



Multi-day fair collaboration in demand-responsive transportation

E. Angelelli¹ · V. Morandi¹  · M. G. Speranza¹

Accepted: 2 September 2023 / Published online: 28 September 2023
© The Author(s) 2023

Abstract

In this paper, we consider the case of companies that offer a demand-responsive transportation service, such as a shared-taxi service, and are engaged in a horizontal collaboration initiative. The goal of the coalition is to optimize the transportation operations in such a way that no company is penalized, in terms of customers served and/or working time. We present an optimization model for a multi-day planning horizon that includes constraints aimed at guaranteeing a level of fairness to all companies that can be controlled over the planning horizon and day-by-day, if beneficial. An adaptive large neighborhood search heuristic is then presented for its solution. The computational experiments show that, although the model constraints the optimization space, it still guarantees substantial savings. Moreover, they show that the model is flexible and can guarantee the sustainability in the long term of the collaboration initiative.

Keywords Collaboration · Demand-responsive transportation · Fairness · Multi-day time horizon

1 Introduction

Transportation operations, of people and goods, are well known to be responsible for a large share of the air pollutants. Technological developments are modifying in different directions the way we move and transport goods. One of these directions aims at reducing the number of kilometers traveled by road vehicles and/or at increasing the used capacity of a vehicle. Examples of initiatives in this direction that concern the transportation of people are ride-sharing, car sharing, mobility-as-a-service (MAAS) (see Mourad et al. 2019) for a comprehensive survey on optimization models for shared mobility). The expressions Demand-Responsive Transport and Flexible Transport Services are often used to collectively indicate this class of services (see Vansteenwegen et al. 2022; Nelson and Wright 2021). While the potential of collaboration initiatives to reduce the number of vehicles on the road is recognized by many, there are also several obstacles that are known to limit the number of successful collaborations (see, for example, Basso et al. 2019).

✉ V. Morandi
valentina.morandi1@unibs.it

¹ University of Brescia, Brescia, Italy

It is well-known that collaboration initiatives may fail because larger companies benefit more than smaller companies from the collaboration and smaller companies may lose customers or even be pushed out of the market (see Cruijssen et al. 2007) for the results of a large survey). In fact, the unconstrained optimization of the exchange of customers, that has as objective the minimization of the total alliance costs, could tend to benefit larger companies and to be detrimental for smaller companies. One of the challenges in the design of successful collaboration schemes is to find mechanisms that make the collaboration beneficial to all the companies involved (see Gansterer and Hartl 2018; Gansterer et al. 2020). We will say that a collaboration scheme is *fair* when it contains such mechanisms. The literature on horizontal collaboration in the transportation area is wide. We refer to Aloui et al. (2021) for a literature review on collaborative sustainable transportation.

Some contributions have appeared that constrain the optimization to be fair towards all companies. In Fernández et al. (2016) this goal is modeled by means of a lower bound on the profit of each company. In Gansterer and Hartl (2021) lower bounds are set on the profit and also on the number of customers served by each company. In Angelelli et al. (2022) the workload exchange is controlled by means of constraints on the number of served customers and on the time traveled by each company. In these papers, the customers exchange is optimized on a daily basis, that is, it is assumed that each day, given the customers of all companies, a constrained optimization model is run and the solution implemented. We observe that controlling the exchanges on a daily basis may fail to exploit opportunities when the planning horizon contains more than one day, as the illustrative example provided later shows. In this context, a multi-day optimization model offers more flexibility, potential for savings and, at the same time, can guarantee fairness in the long run.

In this paper, we consider an extension of the problem approached in Angelelli et al. (2022). According to the literature (see for example Gansterer and Hartl 2018) and Fernández et al. (2016)), we use the term ‘coalition’ to indicate a set of companies involved in a collaboration initiative. A set of companies involved in a shared taxi service have agreed to form a coalition and to exchange customers over a multi-day horizon. On each day of the planning horizon a company may *concede* the service of some of its customers to other companies and may *acquire* customers from other companies.

According to the idea that every company would consider fair to keep a customer exchange that does not modify the workload, different unfairness measures could be considered:

- The *daily customer balance* of a company in a day of the time horizon: the percentage difference between the number of acquired and conceded customers on that day w.r.t. the original number of customers in that day;
- The *cumulated customer balance* of a company up to a day of the time horizon: the percentage difference between the number of acquired and conceded customers w.r.t. the original number of customers up to that day;
- The *daily time balance* of a company in a day of the time horizon: the percentage difference between the acquired and conceded travel time on that day w.r.t. the original travel time in that day;

- The *cumulated time balance* of a company up to a day of the time horizon: the percentage difference between the acquired and conceded travel time w.r.t. the original travel time up to that day.

The nature of the problem is multi-objective. On one side, we want to minimize the overall cost for the alliance in the time horizon. On the other side, we want to reduce the unfairness for each company as much as possible. We approach the problem through an optimization model that assigns customers to companies and identifies the route for each vehicle so that the total routing cost over the time horizon is minimized while unfairness measures for the coalition are bounded (namely, maximum daily customer balance registered by any company on any day of the time horizon, maximum cumulated customer balance registered by any company on the last day, maximum daily time balance for any company on any day of the time horizon, maximum cumulated time balance on last day).

The model we present is able to coordinate decisions among days in the time horizon and obtains better results with respect to the application of a single-day model to each day of the horizon.

A heuristic that follows the Adaptive Large Neighborhood Search (ALNS) algorithmic approach is presented for the solution of the optimization model.

1.1 Motivation and assumptions

Demand-responsive transportation services are usually divided in two streams: services for which the booking is mandatory and has to be done in advance (e.g., dial-a-ride services, transit to hospitals, school services, etc.) and services that can be requested at any time (e.g., taxi services). Our context is the former, where bookings can be booked days in advance. The proposed scheme could be applied to working periods that span from few days to weeks, depending on the nature of services involved.

While the topic of profit distribution among the companies involved in a collaboration initiative is very important, we will not treat this aspect in this paper, as it can be seen as a further a posteriori mechanism of the collaboration scheme.

1.2 An illustrative example

To illustrate the benefits of horizontal collaboration over a multi-day horizon, a 2-day example of two companies is presented where the depot and customer locations of the two companies (5 customers each for the red and blue companies) are depicted in Figs. 1, 2, 3 and 4. For each customer c , the pick-up location is denoted by O_c while the delivery location is denoted by D_c . For the sake of simplicity, in this small example, customers and depots are the same in both days and we take into account only daily and cumulated customer balance.

