \forall ij \\ x'_{ij}, x_{ijk} &\in \{0,1\}, ijk \in \Phi \\ 0 &\leq y_{ijk} \leq 1 \end{aligned}$$

The number of constraints and binary variables of both formulations is shown in table XX, being the capital letters the cardinal of the respective sets. However, the number of continuous variables decreases substantially in the second formulation, and therefore the number of non zero elements.

	Formulation 1	Formulation 2
Constraints	$2IJK + 2IJ + (I + J)K$	$2IJK + 2IJ + (I + J)K$
Continuous Variables	$2IJK$	IJK
Binary Variables	$IJK + IJ$	$IJK + IJ$

If we suppose that there are $I = 1000$ cars and $J = 1000$ parking places to manage during $K = 24$ hours, it can be easily observed that the problem becomes very large, as it will be shown in the case study.

Case Study

To check the validity of the proposed model, as well as its solution time, nine examples have been worked out. The examples have different parameters values to show their influence in the final solution. The first problem is composed of 200 offers and demands, scattered in a large area (see 1), and the number of assignments is prioritized compared to familiarity or distance. The second and third examples have nearer offers and demands, but they differ in the maximum allowed walking distance. The fourth and fifth are examples similar to the second and third ones (see 2), but the offers and demands are nearer than the first example. The next two examples uses 500 offers and demands differing only in the maximum allowed walking distance, as in the previous cases. The last examples use 1000 offers and demands, giving a considerable weight to the familiarity in the last case. The parameters of all these examples are summarized in Table 1.

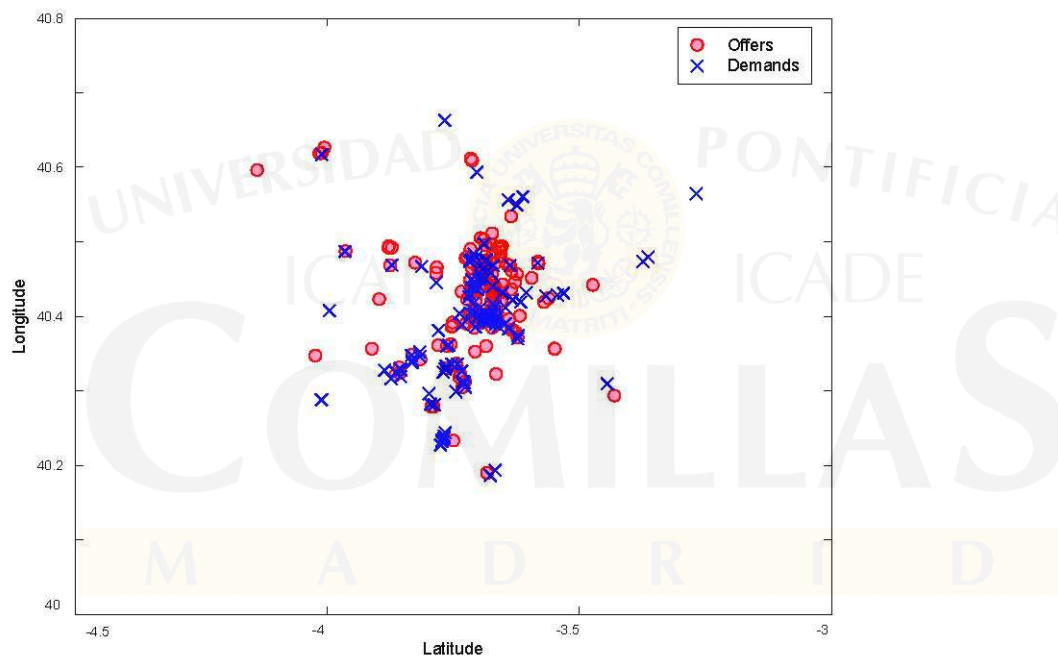


Figure 1 Demands (red circles) and offers (blue cross) of C1

Parameter 'size' corresponds to the number of offers and demands. For simplicity, in these cases the number of demands and offers were the same. The parameters α and β are used to weight distance and familiarity depending on their values. Current cases were focused on maximizing the number of assignments, and therefore small values were used for these parameters. These parameters reflect the decision priorities and therefore may be changed depending on the needs of the application.

