

Minimizing CO_2 emissions on a practical daily carpooling problem

Bruno P. Bruck, Valerio Incerti, Manuel Iori, Matteo Vignoli

DISMI, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy

Via Amendola 2, Pad. Morselli, 42122 Reggio Emilia, Italy

{bruno.petratobruck, valerio.incerti, manuel.iori,
matteo.vignoli}@unimore.it

RESUMO

O aumento da conscientização sobre os danos causados ao meio ambiente por atividades humanas está ganhando mais atenção por parte dos governos, assim como de empresas e pessoas em geral. Neste contexto, a redução das emissões de CO_2 é um fator importante e já existem diversas práticas com o intuito de atingir esta meta. Um exemplo é o chamado carpooling, que pode ser definido como o ato de um grupo de pessoas dividindo o mesmo carro. Neste artigo nós abordamos um caso prático de uma grande empresa Italiana e propomos duas formulações matemáticas junto com dois algoritmos heurísticos. O objetivo é de desenvolver um sistema web integrado afim de ser utilizado pelos trabalhadores da empresa para a organização de carpoolings. Resultados experimentais indicam um grande potencial de redução de emissões de CO_2 pela adoção desta prática e um primeiro protótipo do sistema está para ser instalado na empresa a fim de testes.

PALAVRAS CHAVE. carpooling, roteamento de veículos, logística

Área Principal: roteamento de veículos

ABSTRACT

The increasing awareness on the damages to the environment caused by human activity is getting increased attention from governments, as well as companies and individuals. In this sense, the reduction of CO_2 emissions is an important subject and there are a range of practices that help achieve this goal. One example is carpooling, which is defined as the act of individuals sharing a single car. In this paper we approach a practical case found in a large Italian company and propose two mathematical formulations and two heuristic algorithms. The objective is to develop an integrated web application to be used by the employees of this company in order to organize carpoolings. Experimental results attest for a great potential in CO_2 savings by the use of carpooling in this scenario and a first prototype of the application is being deployed for testing.

KEYWORDS. carpooling, vehicle routing, logistics

Main Area: Vehicle Routing

1. Introduction

The debate on the negative effects of human activities on the environment has been around for a long time now, and the concern about these issues is quite widespread. In particular, pollution is a plague in big industrial cities and a cause of illness for inhabitants. Large companies daily require that thousands of individuals travel to work, heavily increasing car use. The drawbacks of this practice are common knowledge (Gärling e Friman (2015)), causing a range of problems including high emissions of CO_2 in the atmosphere, noise pollution, parking issues and is increasingly becoming unsustainable, as mentioned by Gärling e Friman (2015). The use of public transportation systems could mitigate the problem, but more often than not they are incapable of serving all the demand cost-effectively and many individuals refuse or prefer not to use it.

In this sense, *carpooling*, a car sharing practice, can be an effective tool to help reduce traffic. It can be defined as the act of a group of individuals that ride a single car, splitting travel costs (e.g., Furuhata *et al.* (2013)), and has grown more common in recent years, especially during energy crises.

Carpooling is an interesting transportation habit for individuals as well as companies, as it can reduce transportation costs and directly impact on CO_2 emissions. Governments are also taking action, for instance, in Italy a law of sustainable mobility was included into the national legislation in 1998 to stimulate the use of collective transportation methods and promote the creation of innovative transport systems. Moreover, one of the so-called millennium goals of the United Nations is ensure environmental sustainability (United Nations (2015)) and includes the reduction of CO_2 emissions.

In this paper, we study a practical problem found in a large Italian company, Coopservice S.coop.P.A, whose aim is to encourage its employees to practice carpooling in order to reduce transportation costs and CO_2 emissions. This case study considers employees directed to the same workplace, but possibly having different working shifts. The final objective is to develop an optimized web application in which employees can agree with each other on sharing rides.

The remainder of the paper is organized as follows. In Section 2 we present a brief review of the related literature. Then in Section 3 we describe our case study, along with the developed approaches and computational results. In Section 4 a prototype of the web application is presented and, finally, some conclusions are shown in Section 5.

2. Literature review

During the decades, the interest raised by carpooling spread to many fields of research (Teal (1987)). Eventually, scientists started to think on how to create computer-based applications and models to describe the complexity and the many forms of carpooling.

The first dimension characterizing carpooling is the time frame considered. In literature, two types of models emerged in this sense, the *Daily Carpooling Problem* (DCPP) and the *Long-term Carpooling Problem* (LCPP). While in the former users must agree on how to carpool in a daily basis, in the latter the objective is seen in the long term, where users basically form groups and organize shared rides among the individuals of these group during a certain period of time (Wolfler Calvo *et al.* (2004)).

Our case can be classified as a DCPP, because employees may have different shifts at each day, which makes it impossible to use a long term solution. There are some contributions in the literature on problems of this class. For instance, Wolfler Calvo *et al.* (2004) approach a real-life DCPP and develop an integrated system to organize and manage carpoolings. Given the necessity for a fast response from the system they opted to use an heuristic algorithm to solve it. This algorithm is a 2-step procedure and it is the core of their optimization module. In the first step, a graph containing the distance between each user is evaluated based on a modified version of the Dijkstra shortest path algorithm that considers traffic congestion. Secondly, a greedy constructive heuristic is run under this graph to define an initial solution, which is then given to a local search procedure

that tries to improve it. Results seem promising and instances with less than 400 employees are handled effectively.