We consider the following collaboration schemes:

- the Non-Collaboration (NC) scheme: each company serves its own customers; no exchange is allowed at any time (optimal plan in Fig. 1);

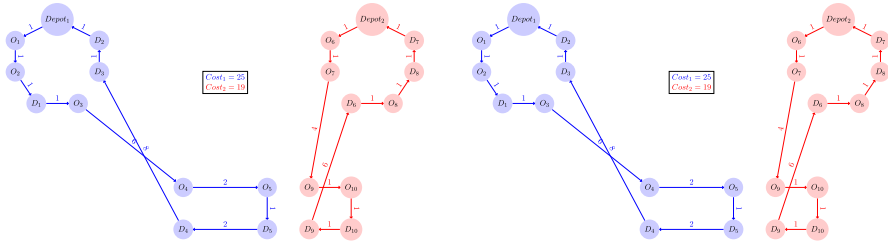


Fig. 1 NC scheme plan in day 1 (left) and day 2 (right)

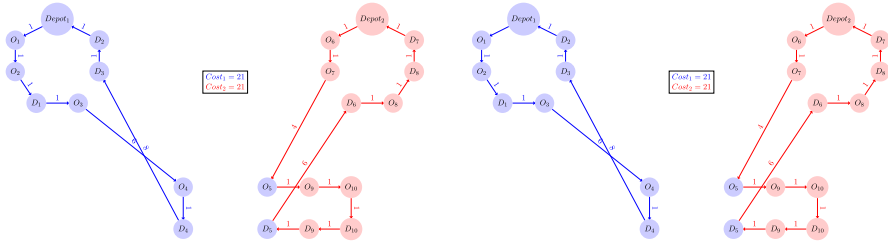


Fig. 2 1-Day Collaboration scheme plan in day 1 (left) and day 2 (right)

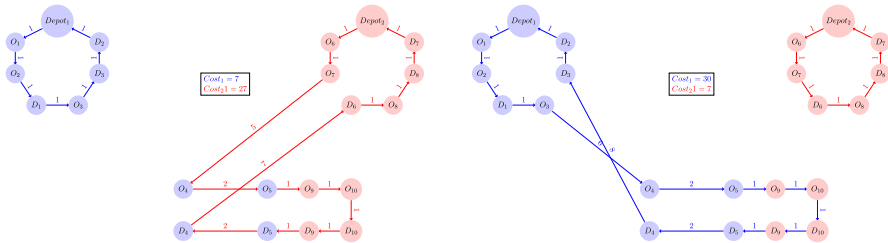


Fig. 3 2-Day Collaboration scheme plan in day 1 (left) and day 2 (right)

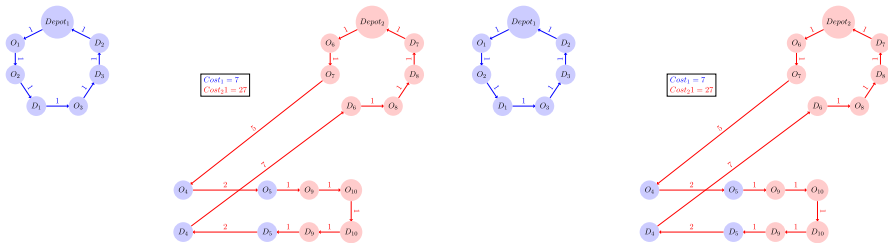


Fig. 4 UC scheme plan in day 1 (left) and day 2 (right)

- the 1-Day Collaboration scheme: the daily customer balance, in absolute value, is bounded to be lower than 20% on each day for each company (optimal plan in Fig. 2);

- the 2-Day Collaboration scheme: the cumulated customer balance on the last day, in absolute value, is bounded to be lower than 20% for each company, while no bound is applied on single days (optimal plan in Fig. 3);
- the Unconstrained Collaboration (UC) scheme: each company could serve any customer without any bound on daily or cumulated customer balance (optimal plan in Fig. 4).

In Table 1, we summarize the main features of the plans produced by the three collaboration schemes with respect to the non-collaboration scheme NC (shown as percentages). Savings are computed, for each company and for the entire coalition (column All), as the percentage reduction of the cost with respect to the non-collaboration cost. More details on these plans are reported in Appendix A.

All collaboration schemes realize savings for the coalition with respect to the NC scheme and, as expected, the UC realizes the maximum savings (+ 22.7%) while the 1-Day scheme ranks last (+ 4.5%). Interestingly enough, not only does the 2-Day scheme produce an overall saving quite close to the UC (+20.5%), but also a positive saving for both companies (+ 28% for blue company and + 10.5% for the red company).

In Fig. 5, a general overview of the results is proposed. Three performance indicators have been considered for each scheme: the savings, the maximum daily customer balance registered by companies on the 2 days, and the maximum cumulated customer balance registered by companies on the second day. The savings are located on the x axis, while the maximum daily customer balance is located on the y axis of the left panel, and the maximum cumulated customer balance is located on the y axis of the right panel. While the UC allows the maximum savings, it also provides the highest unfairness (40% for both daily and cumulated customer balance).

A compromise solution is represented by the 1-Day scheme, for which the maximum daily and cumulated customer balance is halved with respect to the UC, i.e. 20%. However, this comes with a poor performance in terms of saving as the total cost is more similar to NC scheme than to the UC.