Recently, the authors have been developing a system to allow people sharing their parking places based on mobile phone communications with a startup company, see [8], with a similar version of the proposed algorithm. The data for the case study have been taken from real data of this Spanish parking place board website and complemented with random values. Geographical locations correspond to real offers and demands of parking places, while familiarity data have been randomly created.

Distances correspond to real cases. The familiarity matrix takes into account just the last day. That means that each row (demand) has only one 1 corresponding to the parking place where the car was parked the day before. In commercial operation, familiarity should be computed as the time ratio that the car is parked in each place.

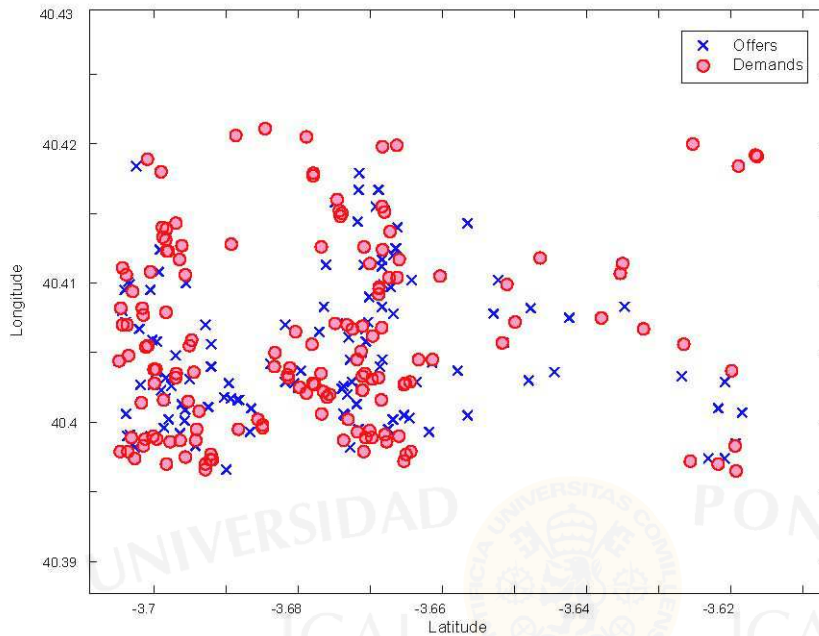


Figure 2 Demands and offers for C4 and C5

	size	α	β	$\bar{D}$ [m]
C1	200	1/500	1/10000	200000
C2	200	1/50	1/1000	1000
C3	200	1/50	1/1000	2000
C4	200	1/50	1/1000	1000
C5	200	1/50	1/1000	2000
C6	500	1/50	1/1000	1000
C7	500	1/50	1/1000	2000
C8	1000	1/50	1/1000	2000
C9	1000	1/50	1/5	2000

Table 1 Case study data

Data corresponding to the sizes of cars and places were randomly generated. The offering and demanding hours for each car and parking place have also been generated randomly, but coherently with everyday real situations. Each offer and demand must correspond to one of the following type of parking demand/offer:

- Night time
- Day time (morning + afternoon)
- Morning time
- Afternoon time

Analysis of the simulation results

The model has been coded in GAMS 22.9 and run in an Intel Xeon 2.33 GHz CPU with 8 cores, 8 GB RAM and Windows Server 2003 Enterprise x64 Edition. CPLEX 11.2 was used as MIP solver with dual simplex method.

Table 2 shows the number of assignments made, the weighted familiarity and distance and the objective function of the solution for each case example. As can be observed the number of assignments has a very important weight with respect to familiarity and distance. In addition, familiarity has a very low weight compared to distance, being 0 in cases C2 to C5. The familiarity matrix, being generated randomly because of the lack of real values, gives familiarity to places and cars that cannot be matched together. This forces the algorithm to choose parking places with no familiarity, resulting also in a low familiarity weight in the objective function. To show the importance of the familiarity, in the last example, the weight given to it has been considerably increased. That way, the familiarity becomes more important than the distance.

