A MIP model to optimize real time maintenance and relocation operations in one-way carsharing systems

Gonçalo Santos\textsuperscript{a*}, Gonçalo Correia\textsuperscript{b}

\textsuperscript{a}Department of Civil Engineering, University of Coimbra, Polo II, 3030-788 Coimbra, Portugal
\textsuperscript{b}Transport and Planning Department, TU Delft, Stevinweg 1, 2628 CN Delft, The Netherlands

Abstract

The daily management of a one way carsharing system is focused on relocation and maintenance operations. On one hand, the freedom of allowing users to return vehicles wherever they want leads to vehicle imbalance problems and on the other hand, a system where the same vehicle is used by different persons, has the need for daily maintenance operations. Demand is a key element that influences both relocation and maintenance needs, and the system has to be prepared to adapt and react to it in an optimized way. This paper describes the optimization model integrated in a tool developed to manage relocation and maintenance operations of a one-way carsharing system in real time. The optimization model developed is a mixed integer linear programming model, instant-based and designed to work using a rolling horizon. The division of the operation time into planning horizons allow the updating of the system status data, which provides a closer connection to the system reality. The model considers that crew elements use mainly the vehicles of the system to move inside the operating area, and orders are remotely transmitted through a wireless communication platform to mobile devices. Three types of activities are attributed to crew elements: waiting, maintaining the vehicles and moving. The mathematical model discriminates each crew element and is able to decide the best schedule for each one. It has flexibility to select between crew trip joining or crew trip splitting, in order to reduce movement cost or relocate a higher number of vehicles. Simulation tests were performed to assess the computer processing time for different problem sizes.

Keywords: carsharing, maintenance, relocation, optimization

* Corresponding author. Tel.: +0 351 239 797122; fax: +0 351 239 722012.
E-mail address: ggduartesantos@gmail.com
1. Introduction

Carsharing is a form of collaborative consumption consisting in the timesharing of vehicles, allowing its use for short periods of time, less than a day. Organizations promoting carsharing own a fleet of vehicles making it available for people to share (Millard-Ball et al., 2005). To gain access to the system, individuals must meet the membership requirements of the carsharing organization, which can include an annual fee. After the membership process is fulfilled, users can have immediate access to the available vehicles whenever they want, without needing to fill a rental agreement each time a car is used. Users pay a usage fee for time and distance travelled, which already embeds fuel and insurance.

In terms of carsharing operational characteristics, two main categories can be distinguished: location of vehicles and allowed movements.

Concerning the location of vehicles, the system can be station based (discrete) or free floating (continuous). In station based services, vehicles are located at pre-defined places. Those can be stations, parking lots or reserved street parking areas. Free floating services, are characterized by having its vehicles parked at any place, with legal public access, inside a pre-defined zone. Free floating system vehicles have on-board GPS equipment to ease management and allow users to locate them by using a smartphone (Shaheen et al., 2015).

Allowed movements can be subdivided in round trip and one way. In roundtrip services users need to return the vehicle to the same place where it has been picked up. In one way carsharing services, movements to another destination different than the origin are allowed, which means that there is no imposition to return the car to any particular place (Barth and Shaheen, 2002).

The simplest operational set up is round trip and station based. This is the choice of systems with a small number of vehicles and stations, since it is easy to manage and doesn’t require many staff hours, nonetheless it is not adapted to users’ needs. By increasing one step on the level of operational complexity, we have one way and station based systems. One way movements give more flexibility to users, being a critical factor to attract new clients to the system (Efthymiou et al., 2013). Additionally, it lets a higher utilization of vehicles as they do not need to be idle during the rental period as it happens when clients are forced to a roundtrip. The downside is that it can lead to having a surplus of vehicles in stations with high demand as destination, and a lack of vehicles in stations with high demand as origin, unbalancing the demand and supply quotient (Barth et al., 2004). The most complex operational set up is one way and free floating. This allows individuals to use a vehicle of the system as if it was their own vehicle. However, it doesn’t mean complete freedom, since vehicles need to be delivered inside an operating area (Shaheen et al., 2015).

The imbalance problems created by one way movements need to be solved by the operator to minimize the rejected demand and increase vehicle availability levels. This can be done by intervening on the demand side or on the supply side (Jorge and Correia, 2013). The amount of vehicle usage in a system with scale to be one way, implies the need for daily maintenance operations, such as vehicle cleaning and refueling. Therefore the use of staff to perform both maintenance and relocation operations should be regarded.