Baldacci *et al.* (2004) also approach a DCP in which individuals are assigned to existing vehicles. In the paper, the authors model their scenario as a dial-a-ride problem with the assumption that all vehicles are equal. They propose both exact and heuristic solution methods. The exact method is a bounding based iterative procedure, where at each iteration a reduced set-partitioning problem is solved. As a constructive algorithm they propose a Lagrangian heuristic, which provides a valid upper bound for the problem. The approaches are tested on instances derived from the literature and yield good results, being able to solve effectively many classes of problems.

As for the LCPP, Yan *et al.* (2011) propose to consider groups of individuals who should travel together rather than single individuals. The authors propose an integer multiple commodity network flow formulation for the LCPP. Their model aims to both reduce global costs and share costs fairly between users. Furthermore, the authors propose a Lagrangian relaxation with subgradient to provide valid lower bounds and then used a Lagrangian heuristic to efficiently derive upper bounds. The approaches are tested on instances from the literature and results indicate that the solution method could be used to solve large size instances in a real case.

3. Case Study

In order to formally introduce the problem, let us define $G = (V, A)$ as a complete directed graph, where $V = \{0\} \cup V'$, vertex 0 represents the workplace, vertices in V' represent the employees, and $A = \{(i, j) : i \in V', j \in V, i \neq j\}$. For convenience of notation, let us partition the set of employees into two subsets, those that own a car and those that do not, namely V_c and V_n , respectively. Each vertex $i \in V_c$ either carpools or uses its own car to go directly to the workplace, and each vertex $j \in V_n$ may use the public transportation to reach the workplace or, alternatively, carpool. Vehicles have a maximum capacity of Q people and we assume that there are no exchanges of passengers between routes. With every arc $(i, j) \in A$ there is an associated non-negative cost c_{ij} , specifying the minimal real distance between the two vertices. Notice that the cost matrix c can be asymmetric, because in our problem we consider distances on a real map. The objective is to minimize the total CO_2 emission, which is calculated based on the total distance traveled by car and by using public transportation.

We consider two different types of solutions. The first one, namely *direct-route*, can be defined as a set of routes, each of them beginning at the house of a given employee and ending at the workplace. In this case, the first employee on the route drives his/her car and picks up the others along the route. In the second type of solutions, namely *tree-route*, we also allow employees to drive to intermediary points and then use a single car from there on. These points can be viewed, for instance, as parking lots or the house of other employees. While this consideration allows for some flexibility it also requires more organization and commitment between participants, which might become an inconvenient in some cases due to waiting times and lack of parking spots at the meeting point. An example of *direct-route* and *tree-route* solutions are shown in Figures 1-(a) and 1-(b), respectively. The workplace is represented by vertex 0, vertices in solid and dashed lines are employees that own a car and those that do not, respectively. The values on the arcs correspond to the distances and $Q = 4$. One can notice that, because of their more restrictive nature, *direct-route* solutions tend to result in higher costs in terms of both distance and CO_2 emissions.

Since the goal is to develop a real application, we decided to take into consideration the maximum acceptable detour from the direct path to the workplace, that drivers might accept to carpool. We selected the average detour value that is proposed in the literature (Rietveld *et al.* (1999)) and that amounts to 17%.

In the following subsections, we first describe how the data set provided by the company was polished to remove inconsistencies. Then, we detail the procedure used to evaluate the non-carpooling scenario, which provides us with a basis for comparison with the solutions originated

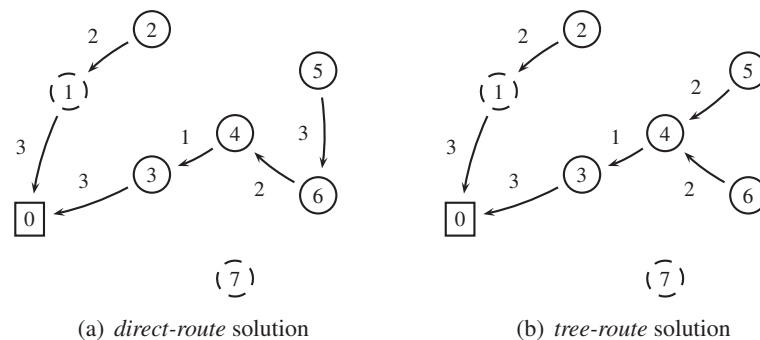


Figure 1: Solution types. The depot is represented by vertex 0, while vertices in solid and dashed lines are employees that own a car and those that do not, respectively.

from our approaches. The developed mathematical formulations and heuristic algorithms are then presented in details, followed by computational experiments to evaluate their efficiency.

3.1. Data Analysis

The company provided us with two data sets regarding employees working at a single workplace. The first one, namely $data^{transp}$, contains information regarding home-to-work travel details of a total of 137 employees. More specifically, it specifies whether they go by car, by bicycle or by public transportation. The second one, $data^{shifts}$, provides information on the shifts of the employees during a period ranging from mid-February to December 2012.

These data sets showed some mutual inconsistencies and internal problems. As a first step to bring them into a consistent state, we performed an analysis to detect the main issues, which are described in the following along with the respective police implemented to correct them.