The 2-Day scheme outperforms the other ones as it produces savings that are just slightly less than those produced by the UC, and a final maximum cumulated customer

Table 1 Savings, daily and cumulated customer balance for each company (%)

Scheme	1-Day			2-Day			UC		
	Blue	Red	All	Blue	Red	All	Blue	Red	All
Savings	16.0	- 10.5	4.5	28.0	10.5	20.5	72.0	- 42.0	22.7
Daily customer balance									
On Day 1	- 20	20		- 40	40		- 40	40	
On Day 2	- 20	20		40	- 40		- 40	40	
Cumulated customer balance	- 20	20		0	0		- 40	40	

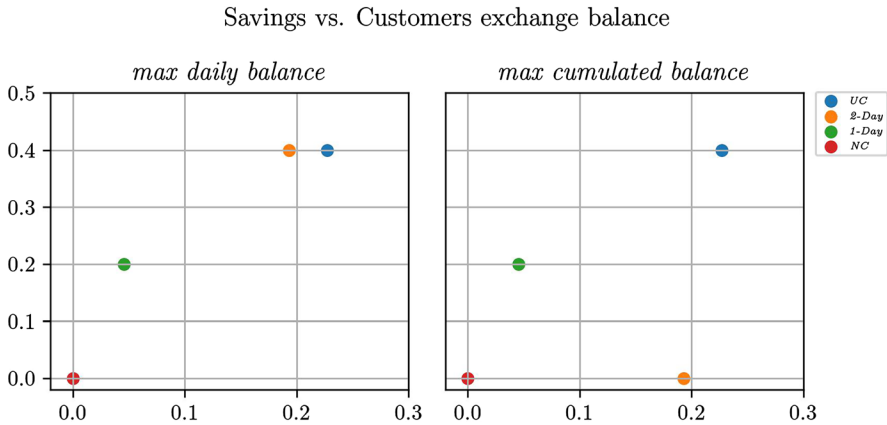


Fig. 5 Savings (x-axis) versus maximum daily and maximum cumulated customer balance (y-axis) for different collaboration schemes

balance equal to zero which definitely dominates the 1-Day scheme. This result comes at the cost of a higher value of the maximum daily customer balance for both the companies that is averaged out through the days of the horizon.

1.3 Contribution of the paper

The main contribution of the paper is the introduction of a flexible optimization model able to model a number of different approaches (collaboration schemes) to the multi-day horizon context. On one side, the model we propose can be used to implement a day-by-day approach where, in line with Angelelli et al. (2022), independent optimizations are executed for every day in the time horizon. On the other side the model can be used to implement a global optimization on the whole time horizon where only the cumulated exchange operations up to the last day are taken into account and bounded. A number of midway schemes can be implemented where both day-by-day and final cumulated exchanges are bounded. An effective ALNS algorithm has been developed for this general model. The model is then used to assess the impact of bounding unfairness in the long-run (multiple days) versus different levels of short-run (daily) unfairness bounding. We provide insights to the management in order to cope with the trade-off between cost and unfairness.

1.4 Structure of the paper

The remainder of the paper is organized as follows. In Sect. 2 the problem studied is presented together with the optimization model. In Sect. 3 the ALNS heuristic is described. Section 4 is devoted to the computational experiments and the results. Finally, in Sect. 5, some concluding remarks are provided.

2 Problem definition and modelization

The problem we deal with can be defined as follows. A set of companies agree to form a coalition offering a shared taxi service where customers book in advance the transportation on a given day of a fixed multi-day time horizon. We call *request* the booking, made at a specific company, of a transportation service for a number of customers on a specific day of the time horizon. The request must be accomplished within given time windows on the pickup and delivery time at origin and destination, respectively. A maximum allowed on-board time (including pick-up and drop-off service time) is known for each request. Each company has a fleet of vehicles of given capacity that have to accomplish their daily routes in a fixed amount of time. On each day of the planning horizon, a company may concede the service of some of its requests to other companies and may acquire requests from other companies. The management has several objectives: minimizing the overall cost of planned routes and minimizing the daily and the cumulated customer and time balance for each company on each day. We model the problem as an optimization problem where the objective is the minimization of the overall routing cost, and bounds are applied to the absolute values of:

- maximum daily customer balance and daily time balance realized by any company over the time horizon;
- maximum cumulated customer balance and cumulated time balance realized by any company on the last day of the time horizon.

Next, we formalize the notation and model the problem as a MILP.

For the sake of simplicity, in the following we describe the set of constraints for a single day f in the horizon $F = \{1, \dots, |F|\}$ and omit the index f from the notation. Later, we will introduce the index f only for those quantities that have a time-dependent role in the multi-day problem (i.e cost, and time and customer balance).

A set of requests C_m is associated with each company $m \in M$, and $C = \bigcup_{m \in M} C_m$ is the set of all requests. Each request $c \in C$ has the following attributes: an origin o_c (pick-up point), a destination d_c (drop-off point), a demand p_c (number of customers that travel from o_c to d_c), a minimum travel time t_c (direct transfer from o_c to d_c), a service time s'_c and s''_c at origin and destination, a time window $[e'_c, l'_c]$ and $[e''_c, l''_c]$ at origin and destination, respectively, and a maximum time allowed on-board T_c (obviously, $T_c \geq t_c$).

From now on, the term 'travel time' for a request is intended as the minimum traveling time t_c from origin to destination of customer c . We denote as $O = \{o_c \mid c \in C\}$ and $D = \{d_c \mid c \in C\}$ the set of all origins and destinations, respectively.

Each company $m \in M$ has the following attributes: an origin depot h'_m and a destination depot h''_m , which we denote together as $H_m = \{h'_m, h''_m\}$, and a set of vehicles K_m . We denote as $H = \bigcup_{m \in M} H_m$ the set of all depots and as $K = \bigcup_{m \in M} K_m$ the

overall coalition fleet, where each vehicle $k \in K$ has capacity Q^k and maximum route duration T^k .

We consider a complete directed graph $G = (V, A)$ where $V = H \cup O \cup D$. Each node $i \in V$ represents either a depot or an origin or a destination, while arc $(i, j) \in A$ represents a direct transfer from node i to node j with an associated traveling time t_{ij} and cost c_{ij} . Each node $i \in V$ representing an origin for a request c , i.e. $i = o_c$ for some $c \in C$, inherits service time $s_i = s'_c$, time window $[e_i, l_i] = [e'_c, l'_c]$, and has an associated flow quantity $q_i = p_c$. Similarly, each node $i \in V$ representing a destination for a request c , i.e. $i = d_c$ for some $c \in C$, inherits service time $s_i = s''_c$, time window $[e_i, l_i] = [e''_c, l''_c]$, and has an associated flow quantity $q_i = -p_c$. Finally, each node $i \in V$ representing a depot for a company m , i.e. $i = h'_m$ or $i = h''_m$ for some $m \in M$, has service time $s_i = 0$, time window $[e_i, l_i] = [0, \infty]$, and a flow quantity $q_i = 0$.