This paper describes a model developed to simultaneously optimize maintenance and relocation operations in an one way carsharing system that can be used for station based as well as free floating system settings. The model is instant-based and designed to work in a rolling horizon planning approach. It uses carsharing system vehicles to assist staff movements, as much as possible.

2. Literature review

The imbalance problem that arises by allowing one-way movements has been addressed in different ways. According to Jorge and Correia (2013), there are three main approaches to assist the daily system management operations: operator-based relocations, user-based relocations, and trip selection. In operator-based relocations, staff is used to periodically drive vehicles from a station with an excess of vehicles to a station with a shortage of vehicles. In user-based relocations, balancing movements are performed by clients reacting to incentive mechanisms, usually based on price. Trip selection consists in controlling the demand by allowing only the trips that are favorable to the balance of vehicle stocks.
Concerning to operator based relocations, most authors address the optimization of the movement of vehicles to balance the system and the movement of staff needed to drive them (e.g.: Barth and Todd, 1999; Wang et al., 2010; Nair and Miller-Hooks, 2011). Kek et al. (2009) goes a bit further by adding the maintenance to the driving tasks and, therefore, optimizing maintenance and relocation operations simultaneously. This is an innovation that meets the needs of operators to manage staff activity, making it ideal for running a carsharing system on a daily basis. Furthermore, it discriminates each member of staff and its activities using binary variables, giving the detailed data needed for employee scheduling. However, the optimization model is computationally cumbersome due to the large sets variables requested for running the model (Correia, 2009) and also lacks the flexibility to allow trip joining of staff, that is, the ability to form staff groups that share vehicles for the same movements. The last, is a feature considered by Barth et al., 2004 and Uesugi et al. 2007, for user-based relocation strategies, and can potentially have a positive impact if implemented to the referred model.

3. Methodology

A new optimization model is designed as an upgrade of the work of Kek et al. (2009). The optimization model is a mixed integer programming (MIP) model that uses a rolling horizon to reduce the uncertainty of forecasting and ease the number of decision variables of the model. A rolling horizon is a common decision making practice to provide decisions in a dynamic stochastic environment. It involves making the most immediate decisions, which need to be made for the beginning of a planning horizon, based on information and forecasts that look further into the future by considering the entire planning horizon (Sethi and Sorger, 1991). The MIP model can be used on a fixed interval rolling horizon or on a request triggered rolling horizon planning (Wang and Kopfer, 2013). In both approaches, the entire operation time is divided into a series of planning horizons. Each planning horizon has two parts: the fixed part and the changeable part. The fixed part consists of the activities already executed or in execution when the new optimized plan is provided. The changeable part consists of the not initialized activities, which are updated when the new optimized plan is provided. The advantage of using a rolling horizon approach is that the orders are updated at the start of each rolling horizon, using the latest information available. The data needed to run the MIP model for each planning horizon is obtained directly from the carsharing system, and from predictions based on historical data. This paper focuses on the description and preliminary tests of the MIP optimization model designed to get the optimized plans for each planning horizon.

4. Mixed Integer Linear programming Model

The base of the MIP model is the work of Kek et al. (2009), with three main improvements. First, the designed model is prepared to be used in a rolling horizon planning approach by allowing the initialization of staff with previous tasks. Second, it allows trip joining of staff, meaning that staff members can travel inside the same vehicle, benefiting from the use of the available vehicle seats to reduce costs. And third, the model, here described, is prepared to consider two maintenance procedures with different time durations.

The use of a rolling horizon approach allows an acceptable data update rate for a real time interaction. Each planning horizon is subdivided in time steps. To accommodate the tasks that come from a previous planning horizon, the lower limit of the modelled period is extended to the beginning of the first staff activity from the previous horizon (B), considering that the first time step of the current planning horizon is equal to one. Therefore, the set of time steps considered is \( I = \{B, ..., i, ..., T\} \). The model discretizes space in stations. The set of stations is \( N = \{1, ..., i, ..., S\} \), being \( S \) the number of considered stations. If the space is discretized in zones, each zone can be considered as a station. A time-space network \( V \) denotes all the \( S \times I \) nodes, \( V = N \times I = \{1_{B}, ..., i_{1}, ..., S_{T}\} \). The set of arcs between the nodes defined in \( V \), is designated by \( A \). The travel time between stations is defined by variable \( t_{ij} \).