- 17 people had missing transportation information. In this case, we assigned a transportation method based on the proximity to the workplace and updated $data^{transp}$ accordingly.
- 30 employees had missing house numbers on the address, and that made it difficult to get the associated geographical coordinates. To approach this issue, we took the middle point of the corresponding street as the location of each of their houses.
- A couple of employees from $data^{transp}$ did not have complete information regarding their shifts and were removed from the case study.

As a second step, we performed an analysis to understand the distribution of shifts along the week. As shown in Figure 2, the number of employees per weekday is usually stable and it ranges, in average, around 80, with the exception of Sundays, where it ranges around 40 instead. A direct consequence of the small number of employees working on Sundays is that there are considerably less carpooling options on these days. Therefore, we remove data regarding Sunday shifts from the study. After this polishing process, we ended up with the two data sets containing information about 135 employees over a period of 276 days.

The few evident fluctuations in Figure 2 represent holidays or special occasions, when the number of employees is naturally lower. These situations are usually predictable and do not have a significant impact on the performance of a carpooling application.

Notice that an important factor, that might heavily affects the probability of success of an application in this context, is the number of shift typologies. In fact, the more the number of different shifts, the fewer are the possible matching options for carpooling. We found a total of 171 different shift typologies during the period considered on the study. However, some of them are much more common than others. For instance, we noticed that the shifts 6:00-12:30 and 14:00-20:30 were the most frequent, with frequency rates of 27% and 22%, respectively. Furthermore, it is interesting to notice that the 11 most common shifts account for 80% of all entries. The findings seem to suggest that deploying a carpooling application in this scenario could lead to interesting

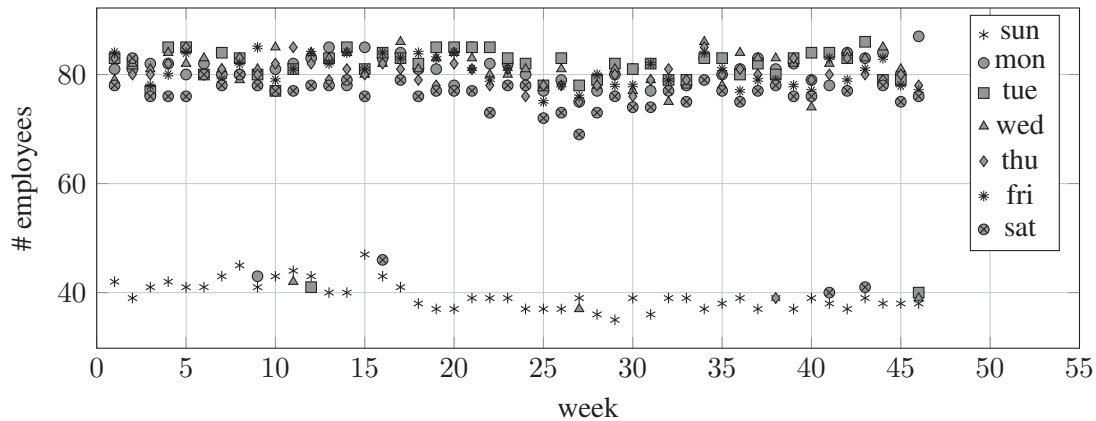


Figure 2: Distribution of employees per weekday

results in terms of a significant reduction in CO_2 emissions. They also indicate that the company can improve the efficiency of the process, by adjusting employees' shifts so as to create larger and denser groups.

3.2. The non-carpooling scenario

Evaluating the non-carpooling scenario is an important step of the study, because it provides us with an upper bound on the CO_2 emissions that can be used as a basis for assessing the possible improvements over the current situation of the company.

Because information regarding the distances between the points, i.e., the employees houses and the workplace, are not given in the dataset, we must first evaluate them. For that end, we developed an application that takes as input a digital map of the region and a list of geographic coordinates for each point, and produces as output a matrix containing the value of the shortest path between each point.

The coordinates are based on the addresses of the employees and the workplace, and the digital map of the region was acquired from the free OpenStreetMap (OSM) database. This map is given in the ShapeFile format, which is a popular geospatial vector data format for geographic information system software.

The application first reads the map and internally represents it as a directed graph. Then it takes the list of points for which the distance matrix should be evaluated. Notice that a map-matching procedure is necessary in order to properly connect the set of points into the graph. This procedure works as follows. Suppose a given point p must be added. At first, we search for all vertices in a given distance range from p and select the nearest one as the best candidate for the matching. This situation is depicted in Figure 3-(a). Notice that only vertices A and C are in range, but A is the nearest, thus, it is selected as the possible matching. Although it is fairly uncommon, in case no vertex is found, we expand the diameter of the range and try again. This is done until at least one match is found.

Secondly, we project p into all edges in range, adding the possibility of including a new vertex to an edge and connect it to p . In our example, shown in Figure 3-(b), the nearest projection found is over the arc (A, C) . The best option between the nearest neighbor and the nearest projection is then chosen to connect p to the map.

Finally, to evaluate the shortest path from each point to the other we implemented the classical A Star algorithm, solved under our modified graph. Then, with this procedure, it becomes straightforward to evaluate the cost in terms of total distance traveled by employees when not sharing rides.