The decision variables are:

- binary variables y_c^k : a variable takes value 1 if request $c \in C$ is served by vehicle $k \in K$, and 0 otherwise;
- binary variables x_{ij}^k : a variable takes value 1 if arc $(i, j) \in A$ is traversed by vehicle $k \in K$, and 0 otherwise;
- integer variables w_i^k : a variable represents the number of on-board customers when vehicle $k \in K$ leaves node $i \in V$;
- continuous variables u_i^k : a variable represents the arrival time of vehicle k at node $i \in V$;
- continuous variables r_c^k : a variable represents the time spent by request $c \in C$ on-board vehicle k (excluding service time);
- continuous variable $Cost$ represents the routing cost.

The notation for the constraints, that have to be satisfied for each day $f \in F$, is summarized in Table 2.

$$Cost = \sum_{m \in M} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K_m} c_{ij} x_{ij}^k \tag{1}$$

$$\sum_{j \in V} x_{h'_m j}^k = \sum_{i \in V} x_{i h''_m}^k = 1 \quad m \in M \quad k \in K_m \tag{2}$$

$$\sum_{i \in V} x_{i, l}^k - \sum_{j \in V} x_{l, j}^k = 0 \quad l \in V \quad k \in K \tag{3}$$

$$\sum_{k \in K} y_c^k = 1 \quad c \in C \tag{4}$$

$$\sum_{j \in V} x_{o_c j}^k = y_c^k \quad c \in C \quad k \in K \tag{5}$$

Table 2 Notation—Part I

<i>Sets</i>	
M	Set of companies in the coalition
C_m	Set of requests of company $m \in M$
C	Set of all requests, $C = \bigcup_{m \in M} C_m$
$H_m = \{h'_m, h''_m\}$	Starting and ending depot, for company $m \in M$
H	Set of all depots $H = \bigcup_{m \in M} H_m$
K_m	Set of vehicles of company $m \in M$
K	Set of all vehicles, $K = \bigcup_{m \in M} K_m$
<i>Requests attributes</i>	
o_c	Origin of request $c \in C$
d_c	Destination of request $c \in C$
O	Set of origins $O = \{o_c \mid c \in C\}$
D	Set of destinations $D = \{d_c \mid c \in C\}$
p_c	Number of customers in request $c \in C$
t_c	Minimum travel time (direct transfer) from o_c to d_c of request $c \in C$
s'_c	Service time at origin o_c for request $c \in C$
s''_c	Service time at destination d_c for request $c \in C$
T_c	Maximum time allowed on-board for request $c \in C$
$[e'_c, l'_c]$	Time window for service at origin o_c for request $c \in C$
$[e''_c, l''_c]$	Time window for service at destination d_c for request $c \in C$
<i>Vehicles attributes</i>	
Q^k	Capacity of vehicle $k \in K$
T^k	Maximum route duration for vehicle $k \in K$
<i>Graph attributes</i>	
$V = H \cup O \cup D$	Set of vertices
A	Set of arcs
t_{ij}	Traveling time on arc $(i, j) \in A$
c_{ij}	Cost of arc $(i, j) \in A$
s_i	Service time in node $i \in V$
$[e_i, l_i]$	Time window for service at node $i \in V$
q_i	Number of customers at node $i \in V$
<i>Decision variables</i>	
y_c^k	1 if request $c \in C$ is served by vehicle $k \in K$, 0 otherwise
x_{ij}^k	1 if arc $(i, j) \in A$ is used by vehicle $k \in K$, 0 otherwise
w_i^k	Number of customers on vehicle $k \in K$ after visiting node $i \in V$
u_i^k	Arrival time of vehicle $k \in K$ at node $i \in V$
r_c	Maximum on-board time of request $c \in C$

$$\sum_{j \in V} x_{o_c j}^k - \sum_{i \in V} x_{i d_c}^k = 0 \quad c \in C \quad k \in K \quad (6)$$

$$u_j^k \geq u_i^k + s_i + t_{ij} - U_{ij}(1 - x_{ij}^k) \quad i, j \in V \quad k \in K \quad (7)$$

$$r_c^k = u_{d_c}^k - (u_{o_c}^k + s_{o_c}) \quad c \in C \quad k \in K \quad (8)$$

$$u_{h_m'}^k - u_{h_m''}^k \leq T^k \quad m \in M \quad k \in K_m \quad (9)$$

$$e_i \leq u_i^k \leq l_i \quad i \in V \quad k \in K \quad (10)$$

$$t_{o_c d_c} \leq r_c^k \quad c \in C \quad k \in K \quad (11)$$

$$r_c^k \leq T_c \quad c \in C \quad k \in K \quad (12)$$

$$w_j^k \geq w_i^k + q_i - W_i^k(1 - x_{ij}^k) \quad i, j \in V \quad k \in K \quad (13)$$

$$\max\{0, q_i\} \leq w_i^k \leq \min\{Q^k, Q^k + q_i\} \quad i \in V \quad k \in K \quad (14)$$

$$w_{h_m'}^k = 0 \quad m \in M \quad k \in K_m \quad (15)$$

$$y_c^k \in \{0, 1\} \quad k \in K \quad c \in C \quad (16)$$

$$x_{ij}^k \in \{0, 1\} \quad k \in K \quad i \in V \quad j \in V \quad (17)$$

$$w_i^k \in \mathbb{N} \quad k \in K \quad i \in V \quad (18)$$

$$u_i^k \geq 0 \quad k \in K \quad i \in V. \quad (19)$$

Equation (1) defines the daily cost of operations. Constraints (2)–(6) define the routing constraints of the model. Constraints (2) guarantee that each vehicle $k \in K_m$ traverses exactly one arc exiting from the origin depot h_m' and exactly one arc entering the destination depot h_m'' allowing only one trip per day to each vehicle. Constraints (3) guarantee flow conservation through nodes. Constraints (4) guarantee that each request $c \in C$ is served by exactly one vehicle of the coalition. Constraints (5) guarantee that vehicle k exits origin node o_c if and only if it serves request $c \in C$. Constraints (6) guarantee that vehicle k visits origin node o_c if and only if it also visits destination node d_c . Subtour elimination constraints are implied by constraints on