A set of staff \( L = \{1, ..., k, ..., W\} \) is available to carry out the maintenance and relocation activities. Members of staff are discretized and each member is assigned to perform only one activity at each time. Idling \( y^{i}(i_{i}, i_{t+1}) \), moving inside a carsharing system vehicle \( u^{i}(i_{ij}, i_{t+j}) \), moving by using other means of transportation \( s^{i}(i_{ij}, i_{t+j}) \), and performing maintenance, are the activities that can be attributed to staff members.
Staff uses vehicles to simultaneous move along the network according to maintenance needs and to perform relocation activities. The maintenance procedures performed by each staff member \((k)\) are: simple maintenance \(z^\beta_k(i_t, t_i+t_{i\beta})\) and refueling \(z^\gamma_k(i_t, t_i+t_{i\gamma})\). Each procedure has the duration \(t_{i\beta}\) and \(t_{i\gamma}\), respectively. Simple maintenance includes cleaning and other small tasks that can be performed locally. In refueling operations, members of the staff move vehicles to the nearest gas station, leaving it afterwards at positions inside the same station (the model considers petrol engine vehicles but can be easily adapted to enfold electric vehicles). Relocation activities are a response to the client demand requests. Demand requests, \(d(i_t)\), are defined before the optimization process for each instant \(t>0\) and for each station \(i\). The model considers that the demand does not need to be totally fulfilled, but the rejected demand \(d^r(i_t)\) is penalized in the objective function. The fulfilled demand at each station and instant is described by \(k(i_t)\). The vehicles requested by clients “disappear” from the \(a(i_t)\) vector of the optimization results, though the number is updated on the next planning horizon when information is retrieved from the system.

The number of vehicles per status type per station is input data of the MIP model, and is given previous to each planning horizon. The status considered are: number of available vehicles \(a(i_t)\), the number of vehicles needing simple maintenance \(\beta(i_t)\), and the number of vehicles needing to be refueled \(\gamma(i_t)\), at each station \(i\) and time step \(t\). Clients can only use available vehicles \(a(i_t)\) and staff can use all the vehicles.

Vehicle movements are described in aggregated variables. The variables \(v_d(i_t, j_t+t_{ij})\), \(v_\beta(i_t, j_t+t_{ij})\), and \(v_\gamma(i_t, j_t+t_{ij})\), quantify the number of available vehicles, the number of vehicles needing simple maintenance, and the number of vehicles needing to be refueled, respectively, moving from station \(i\) at time step \(t\) to station \(j\) at time step \(t+t_{ij}\). To quantify the number of seats available on those movements, the number of vehicles moving is multiplied by the constant \(v_c\), the vehicle capacity in number of seats. It is considered that all vehicles have the same capacity.

The problem formulation has nine decision variables:

\[
v_d(i_t, j_t+t_{ij}) \quad \text{variable} \quad \text{quantifying the number of available vehicles not needing maintenance moving from station } i \text{ at time step } t \text{ to station } j \text{ at time step } t+t_{ij};
\]

\[
v_\beta(i_t, j_t+t_{ij}) \quad \text{variable} \quad \text{quantifying the number of vehicles needing maintenance type } \beta \text{ moving from station } i \text{ at time step } t \text{ to station } j \text{ at time step } t+t_{ij};
\]

\[
v_\gamma(i_t, j_t+t_{ij}) \quad \text{variable} \quad \text{quantifying the number of vehicles needing maintenance type } \gamma \text{ moving from station } i \text{ at time step } t \text{ to station } j \text{ at time step } t+t_{ij};
\]

\[
u^k(i_t, j_t+t_{ij}) \quad \text{binary variable associated with a staff movement inside a vehicle, taking the value } 1 \text{ if staff } k \text{ moves from station } i \text{ at time step } t \text{ to station } j \text{ at time step } t+t_{ij}, \text{ and } 0 \text{ otherwise;}
\]

\[
u^k(i_t, j_t+t_{ij}) \quad \text{binary variable associated with a staff movement using other modes, taking the value } 1 \text{ if staff } k \text{ moves from station } i \text{ at time step } t \text{ to station } j \text{ at time step } t+t_{ij}, \text{ and } 0 \text{ otherwise;}
\]

\[
u^k(i_t, j_t+t_{ij}) \quad \text{binary variable associated with a staff member waiting for next task, taking the value } 1 \text{ if staff } k \text{ is waiting at station } i \text{ from time step } t \text{ to time step } t+1, \text{ and } 0 \text{ otherwise;}
\]