Tables 3 and 4 show the sizes of the corresponding optimization problems, expressed in R constraints, CV continuous variables, BV binary variables and E non zero elements. These examples show that the model is fast enough to be used in a production environment. Indeed every problem with a size lower than 1000 offers and demands can be solved in around 5 seconds. Furthermore, the solver is able to find the optimal solution. The second formulation requires much less computing time, as can be seen in all the tested cases, but especially in the last ones which are the largest ones.

	# of Assign.	Familiar.	Distance	O.F.
C1	136	0.001	-0.013	135.988
C2	150	0	-1.119	149.881
C3	170	0	-0.935	169.065
C4	151	0	-1.148	149.852
C5	168	0	-0.944	167.056
C6	286	0.001	-2.616	283.385
C7	391	0.001	-3.338	387.663
C8	875	0.002	-6.617	868.385
C9	875	1.6	-6.706	869.894

Table 2 results for each case

	R	CV	B	E	Time [s]
C1	61403	242505	38933	471453	1.4
C2	19454	71646	11066	136318	0.4

C3	37963	149021	23209	284520	0.9
C4	20703	78507	11399	145993	0.5
C5	41281	165893	24241	309961	0.9
C6	23064	69235	11247	137197	0.4
C7	54651	189641	30853	374225	1.2
C8	221962	813848	132532	1612658	5.2
C9	221962	813848	132532	1612658	5.8

Table 3 size for each case on formulation 1

As can be seen, cases C8 and C9 have the same values in time and sizes. The input for both cases is the same. The only difference is the value of β , the weight parameter of the familiarity. The table 2 shows that in the latter, the familiarity has increased considerably, to the detriment of the distance. The algorithm has chosen more familiar places although they are farther due to the importance of the familiarity in this case.

For problem sizes much bigger than those presented in the table a division by city districts can be attempted. Therefore, suboptimal solutions can be found in reasonable solution times. As the parking place assignment problem is solved for every hour for the next 24 hours a verification mechanism is introduced in real life for each client to explicitly confirm his/her demand or offer, so the problem is solved only for those daily confirmed requests.

	R	CV	B	E	Time [s]
C1	61403	140721	38933	425314	0.7
C2	19454	41358	11066	120063	0.2
C3	37963	86117	23209	251298	0.5
C4	20703	44955	11399	125496	0.3
C5	41281	95069	24241	267975	0.5
C6	23064	40243	11247	126393	0.3
C7	54651	110249	30853	343508	0.7
C8	221962	473192	132532	1485781	3.2
C9	221962	473192	132532	1485781	3.3

Table 4 size of each case on formulation 2

Conclusions

This paper proposes a new algorithm to solve the parking place assignment problem. It maximizes the number of assignments, minimizing the distance between the assigned place and the final destination requested. It also considers as additional criterion the maximization

of the familiarity of the client with a given place.

The model has been formulated as a MIP problem, coded in GAMS language and solved using CPLEX XX.XX. It can be executed on a rolling mode at least once per hour with a time scope of 24 h in advance, taking into account just the demands and offers previously confirmed by the clients.

The model finds the optimal solution in a very short time for all the real cases studied. A previous version has been implemented in a commercial environment [8], and is currently supplied with real data and continuously making assignments, proving its applicability in a real environment.

If the problem sizes become too big computer memory could limit worsening the solution process. In that case metaheuristic algorithms (such as tabu search, genetic algorithms, etc., see [17] and [18] for recent developments in using heuristic and evolutionary algorithms for timetabling problems), or other alternative approaches (such as constraint programming) could be used to obtain optimal or quasi-optimal solutions with less computer resources, or to cooperate with traditional methods to seek for optimal assignments.

Acknowledgment

Authors would like to thank to company *TuPlaza Movilidad y Desarrollo S.L.* for the valuable data provided.