In order to analyze the environmental impact we use a set of coefficients developed by a research-oriented company. These coefficients directly relates CO_2 emission to the distance traveled. However, because the car models used by the employees are unknown, we evaluate an average coefficient γ_c to estimate the CO_2 emitted by them. Similarly, because we assume that employees

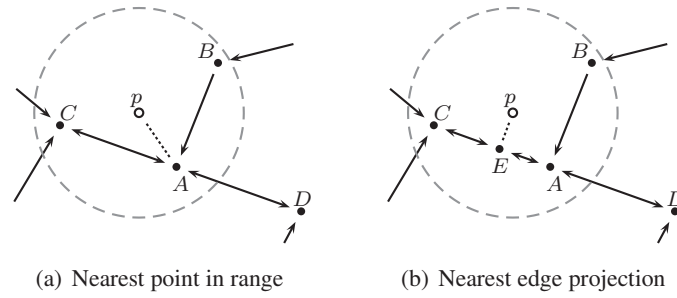


Figure 3: Example of the map-matching procedure

that neither own a car nor carpool use the public transportation system, we evaluate an average coefficient γ_b for this type of transportation.

3.3. Exact and heuristic solution approaches

As a first approach to the problem, we propose an integer programming formulation, which provides a lower bound on the optimal scenario, where employees organize carpools among themselves in the best possible way, so as to minimize the total CO_2 emissions. This lower bound is useful to help evaluating the performance and solution quality of our heuristic algorithms.

To describe the formulation, consider a binary variable x_{ij}^k as having value 1 in case employee $k \in V'$ is traveling through arc $(i, j) \in A$ and let y_{ij} be a binary variable taking value 1 only if arc $(i, j) \in A$ is used. Furthermore, to model the possible behaviors of employees without a car, we define an additional binary variable v_i for each $i \in V_n$. This variable takes value 1 if employee i decides to use the public transportation system to reach the workplace instead of carpooling, and 0 otherwise. Then, we define the following tree-toute formulation (F_{TR}).

$$(F_{TR}) \quad \min z_{F_{TR}} = \sum_{i \in V'} \sum_{j \in V} (c_{ij} \gamma_c) y_{ij} + \sum_{i \in V_n} (c_{i0} \gamma_b) v_i \quad (1)$$

subject to

$$\sum_{j \in V} x_{kj}^k = 1 \quad \forall k \in V_c \quad (2)$$

$$\sum_{j \in V} x_{kj}^k = 1 - v_k \quad \forall k \in V_n \quad (3)$$

$$\sum_{i \in V'} x_{i0}^k = 1 \quad \forall k \in V_c \quad (4)$$

$$\sum_{i \in V'} x_{i0}^k = 1 - v_k \quad \forall k \in V_n \quad (5)$$

$$\sum_{i \in V'} x_{ij}^k - \sum_{i \in V} x_{ji}^k = 0 \quad \forall j, k \in V', j \neq k \quad (6)$$

$$\sum_{k \in V'} x_{ij}^k \leq Q y_{ij} \quad \forall i \in V', j \in V \quad (7)$$

$$\sum_{i \in V'} \sum_{j \in V} c_{ij} x_{ij}^k \leq c_{k0} (1 + \delta) \quad \forall k \in V' \quad (8)$$

$$\sum_{j \in V} y_{ij} \leq 1 \quad \forall i \in V_c \quad (9)$$

$$\sum_{i \in V} y_{ij} \geq 1 - v_j \quad \forall j \in V_n \quad (10)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in V, k \in V' \quad (11)$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in V', j \in V \quad (12)$$

$$v_i \in \{0, 1\} \quad \forall i \in V_n \quad (13)$$

The objective function (1) minimizes the total CO_2 emission, and consists on two components. While the first evaluates the total emission by the cars, the second estimates the emissions of public transportation vehicles used by the employees that are not carpooling. Constraints (2) specify that all employees that own a car must be on a route, even if not sharing a ride, whereas in (3) we define that those without a car might not be on any route at all and, instead, use public transportation. The sets (4) and (5) assure that all employees that are on any route, must arrive at the destination and (6) guarantees that, once in a route an employee cannot leave at intermediary points. Furthermore, the capacity of a car cannot be exceed (7) and the maximum detour constraint must hold (8). Constraint (9) define that at most one car can leave a given point and (10) specify that an employee that does not own a car is either picked up by someone or does not carpool. Finally, (11), (12) and (13) define the formulation variables.

Notice that, the formulation is named based on the observation that it allows the so-called *tree-route* solutions, because the number of incoming arcs to vertices is not restricted. However we can easily adapt it for the case of only *direct-route* solutions by introducing the following set of constraints.

$$\sum_{i \in V'} y_{ij} \leq 1 \quad \forall j \in V' \quad (14)$$

From now on, we refer to the formulation defined by constraints (1)-(13), (14) as F_{DR} . Notice that the problem induced by F_{TR} is NP-hard because it can be reduced into the capacitated minimum spanning tree (CMST) when $V_n = \emptyset$, c is symmetric and the triangle inequality holds. We try to overcome this difficult in two ways, first by improving F_{TR} , and secondly by developing heuristic algorithms.