traveling times (7)–(12). In particular, constraints (7) ensure the consistency of the arrival time at node j visited by vehicle k after node i , provided that the constant U_{ij} is such that $U_{ij} \geq \max\{0, l_i + s_i + t_{ij} - e_j\}$. Constraints (8) evaluate the on-board time r_c^k for each request $c \in C$ as the difference between the arrival time of vehicle $k \in K$ at destination node d_c and the departure time (arrival time plus service time) from origin node o_c . Constraints (9) bound with T^k the route duration for each vehicle k . Constraints (10) implement the time window constraints for the visited nodes. Constraints (11), together with constraints (8), enforce precedence constraints between origin and destination of each request as, by imposing r_c^k non-negative, we enforce $u_{d_c}^k$ in being larger than $u_{o_c}^k$. Furthermore, according to Cordeau (2006), bounding r_c^k to be greater than t_{o_c, d_c} gives a stronger formulation. Constraints (12) bound with T_c the on-board time of each request $c \in C$. Capacity constraints are implemented in (13)–(15). In particular, constraints (13) ensure the consistency of vehicle load at nodes visited by vehicle k provided that constant W_i^k are such that $W_i^k \geq \min\{Q^k, Q^k + q_i\}$. Constraints (14) ensure the respect of vehicle capacity. Constraints (15) set the initial load of each vehicle k . Finally, constraints (16)–(19) define the variables domains.

In the following, we move from the single-day horizon to the multi-day horizon context. We formally introduce the daily and multi-day fairness constraints and present the complete multi-day optimization model. To this aim, we index quantities and variables that depend on the day of the horizon with the superscript $f \in F$. For example, C_m^f is the set of requests of company $m \in M$ in day f and $Cost^f$ is the total cost in day f . To avoid a cumbersome notation, we do not make a detailed list of the quantities and variables that depend on the day f . Constraints (1)–(19) will be imposed on each day f , for all $f \in F$.

Table 3 Notation—Part II

<i>Sets</i>	
F	Set of days in the time horizon
<i>Companies requests and balances</i>	
C_m^f	Set of requests of company $m \in M$ in day $f \in F$
B_m^f	cumulated customer balance for company $m \in M$ on any day $f \in F$
S_m^f	cumulated time balance for company $m \in M$ on any day $f \in F$
<i>Fairness parameters</i>	
α^{DC}	Maximum percentage daily customer balance allowed for any company $m \in M$ on any day $f \in F$
α^{DT}	Maximum percentage daily time balance allowed for any company $m \in M$ on any day $f \in F$
α^{WC}	Maximum overall percentage cumulated customer balance allowed for company $m \in M$ on last day $f = F $
α^{WT}	Maximum overall percentage cumulated time balance allowed for company $m \in M$ on last day $f = F $
<i>Cost</i>	
$Cost^f$	Cost in day $f \in F$

The objective function, to be minimized, is the total cost, that is the sum, over the time horizon, of the daily costs $\sum_{f \in F} Cost^f$, where $Cost^f$ is defined in (1). In Table 3 we summarize only the notation dependent on the day f that we will explicitly use. The level of unfairness in terms of customers and time exchange is measured through the daily and the cumulated customers and time balances, whose absolute values will be constrained by upper bounds to reduce unfairness among companies.

According to definition, the *daily customer balance* for company $m \in M$ on day $f \in F$ is computed as

$$B_m^f = \frac{\sum_{k \in K_m} \sum_{c \in C \setminus C_m^f} P_c y_c^k - \sum_{k \in K \setminus K_m} \sum_{c \in C_m^f} P_c y_c^k}{\sum_{c \in C_m^f} P_c},$$

and the *daily time balance* for company $m \in M$ on day $f \in F$ is computed as

$$S_m^f = \frac{\sum_{k \in K_m} \sum_{c \in C \setminus C_m^f} t_c y_c^k - \sum_{k \in K \setminus K_m} \sum_{c \in C_m^f} t_c y_c^k}{\sum_{c \in C_m^f} t_c}.$$

In order to measure balances in the long-run and according to the definition we gave in the introduction, the *cumulated customer balance* for company $m \in M$ until day $f \in F$ is computed as

$$cumB_m^f = \frac{\sum_{j \leq f} (\sum_{k \in K_m} \sum_{c \in C \setminus C_m^f} P_c y_c^k - \sum_{k \in K \setminus K_m} \sum_{c \in C_m^f} P_c y_c^k)}{\sum_{j \leq f} \sum_{c \in C_m^f} P_c},$$

and the *cumulated time balance* for company $m \in M$ until day $f \in F$ is computed as

$$cumS_m^f = \frac{\sum_{j \leq f} (\sum_{k \in K_m} \sum_{c \in C \setminus C_m^f} t_c y_c^k - \sum_{k \in K \setminus K_m} \sum_{c \in C_m^f} t_c y_c^k)}{\sum_{j \leq f} \sum_{c \in C_m^f} t_c}.$$

We recall that $cumB_m^{|F|}$ and $cumS_m^{|F|}$ refer respectively to the *cumulated customer balance* and to the *cumulated time balance* for company $m \in M$ at the end of the time horizon.

The complete model is then formulated as follows:

$$\min z = \sum_{f \in F} Cost^f \tag{20}$$

+ constraints (1)–(19) $f \in F$

$$\max_{m \in M, f \in F} |B_m^f| \leq \alpha^{DC} \tag{21}$$

$$\max_{m \in M, f \in F} |S_m^f| \leq \alpha^{DT} \tag{22}$$

$$\max_{m \in M} |cum B_m^{|F|}| \leq \alpha^{WC} \quad (23)$$

$$\max_{m \in M} |cum S_m^{|F|}| \leq \alpha^{WT} \quad (24)$$

Observe that constraints (2)–(19) are replicated for each day of the time horizon $f \in F$.

To control unfairness on a daily basis, Constraint (21) bounds the maximum, among all companies $m \in M$ and all days $f \in F$, of the absolute value of the daily customers balance to be not greater than a fixed value α^{DC} . Constraint (22) is its daily time balance counterpart.

To control unfairness on the long-run, Constraint (23) bounds the maximum, among all companies $m \in M$, of the absolute value of the cumulated customers balance on the last day of the time horizon $f = |F|$ to be not greater than the fixed value α^{WC} . Constraint (24) is its cumulated time balance counterpart.