\[
u^k(i_t, j_t+t_{ij}) \quad \text{binary variable associated with the maintenance activity type } \beta, \text{ taking the value } 1 \text{ if staff } k \text{ is maintaining a vehicle at station } i \text{ from time step } t \text{ to time step } t+t_{j\beta}, \text{ and } 0 \text{ otherwise;}
\]

\[
u^k(i_t, j_t+t_{ij}) \quad \text{binary variable associated with the maintenance activity type } \gamma, \text{ taking the value } 1 \text{ if staff } k \text{ is maintaining a vehicle at station } i \text{ from time step } t \text{ to time step } t+t_{j\gamma}, \text{ and } 0 \text{ otherwise;}
\]

\[
u^r(i_t) \quad \text{number of rejected vehicle demand requests at station } i \text{ at time step } t.
\]

With four additional dependent variables:
\(a(i_t):\) number of vehicles available to clients at station \(i\) at time step \(t\);
\(\beta(i_t):\) number of vehicles needing maintenance type \(\beta\) at station \(i\) at time step \(t\) (can only be used by staff);
\(\gamma(i_t):\) number of vehicles needing maintenance type \(\gamma\) at station \(i\) at time step \(t\) (can only be used by staff);
\(k(i_t):\) number of fulfilled client demand at station \(i\) at time step \(t\).

The known constants are:

\(v_c: \) vehicle capacity in number of seats;
\(t_{ij}: \) travel time between stations;
\(t_\beta: \) time to complete maintenance procedure type \(\beta\);
\(t_\gamma: \) time to complete maintenance procedure type \(\gamma\);
\(d(i_t): \) number of demand requests by clients at station \(i\) at time step \(t\);
\(a(i_0): \) number of vehicles available to clients at station \(i\) at time step \(t=0\);
\(\beta(i_0): \) number of vehicles needing maintenance type \(\beta\) at station \(i\) at time step \(t=0\);
\(\gamma(i_0): \) number of vehicles needing maintenance type \(\gamma\) at station \(i\) at time step \(t=0\);
\(c_v(i,j): \) cost of a vehicle movement by staff between stations \(i\) and \(j\);
\(c_s(i,j): \) cost for staff movement using other transport modes between stations \(i\) and \(j\);
\(c_d: \) penalty for rejecting one client demand request;
\(c_\beta: \) penalty for maintenance request \(\beta\) not fulfilled or delayed to the next time step;
\(c_\gamma: \) penalty for maintenance request \(\gamma\) not fulfilled or delayed to the next time step.

The mixed integer linear programming formulation for the problem is:

\[
\min(\Pi) = \sum_{m \in [a,\beta,\gamma]} \sum_{(i_t,j_{t+1}) \in A} c_v(i,j) \cdot V_m(i_t,j_{t+1}) + \sum_{k \in L} \sum_{(i_t,j_{t+1}) \in A} c_s(i,j) \cdot S^k(i_t,j_{t+1}) + c_d \sum_{i_t \in V} \sum_{i \in V} d^v(i_t) + \\
+ c_\beta \sum_{i_t \in V} \beta(i_t) + c_\gamma \sum_{i_t \in V} \gamma(i_t)
\]

Subject to:

\[
\sum_{t=B}^{t=0} \sum_{i \in N} \gamma^k(i_t,i_{t+1}) + \sum_{t=B}^{t=0} \sum_{i \in N} \beta^k(i_t,i_{t+1}) + \sum_{t=B}^{t=0} \sum_{i \in N} \gamma^k(i_t,i_{t+1}) + \sum_{t=B}^{t=0} \sum_{i,j \in N} u^k(i_t,j_{t+1}) + \\
+ \sum_{t=B}^{t=0} \sum_{i,j \in N} S^k(i_t,j_{t+1}) = 1 , \forall k \in L
\]
\[
\sum_{i=0}^{t-0} \sum_{k \in L} u^k(i, t_{\text{set}_0}) = \sum_{m \in \{a, \beta, \gamma\}} \sum_{i=0}^{t-0} v_m(i, t_{\text{set}_0}), \forall (i, t_{\text{set}_0}) \in A
\]

(3)

\[
y^k(i, t_{\text{set}}) + z^k_{\beta}(i, t_{\text{set}}) + z^k_{\beta}(i, t_{\text{set}}) + \sum_{(j, t_{\text{set}}) \in A} u^k(j, t_{\text{set}}) + \sum_{(j, t_{\text{set}}) \in A} s^k(j, t_{\text{set}}) - y^k(i, t_{\text{set}}) - z^k_{\beta}(i, t_{\text{set}}) - z^k_{\beta}(i, t_{\text{set}}) - \sum_{(j, t_{\text{set}}) \in A} u^k(j, t_{\text{set}}) - \sum_{(j, t_{\text{set}}) \in A} s^k(j, t_{\text{set}}) = 0
\]