Indeed, it is possible to strengthen the formulations by fixing certain arcs based on the maximum detour allowed from the directed path to the destination. Let $\gamma(i, j)$ be a function that returns 1 in case vertex $j \in V'$ is reachable from vertex $i \in V'$, i.e., $c_{ij} \leq c_{i0}(1 + \delta)$, where δ represents the maximum detour allowed, ranging in the interval $[0, 1]$; and 0 otherwise. Then, consider the following additional set of constraints, incorporated into both formulations.

$$y_{ij} = 0 \quad \forall i, j \in V' : \gamma(i, j) = 0 \quad (15)$$

The first heuristic, namely H_{DR} , is guided by a regret measure. This type of heuristic, as mentioned by Pisinger e Ropke (2007), tries to overcome the problem of postponing the insertion of difficult requests by considering a kind of lookahead information. Basically, the measure of the regret of a certain request is calculated by taking the cheapest insertion cost and subtracting it from the second cheapest. One can notice that requests with a high regret value are probably critical, in the sense that they are not very flexible and, if left unattended for long, the solution value can be significantly increased.

The algorithm works as in the following: at each iteration, a new route is created, the employee with the greatest regret value that can drive and is still not in the solution is chosen to initialize the route. At this point, the route is simply the direct path from the chosen employee to the workplace. Next, we try to insert as many employees as possible into this route, following the regret order and respecting maximum detour and capacity constraints. Once there are no possible feasible insertions the route is added to the solution and a new iteration begins, following the same procedure. The algorithm stops when no more routes can be created. In case it terminates and there are employees not participating in any rides, we consider that they use the public transportation system and adjust the solution value accordingly.

An important detail about the aforementioned algorithm is that it only generates *direct-route* solutions. Thus, we have developed a second heuristic approach, namely H_{TR} , which is

capable of yielding *tree-route* solutions. The algorithm is based on the well known heuristic proposed by Esau e Williams (1966) for solving the CMST and is guided by the evaluation of savings as well.

To explain how we evaluate the savings in our algorithm, consider the example depicted in Figure 4 and, for the sake of simplicity, assume route capacity as 6. In addition, let us redefine the concept of *gate* of a vertex i , namely g_i , as the arc connecting i to the next vertex in the route to the workplace, e.g. g_7 refers to arc $(7, 6)$. Then, the saving $s_{ij} = g_i - c_{ij}$. For instance, in Figure 4, $s_{73} = 3 - 1 = 2$.

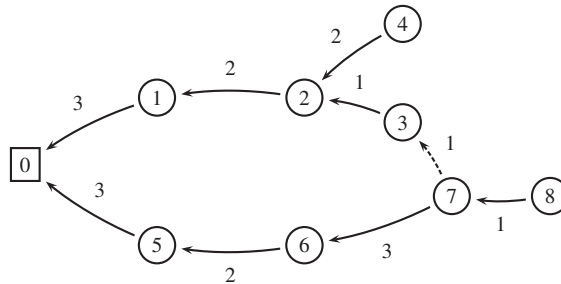


Figure 4: Savings evaluation for arc $(7, 3)$: $S_{73} = g_7 - c_{73} = 2$

The algorithm starts with a star tree solution in which every employee is directly connected to the workplace by a single arc. Then, at each iteration we follow a 4-step procedure as in the following.

- Step 1 : Find two vertices i and j belonging to different subtrees and yielding the feasible saving of largest positive value, if any. If no such pair is found, go to Step 3.
- Step 2 : Remove the arc associated with the gate of i and add (i, j) into the solution as the new g_i . Then return to Step 1.
- Step 3 : Remove from the solution the gate of any employee that is assigned to the tail of any route, but that does not own a car (as he/she cannot be a driver).
- Step 4 : Adjust the solution value by adding public transportation costs for all vertices not assigned to a route.

3.4. Computational experiments

All algorithms were implemented in C++ and the computational experiments were performed on a PC with Intel Core i7-3770 3.40 GHz and 8 Gb of RAM. In order to solve the mathematical models we used IBM Cplex 12.6.1 with the default options.

To test the performance of our approaches we created a set of benchmark instances based on the information available in our data sets. Each of the 276 days was considered as an individual instance of the problem. The capacity of the car was set to 4 passengers.

Clearly an employee has to perform two trips every day, one to go to the workplace and another one to return. This allows for some flexibility on how to select which employees can carpool together. Basically, we have two different scenarios. In the first one, namely *full-shift-grouping* (FSG), we group employees that have the exact same shift, i.e., same start time for both trips, and determine a carpool solution for each outgoing trip and each individual group. The total cost of the carpooling plan is then simply doubled, to account for the return trip. In the second scenario, namely *partial-shift-grouping* (PSG), we propose a different solution for each of the two trips, grouping employees that have the same start time in that particular trip. In this case the optimal carpool solution is computed for both trips.

In Table 1, we present average results per weekday for the FSG grouping strategy. For each of our four approaches we show the average CO_2 emissions and the improvement over the

non-carpooling case shown in the first column. We only report average solving time for the formulations, because the heuristic algorithms are practically instantaneous for our benchmark instances. Notice that the exact approaches manage to reduce, in average, around 22% of emissions, which is a great improvement and shows the potential benefits of carpooling. Moreover, the heuristic algorithm H_{TR} managed to significantly reduce CO_2 emissions, showing a surprisingly great potential, yielding high quality solutions very close to the optimal. The poorer performance of H_{DR} might be due to the rather naive greedy approach of only using the regret measure to create solutions. However it still yields reductions of about 11%, in average.