Constraints (21) can be easily linearized with the matrix of constraints

$$-\alpha^{DC} \leq B_m^f \leq \alpha^{DC} \quad m \in M, f \in F.$$

Constraints (22)–(24) can be linearized in the same way.

3 The algorithm

The Adaptive Large Neighborhood Search (ALNS) algorithmic approach has been widely used in the literature, especially for routing and scheduling problems (see Muller 2009; Pisinger and Ropke 2010 and references therein). The ALNS scheme, first proposed in Ropke and Pisinger (2006), is an extension of the Variable Neighbourhood Search (VNS, see Mladenović and Hansen 1997 for details) in which a number of destroy and repair operators are iteratively and pseudo-randomly applied in order to improve the current solution. The underlying metaheuristic scheme adopted is the simulated annealing (as in Ropke and Pisinger 2006) and the random choice of destroy and repair operators is guided by the success frequency they had in previous iterations.

The general ALNS heuristic is outlined in Algorithm 1. To start, the ALNS algorithm needs a feasible solution x_0 , which is used to initialize the current solution x and the incumbent solution x^* , along with a pool of destroy and a pool of repair operators. In our case, the initial solution is the non collaborative solution, i.e. each company serves its own customers. Since the ALNS uses the simulated annealing as underlying metaheuristic, two parameters are needed: T_{max} , representing the starting temperature, and γ , representing the cooling rate at each iteration. At each iteration, in order to improve the current solution x , a destroy operator and a repair operator are randomly drawn from their respective pools. Each operator has a probability to be chosen which depends on how many times the operator has been effective, in past iterations, in finding improving solutions. The drawn destroy operator removes q requests from the current solution x and, then, the selected repair operator reconstructs a feasible, possibly improved, solution x' from the destroyed one. The repaired solution might replace the current solution

depending on a probability defined according to the simulated annealing paradigm. Every R iterations in which a new incumbent solution is found, the probability to be chosen for each operation is reset to default values, i.e. equal probability to be selected.

Algorithm 1: ALNS scheme

input : $x_0, T_{\max}, \gamma, R, q$
output: x^* heuristic solution

- $x = x_0$ as current solution and $x^* = x_0$ incumbent solution;
- Initialize r and w to zero;
- Set destroy and repair operators' scores to initial values;

while $T_{\max} > 1$ **do**

- Draw a destroy and a repair operator;
- Destroy the current solution x ;
- Repair the destroyed solution and obtain x' ;
- **if** $cost(x') \geq cost(x^*)$ **then**
 - └ $-w = w + 1$;
- u randomly drawn in $U(0, 1)$;
- if** $u < e^{-\frac{cost(x^*) - cost(x')}{T_{\max}}}$ **then**
 - $x = x'$;
 - if** $cost(x') < cost(x^*)$ **then**
 - $x^* = x'$;
 - **if** $r > R$ **then**
 - └ $-r = 0$;
 - └ Set destroy and repair operators' scores to initial values;
 - else**
 - └ Update destroy and repair operators' scores;
 - └ $-r = r + 1$;
- └ $T_{\max} = T_{\max} * \gamma$;

- **return** x^* ;

In the ALNS heuristic we propose six destroy and three repair operators have been used. The destroy operators are the random removal, the worst removal, the related removal, the proximity removal, the closeness removal, and the interchangeability removal. The former three destroy operators resemble the original ones proposed in the seminal work of Ropke and Pisinger (2006), the fourth operator is inspired by the one introduced by Molenbruch et al. (2017) and the fifth and sixth destroy operators are inspired by the ones proposed in Angelelli et al. (2022). On the other hand, the selected repair operators are the best insertion, the k-regret insertion (with $k = 2, 3$ and 4) and the closeness insertion. The former two repair operators follow the original ones suggested as performing operators for routing problems in the seminal work of Ropke and Pisinger (2006). The third repair operator is derived by the one proposed in Angelelli et al. (2022). In the following, we describe how these operators have been adapted to the multy-day horizon scenario.

Destroy operators. Each destroy operator takes in input the current solution along with the degree of destruction q . Each destroy operator removes the first of the q requests according to the operator rule and, then, removes the remaining requests among same-day requests. Removing same-day customers fosters the

customers exchange among companies as they are allowed to be exchanged, if beneficial. On the contrary, removing a (usually small) number of requests to be served in different days could prevent requests to be easily exchanged as each request cannot be postponed or anticipated.

Repair operators. Given a destroyed solution, in which $|C| - q$ requests are feasibly served and q requests have been removed, repair operators try to feasibly insert excluded requests to routes. A selection probability, according to the operator rules, will be assigned to each destroyed request and, when one is selected to be repaired, the insertion procedure is guided towards feasibility. Once a request has been selected to be inserted in the destroyed solution, a list of eligible companies $M_{Eligible}$ is searched for such that:

- the assignment of the request to the company does not violate the upper bound on the cumulated and/or daily constraints on customers and time,
- after the assignment, the requests still to be assigned are able to cover the possible total violation of the lower bound of cumulated and/or daily constraints on customers and time.

Among the eligible companies $m \in M_{Eligible}$, we search for companies with violated lower bounds on the cumulated and/or daily constraints on customers and time, creating set M_{Likely} . For feasible insertion of a request in the solution, we mean the one proposed in Molenbruch et al. (2017), assessing the feasibility with respect to the maximum trip time of each request and of the involved vehicles. If M_{Likely} is not empty, then the request is feasibly assigned at the minimum cost among these companies. If M_{Likely} is empty or there exists no feasible insertion in such companies, the request is feasibly assigned at minimum cost among companies $m \in M_{Eligible}$. In case no feasible insertion in $M_{Eligible}$ can be found, then the solution we had before destruction is restored. Otherwise, we iterate by selecting a new request to be inserted until all removed requests have been reinserted.

4 Computational analysis

In this section, a computational study is presented to assess the effectiveness of the proposed model and the impact of parameters used to control the level of fairness of the collaboration. In Sect. 4.1, we describe the benchmark instances used. The different tested settings and the collected statistics are presented in Sects. 4.2 and 4.3, respectively. In 4.4 the computational results are provided.