(4)

\[
\forall i_t \in V | t > 0, k \in L
\]

\[
\sum_{k \in L} u^k(i, t_{\text{set}}) \geq \sum_{m \in \{a, \beta, \gamma\}} v_m(i, t_{\text{set}}), \forall (i, t_{\text{set}}) \in A | t > 0
\]

(5)

\[
\sum_{k \in L} u^k(i, t_{\text{set}}) \leq v_t \times \sum_{m \in \{a, \beta, \gamma\}} v_m(i, t_{\text{set}}), \forall (i, t_{\text{set}}) \in A | t > 0
\]

(6)

\[
m(i) \geq v_m(i, t_{\text{set}}) \land m \in \{a, \beta, \gamma\}, \forall (i, t_{\text{set}}) \in A | t > 0
\]

(7)

\[
a(i) = a(i_{t-1}) + \sum_{(j, t_{\text{set}}) \in A} v_a(i_{t-1}, t_{\text{set}}) - \sum_{(j, t_{\text{set}}) \in A} v_a(i, t_{\text{set}}) + \sum_{k \in L} z^k_{\beta}(i, t_{\text{set}}) + \sum_{k \in L} z^k_{\beta}(i, t_{\text{set}}) - k(i), \forall t \in T | t > 0
\]

(8)

\[
\beta(i) = \beta(i_{t-1}) + \sum_{(j, t_{\text{set}}) \in A} v_{\beta}(i_{t-1}, t_{\text{set}}) - \sum_{(j, t_{\text{set}}) \in A} v_{\beta}(i, t_{\text{set}}) - \sum_{k \in L} z^k_{\beta}(i, t_{\text{set}}), \forall t \in T | t > 0
\]

(9)

\[
\gamma(i) = \gamma(i_{t-1}) + \sum_{(j, t_{\text{set}}) \in A} v_{\gamma}(i_{t-1}, t_{\text{set}}) - \sum_{(j, t_{\text{set}}) \in A} v_{\gamma}(i, t_{\text{set}}) - \sum_{k \in L} z^k_{\gamma}(i, t_{\text{set}}), \forall t \in T | t > 0
\]

(10)

\[
d(i) = k(i) + d^r(i), \forall t \in T | t > 0
\]

(11)

\[
u^k(i, t_{\text{set}}), y^k(i, t_{\text{set}}), z^k_{\beta}(i, t_{\text{set}}), z^k_{\beta}(i, t_{\text{set}}), s^k(i, t_{\text{set}}) = \{0, 1\}, \forall k \in L
\]

(12)
The objective function (1) minimizes the generalized cost function $\Pi$ that includes: the cost of vehicle movements used for staff operations, the cost of staff moving using other modes of transport, the cost of rejected demand, and a penalty for maintenance requests not fulfilled. Constraints (2) and (3) aid the initialization of variables from the previous planning period. Constraint (2) limits the staff operations initiated until the instant $t=0$, to only one task per staff member. Constraint (3) limits the vehicle movements initiated until $t=0$, to only the ones presented as an input. Constraint (4) assures the conservation of staff activities at each station $i$ and time instant $t \geq 0$. It restricts staff to start a new activity only after finishing the previous one. Constraints (5) and (6) relate the staff movements in the system vehicles, by imposing a minimum and maximum number of staff traveling in each vehicle. The minimum of one member is needed to drive each vehicle and the maximum number is related to the vehicle’s capacity in number of seats. This allows having trip joining and trip splitting in staff movements. Constraint (7) imposes that a certain vehicle with a status type, can only depart from a station where vehicles with the same status type exist. Constraints (8) to (10) update the values of the variables between time steps. Constraint (11) is a conservation equation relating total demand, with fulfilled and rejected demand requests. Constraint (12) sets the binary variables, and constraints (13) and (14) set the non-negative integer variables.