Table 1: Improvement over the non carpooling case for the FSG grouping strategy. CO_2 emissions are shown in kg

	No CP	H_{DR}		H_{TR}		F_{DR}			F_{TR}		
	CO_2	CO_2	imprv	CO_2	imprv	CO_2	imprv	sec	CO_2	imprv	sec
Mon	99.40	89.11	10.4%	79.34	20.2%	78.94	20.6%	0.53	77.67	21.9%	0.48
Tue	97.33	86.41	11.2%	77.22	20.7%	76.72	21.2%	0.53	75.83	22.1%	0.50
Wed	98.12	86.63	11.7%	77.36	21.2%	76.76	21.8%	0.77	75.94	22.6%	0.66
Thu	99.71	89.00	10.7%	79.31	20.5%	78.84	20.9%	0.56	77.90	21.9%	0.48
Fri	100.89	89.78	11.0%	79.87	20.8%	79.42	21.3%	0.53	78.41	22.3%	0.47
Sat	94.61	80.92	14.5%	70.82	25.2%	70.45	25.5%	0.66	69.70	26.3%	0.63
avg	98.34	86.97	11.6%	77.32	21.4%	76.86	21.9%	0.60	75.91	22.8%	0.54

Similarly, Table 2 shows average results for the PSG grouping strategy. Columns have the same meaning as in the aforementioned table. Notice that the gain with this type of strategy is even greater and formulation F_{TR} manages to reach, in average, 28% of savings. Another important remark is that, by analyzing the differences between scenarios FSG and PSG , the results reinforce that the company can improve the efficiency of the carpooling process, by adjusting employees' shifts so as to create larger and denser employees group.

Table 2: Improvement over the non carpooling case for the PSG grouping strategy. CO_2 emissions are shown in kg

	No CP	H_{DR}		H_{TR}		F_{DR}			F_{TR}		
	CO_2	CO_2	imprv	CO_2	imprv	CO_2	imprv	sec	CO_2	imprv	sec
Mon	99.40	84.60	14.9%	73.95	25.6%	72.94	26.6%	2.53	72.11	27.5%	2.26
Tue	97.33	82.00	15.7%	71.93	26.1%	70.86	27.2%	2.82	69.98	28.1%	2.43
Wed	98.12	82.37	16.1%	72.36	26.3%	71.22	27.4%	3.09	70.43	28.2%	2.46
Thu	99.71	84.36	15.4%	74.05	25.7%	72.93	26.9%	3.26	72.04	27.8%	3.10
Fri	100.89	84.91	15.8%	74.39	26.3%	73.37	27.3%	2.90	72.41	28.2%	3.08
Sat	94.61	78.49	17.0%	67.85	28.3%	67.04	29.1%	1.92	66.24	30.0%	1.84
avg	98.34	82.79	15.8%	72.42	26.4%	71.39	27.4%	2.76	70.54	28.3%	2.53

In Figure 5, we present the results of the evaluation of the total CO_2 emission (in kg) per week for each of our approaches. With this analysis, it becomes even more clear the conclusions drew from the previous results, where the gain over the non-carpooling scenario is significant. In addition, we point out that, in the worst case scenario, the potential savings found by F_{TR} ranges in between 113 to 223 kg per week, which corresponds to possible improvement of 23% to 31%. Moreover, by evaluating the total emission in the whole period of 276 days, we found that it is possible to save up to approximately 7.6 tons of CO_2 .

Given that F_{TR} , solved under the PSG scenario, provides a lower bound for all the other approaches, we also evaluate the gap between this approach and the others. This analysis is depicted in Figure 6. It is interesting to notice that, under the PSG scenario, H_{TR} is almost as efficient as formulation F_{DR} . Moreover under FSG sometimes H_{TR} is even better than F_{DR} .