The results were obtained on a Windows 64-bit computer with Intel Xeon processor E5-1650, 3.50 GHz, and 16 GB RAM. The ALNS parameters have been tuned and fixed at $T_{max} = 10^7$, $\gamma = 0.001$ and $q = 2$. The tuning of parameters has been done by testing a small set of instances with increasing values of T_{max} , γ and q , following the scheme provided in Angelelli et al. (2022). We observed that greater values of T_{max} and smaller

values of γ considerably affect computational time without significant improvements in the objective function. We also observed that enlarging the destroy radius q reduces the possibility to find improving feasible solutions in the repair phase.

4.1 The benchmark instances

To generate the instances for the computational experiments, we used the benchmark instances of groups C and D introduced in Angelelli et al. (2022). The instances have 4 companies and 48 customers for Group C and 10 companies and 100 customers for Group D on a daily basis. Each group contains instances based on 4 big European metropolitan areas (Paris, Berlin, London and Rome). For each metropolitan area, 7 demand scenarios have been generated and grouped in a 7-day (week) instance for the multi-day context. Thus, we obtained, from Group C, 4 instances with 4 companies and 48 customers per day and, from Group D, 4 instances with 10 companies and 100 customers per day. All instances are available at <http://or-brescia.unibs.it/instances>.

Instances have been generated by randomly choosing origin and destination addresses in the city network area and choosing a random position for the companies' depot within the same area. Although these positions could affect the final results, in Angelelli et al. (2022), a thorough study of the impact of depot positioning has been conducted. It was found out that, when the number of customers is sufficiently large and customers origin and destination points are randomly located, as for the instances we used in this study, depots position does not impact on the effectiveness of the collaboration. In fact, depot position is just one of the many factors that influence the final results of horizontal collaboration optimization such as customers composition, the time windows distribution over customers, heterogeneity of companies in terms of number of customers.

4.2 Settings and collaboration schemes

We call *setting* the values we assign to parameters α^{DC} , α^{DT} , α^{WC} and α^{WT} . Values may be assigned according to some dependency rule in order to implement a particular operational logic, which we call *collaboration scheme* (or just *scheme* for short). In this paper, we consider the following collaboration schemes:

1. Independent daily collaboration (*Daily*). The collaboration is managed every day independently of each other. The same percentage is applied to bound daily customer and time balance. No bound is applied on weekly customer and time balance. The scheme implies

$$\alpha^{WC} = \alpha^{WT} = \infty.$$

We tested the following settings

α^{DC}	α^{DT}	α^{WC}	α^{WT}
0.1	0.1	∞	∞
0.2	0.2	∞	∞
0.4	0.4	∞	∞

The choice of values is meant to give enough margin to the model to achieve significant savings compared to the non-collaborative scheme on one side, and to maintain an acceptable level of fairness on the other side. Since a single value is enough to characterize this scheme, we may shortly represent the implemented settings as *Daily*(α) with $\alpha = \alpha^{DC} = \alpha^{DT}$. Observe that the percentage applied every day to bound the balance implies that the same bound will be respected also for the cumulated balance at the end of the week.

2. Weekly collaboration with daily bounds (*Weekly-DB*). The collaboration is managed every day like in *Daily*, but a percentage is applied to bound weekly the customer and time balance and the daily bound is relaxed by doubling this percentage. The scheme implies the following relationship on the four parameters

$$\alpha^{DC} = \alpha^{DT} = 2 \cdot \alpha^{WC} = 2 \cdot \alpha^{WT}.$$

We tested the following settings

α^{DC}	α^{DT}	α^{WC}	α^{WT}
0.2	0.2	0.1	0.1
0.4	0.4	0.2	0.2

The choice of values is meant to allow an easy and direct comparison to the *Daily* scheme. For example, *Weekly-DB*(0.1) applies the same bound as *Daily*(0.1) on the cumulated balance at the end of the week. Daily bounds in *Weekly-DB*(0.1) have been doubled with respect to *Daily*(0.1) in order to allow much more margin to the model on single days and to better exploit the benefits of the collaboration. Since a single value is enough to characterize this scheme, we may shortly represent the implemented settings as *Weekly-DB*(α) with $\alpha = \alpha^{WC} = \alpha^{WT}$ and $\alpha^{DC} = \alpha^{DT} = 2\alpha$.

3. Weekly collaboration (*Weekly*). Two distinct percentages α^{WC} and α^{WT} are applied to bound the overall customer and time balance at the end of the week. No bound is applied on the daily customer and time balance. The scheme implies

$$\alpha^{DC} = \alpha^{DT} = \infty.$$

We tested the following settings

α^{DC}	α^{DT}	α^{WC}	α^{WT}
∞	∞	0	0.05
∞	∞	0	∞
∞	∞	0.05	0.05
∞	∞	0.05	∞
∞	∞	0.1	0.05
∞	∞	0.1	∞

where a set of ‘small’ bounds on the weekly cumulated customer balance are tested versus on/off bound for the time ($\alpha^{WT} \in \{0, \infty\}$). Actually, as perfect null balance on times is very hard to satisfy, we used 0.05 in place of 0. This scheme is characterized by the two values of α^{WC} and α^{WT} . Thus, we may shortly denote the implemented settings as *Weekly*(α^{WC}, α^{WT}). The most interesting assignment for the parameters is *Weekly*(0.05, 0.05) meaning that we want the daily balance to be practically ruled out by the end of the week. We will see in the computational experiments that this collaboration scheme is very effective at almost reaching the objective of minimizing coalition costs and attaining the fairness among companies.

4. Unconstrained collaboration (*UC*). No bound is applied, neither on the daily balances nor on the weekly balance. This scheme provides a lower bound on overall cost for all other collaboration schemes. We have only one setting with

$$\alpha^{DC} = \alpha^{DT} = \alpha^{WC} = \alpha^{WT} = \infty.$$

Observe that, according to their description, *Daily*(2α) is a relaxation of *Weekly-DB*(α) which in turn is a relaxation of *Daily*(α).

4.3 Statistics

Comparing the collaborative models with the non-collaborative one, we consider the following measures.

- *Savings* : percentage cost savings obtained by collaborative setting with respect to the non-collaborative one;
- $maxB = \max_{m,f}(|B_m^f|)$: maximum absolute value of daily customer balance paid by any company on any day;
- $maxS = \max_{m,f}(|S_m^f|)$: maximum absolute value of daily time balance paid by any company on any day;
- $maxFinB = \max_m(|cumB_m^{F1}|)$: maximum absolute value of cumulated customer balance paid by any company on last day of the time horizon;
- $maxFinS = \max_m(|cumS_m^{F1}|)$: maximum absolute value of cumulated time balance paid by any company on last day of the time horizon.