5. Testing

The MIP model has nine sets of decision variables, which number of elements depends on the problem characteristics. The number of variables for the primal problem related to $i$ is $W \times S \times (T-1)$, since the relation between start time and ending time is $t_{\text{end}} = t_{\text{start}} + 1$. The two $z$ sets of variables $z_b$ and $z_r$ have $W \times S \times (T-t_{\beta})$ and $W \times S \times (T-t_{\gamma})$, respectively. This is, once more, due to the ending time being dependent to the start time. For the sets $u$ and $s$, the total number of variables is equal to $2 \times W \times (S^2 - S) \times (T^2 - \sum_{t_{\beta}}^T t_{\beta})$. The travel times between stations vary according to origin-destination vector, although there are no trips occurring from $t_2$ to $t_1$ ($t_2 > t_1$), enabling to subtract $\sum_{t_{\beta}}^T t_{\beta}$ to $T^2$. One is also able to subtract $S$ to $S^2$, since there is no sense on having movements or relocations between the same station. To the sets $v_a$, $v_b$, and $v_r$ corresponds a total number of variables equal to $3 \times (S^2 - S) \times (T^2 - \sum_{t_{\beta}}^T t_{\beta})$. The rejected demand variable set $d'$ has $S \times T$ elements, since demand is only modeled for the optimization period, $t > 0$.

Due to the high number of variables, it is important to understand the dimensions of the problem that allow a timely optimization. The model was tested using the optimization software Xpress-MP. For that purpose, we developed a computer routine to generate random input data. The routine has flexibility to change the dimension of the problem, allowing creating data input files with different characteristics. Different input data files were created by changing the number of stations (S) and the number of staff (W). The planning horizon extension used was six time steps ($T=6$). Additionally, it was considered one element of staff per 20 vehicles. The demand data and maintenance requests data was randomly generated by the computer routine created. Optimization tests were performed on an Intel Core i5 with 1.70 GHz of processor speed and 4 GB of RAM.

The first tests using random generated input data with smaller dimensions, allowed verifying that the MIP formulation worked accordingly. Staff is correctly initiated with the previous tasks, each one starting at time steps $t < 1$, and trip-joining is being properly applied when needed to minimize costs.

Increasing the dimension of the problem, the time in seconds to reach a solution increases (see Table 1). For dimensions of 30 and 50 stations with 40 staff members, the optimization process didn’t reach the end inside the limit imposed for processing time (2000 seconds) and the solutions retrieved had a considerable duality gap. In the case of 70 stations with 40 staff members, the optimization process didn’t provide a solution before reaching the time limit.
6. Conclusions

A mixed integer programming formulation was designed as an upgrade of a previous model developed by Kek et al. (2009). The new model is aimed to optimize the staff activity in real time. The considered staff activity consists in maintenance tasks and relocations. The formulation of the model includes two types of maintenance with distinct durations. Staff members move using mainly the system vehicles and are able to share vehicles – staff trip joining. Nevertheless, staff can travel by other modes if necessary. The MIP model is designed to include incomplete staff activities, making it possible to consider a rolling horizon optimization planning approach. Each staff member is discretized in separate variables allowing a detailed tracing and planning. The tests showed that dividing the operation period into smaller planning periods (T=6), reduces the problem dimension, and consequently allows getting interesting computational times for considerable problem sizes. This is suitable for real time interaction based on a rolling horizon planning approach.

Table 1. Xpress-MP Results

<table>
<thead>
<tr>
<th>stations, S</th>
<th>staff, W</th>
<th>#solution</th>
<th>duality gap (%)</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>10</td>
<td>4</td>
<td>0.10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>5</td>
<td>0.008</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>8</td>
<td>0.02</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>9</td>
<td>0.01</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>23.31</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>2</td>
<td>23.07</td>
<td>1216</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>6</td>
<td>6.14</td>
<td>1220</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1</td>
<td>55.82</td>
<td>582</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>no more solutions in 2000s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>2</td>
<td>0.04</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3</td>
<td>0.006</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>2</td>
<td>0.11</td>
<td>204</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>3</td>
<td>0.10</td>
<td>277</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>2</td>
<td>0.28</td>
<td>541</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>no more solutions in 2000s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1</td>
<td>15.62</td>
<td>1846</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>no more solutions in 2000s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>10</td>
<td>2</td>
<td>0.03</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3</td>
<td>0.004</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1</td>
<td>0.07</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>2</td>
<td>0.02</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>1.05</td>
<td>593</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>2</td>
<td>0.08</td>
<td>997</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>4</td>
<td>0.004</td>
<td>1816</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>no solutions in 2000s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Acknowledgements

The authors would like to thank the Portuguese Science Foundation for financing the InnoVshare project (PTDC/ECM-TRA/0528/2012).

References