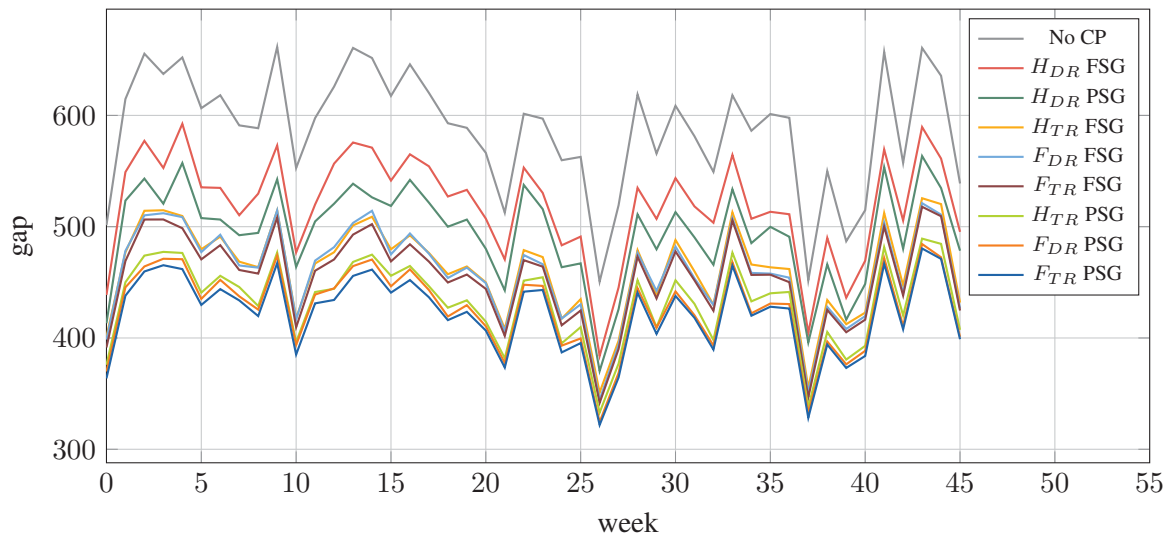
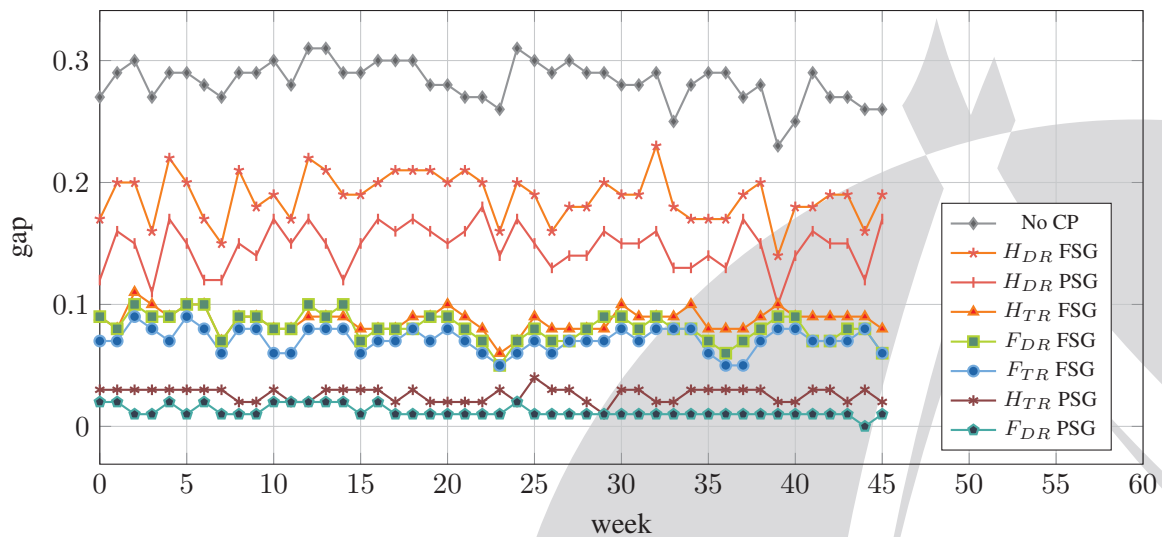

 Figure 5: Weekly CO_2 emissions for different algorithmic solutions.


Figure 6: Gap analysis

4. Web application

The web based application is still under development and a first prototype is being deployed at the company for testing. The application can be mainly divided into two components. The first one is the optimization core, responsible for analyzing the data and providing the possible carpooling groups and routes for users. Similar to Wolfer Calvo *et al.* (2004), this module runs on a daily basis to solve the DCP for the following day. The first prototype is being shipped with the H_{DR} , however in final version of the application the algorithm H_{TR} will be used instead. This will provide us with the opportunity to compare in practice if the difference between both approaches is as great as shown in the computational experiments.

The second component refers to the visual aspect of the application and provides users with an interface in which they are able to interact with the solutions provided by the optimization module and coordinate the actual carpooling with the other participants. In Figure 7 we present one of its screens, in which users are able to request a ride and visualize the route on the map. For privacy reasons the name of the employees used in the example were concealed.

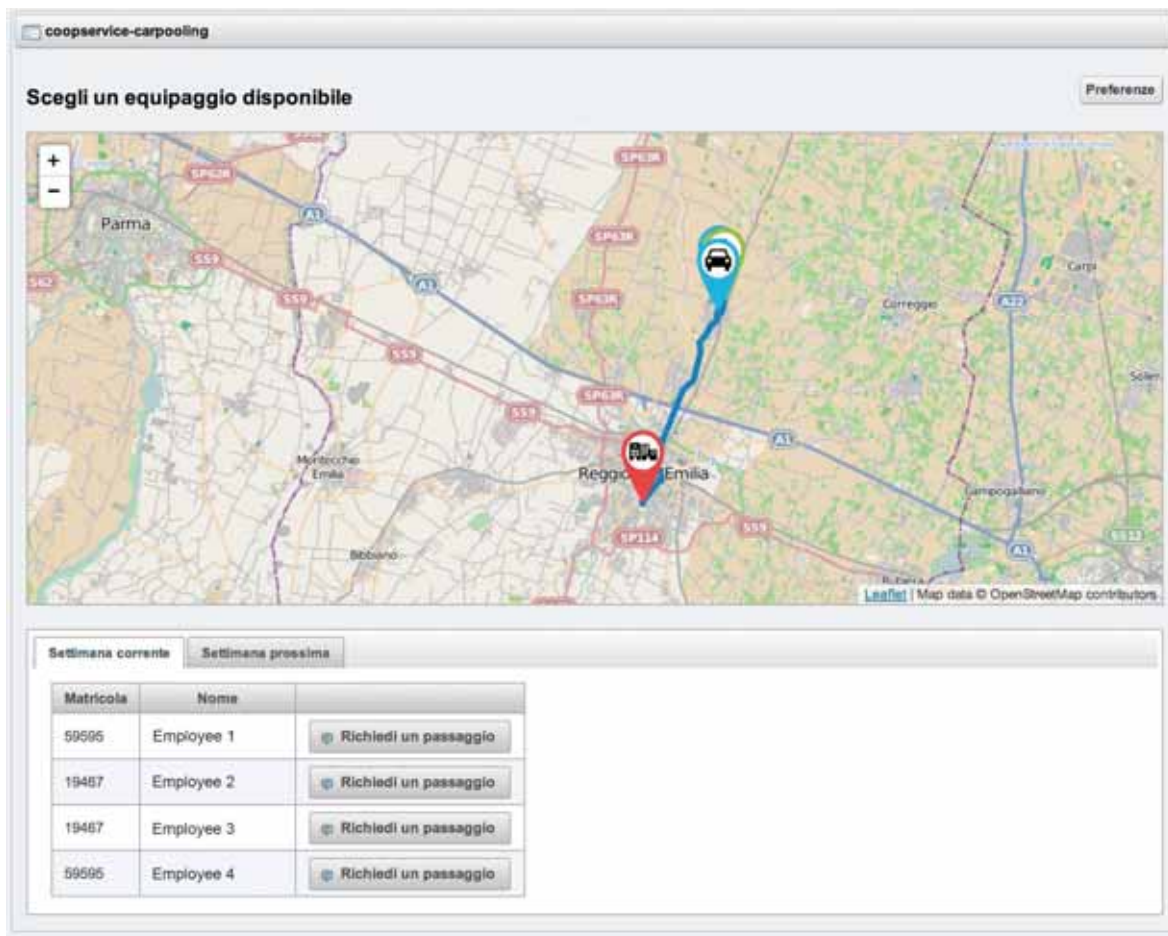


Figure 7: Print screen of the web application prototype

5. Conclusions

In this paper we have approached a practical Daily Carpooling Problem that aims at reducing CO_2 emissions related to car usage of employees of an Italian company. We have proposed two mathematical formulations to define the problem, two heuristic algorithms and a prototype of the application.

The computational experiments on the models indicate a great potential on reducing CO_2 emissions. More specifically, under the FSG grouping strategy the potential savings given by formulation F_{TR} , in average, ranged around 22%, while for the PSG strategy this number is even greater, arriving at 28%. These results also indicate that the company can directly affect the efficiency of the carpooling practice on this scenario by optimizing employees' shifts in order to create more denser groups.

Regarding the heuristics we can notice that, in general, both manage to significantly reduce emissions, specially H_{TR} , which provided solutions very close to the optimal and showed great potential for being applied in practice.

The web application is still under development and a first prototype is being deployed for testing at the company. This prototype includes only the heuristic H_{DR} , however the final version is expected to use the H_{TR} heuristic in order to propose better and more flexible solutions. This will allow us to perform a posterior analysis to validate the results of our computational experiments, checking whether or not they hold in practice.

Acknowledgements

The authors would like to thank Coopservice S.coop.P.A for providing the data and the financial support for the development of this project. Manuel Iori acknowledges financial support also by Capes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) under grant PVE no. A007/2013.

References

- Baldacci, R., Maniezzo, V. e Mingozzi, A.** (2004), An Exact Method for the Car Pooling Problem Based on Lagrangean Column Generation. *Operations Research*, v. 52, n. 3, p. 422–439.
- Esau, L. R. e Williams, K. C.** (1966), On teleprocessing system design. Part 2. *IBM System Journal*, v. 5, n. 3, p. 142–147.
- Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X. e Koenig, S.** (2013), Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, v. 57, p. 28–46.
- Gärling, T. e Friman, M.** (2015), Unsustainable travel becoming (more) sustainable. p. 163–177.
- Pisinger, D. e Ropke, S.** (2007), A general heuristic for vehicle routing problems. *Computers & Operations Research*, v. 34, n. 8, p. 2403–2435.
- Rietveld, P., Zwart, B., van Wee, B. e van den Hoorn, T.** (1999), On the relationship between travel time and travel distance of commuters. *The Annals of Regional Science*, v. 33, n. 3, p. 269–287.
- Teal, R.** (1987), Carpooling: who, how and why. *Transportation Research Part A: General*, v. 21, n. 3, p. 203–214.
- United Nations** Millennium Development Goals. Available at <http://www.un.org/millenniumgoals/>, 2015.
- Wolfler Calvo, R., de Luigi, F., Haastруп, P. e Maniezzo, V.** (2004), A distributed geographic information system for the daily car pooling problem. *Computers & Operations Research*, v. 31, n. 13, p. 2263–2278.
- Yan, S., Chen, C. e Lin, Y.** (2011), A model with a heuristic algorithm for solving the long-term many-to-many car pooling problem. *Intelligent Transportation Systems, IEEE Transactions on*, v. 12, n. 4, p. 1362–1373.