We observe that statistics $maxB$, $maxS$, $maxFinB$ and $maxFinS$ are explicitly considered in constraints (21)–(24).

4.4 Computational results

For the sake of completeness, the 4 parameters α^{DC} , α^{DT} , α^{WC} and α^{WT} are reported in the tables that follow. Reported results are calculated on 4 instances (4 cities {*Berlin, London, Paris, Rome*}) for each of the two coalition sizes (i.e. *a4c48, a10c100*).

4.4.1 Average savings

In Table 4 we report the average *Savings* obtained by the tested settings with respect to the non-collaborative scheme. We observe that *Savings* obtained for the *Weekly* collaboration are comparable with those of the *UC* when $\alpha^{WC} \geq 5\%$. Only when $\alpha^{WC} = 0\%$ we have a strong reduction in performance, which cannot be recovered even by allowing an unlimited balance of time workload ($\alpha^{WT} = \infty$). However, the performance of these schemes remain comparable with that of other schemes in which a weekly bound on customer and time balance greater than 5% is allowed. As expected, the performances of *Weekly-DB* and *Daily* are, in terms of *Savings*, such that $Daily(0.1) < Weekly-DB(0.1) < Daily(0.2) < Weekly-DB(0.2) < Daily(0.4)$.

4.4.2 Balances

In Tables 5 and 6 we analyze how imbalanced the most imbalanced company during the week is. The maximum of daily percentage balance realized by companies in the

Table 4 *Savings* averaged on 4 instances for two coalition sizes

Collaboration	α^{DC}	α^{DT}	α^{WC}	α^{WT}	a4c48 (%)	a10c100 (%)
<i>UC</i>	∞	∞	∞	∞	24.11	29.17
<i>Weekly</i>	∞	∞	0	0.05	15.33	11.28
<i>Weekly</i>	∞	∞	0	∞	17.00	12.52
<i>Weekly</i>	∞	∞	0.05	0.05	22.26	26.97
<i>Weekly</i>	∞	∞	0.05	∞	22.26	29.00
<i>Weekly</i>	∞	∞	0.1	0.05	23.71	27.71
<i>Weekly</i>	∞	∞	0.1	∞	24.08	29.08
<i>Weekly-DB</i>	0.2	0.2	0.1	0.1	17.76	15.16
<i>Weekly-DB</i>	0.4	0.4	0.2	0.2	20.24	21.17
<i>Daily</i>	0.1	0.1	∞	∞	15.97	10.05
<i>Daily</i>	0.2	0.2	∞	∞	18.31	15.18
<i>Daily</i>	0.4	0.4	∞	∞	21.79	22.79

Table 5 Average and Maximum of maximum customer daily exchange (*maxB*) on 4 instances for 2 coalition sizes

Collaboration	α^{DC}	α^{DT}	α^{WC}	α^{WT}	a4c48		a10c100	
					Avg. <i>maxB</i> (%)	Max. <i>maxB</i> (%)	Avg. <i>maxB</i> (%)	Max. <i>maxB</i> (%)
<i>UC</i>	∞	∞	∞	∞	41.67	50.00	50.00	60.00
<i>Weekly</i>	∞	∞	0	0.05	33.33	41.67	20.00	20.00
<i>Weekly</i>	∞	∞	0	∞	41.67	50.00	22.50	30.00
<i>Weekly</i>	∞	∞	0.05	0.05	43.75	50.00	57.50	70.00
<i>Weekly</i>	∞	∞	0.05	∞	43.75	50.00	50.00	60.00
<i>Weekly</i>	∞	∞	0.1	0.05	43.75	50.00	57.50	70.00
<i>Weekly</i>	∞	∞	0.1	∞	43.75	50.00	50.00	60.00
<i>Weekly-DB</i>	0.2	0.2	0.1	0.1	16.67	16.67	20.00	20.00
<i>Weekly-DB</i>	0.4	0.4	0.2	0.2	33.33	33.33	40.00	40.00
<i>Daily</i>	0.1	0.1	∞	∞	8.33	8.33	10.00	10.00
<i>Daily</i>	0.2	0.2	∞	∞	16.67	16.67	20.00	20.00
<i>Daily</i>	0.4	0.4	∞	∞	33.33	33.33	40.00	40.00

Table 6 Average and Maximum of maximum time daily exchange (*maxS*) on 4 instances for 2 coalition sizes

Collaboration	α^{DC}	α^{DT}	α^{WC}	α^{WT}	a4c48		a10c100	
					Avg. <i>maxS</i> (%)	Max. <i>maxS</i> (%)	Avg. <i>maxS</i> (%)	Max. <i>maxS</i> (%)
<i>UC</i>	∞	∞	∞	∞	42.91	53.18	69.38	81.56
<i>Weekly</i>	∞	∞	0	0.05	39.40	47.30	28.43	33.10
<i>Weekly</i>	∞	∞	0	∞	42.26	47.30	35.19	55.75
<i>Weekly</i>	∞	∞	0.05	0.05	47.69	62.34	58.73	69.02
<i>Weekly</i>	∞	∞	0.05	∞	47.69	62.34	71.85	83.52
<i>Weekly</i>	∞	∞	0.1	0.05	46.22	62.34	67.57	75.96
<i>Weekly</i>	∞	∞	0.1	∞	43.93	53.18	69.61	81.56
<i>Weekly-DB</i>	0.2	0.2	0.1	0.1	18.81	19.96	19.69	19.91
<i>Weekly-DB</i>	0.4	0.4	0.2	0.2	34.94	37.09	38.87	39.79
<i>Daily</i>	0.1	0.1	∞	∞	9.56	9.71	9.78	9.96
<i>Daily</i>	0.2	0.2	∞	∞	18.19	19.83	19.54	19.87
<i>Daily</i>	0.4	0.4	∞	∞	36.33	39.08	39.64	40.00

tested schemes is reported for customers (*maxB*) and times (*maxS*), respectively. In particular, for each instance we consider the maximum percentage balance realized by the companies during the week (maximum out of 7×4 values for coalition size *a4c48*, and out of 7×10 values for coalition size *a10c100*). We, thus, obtain 4 values for each coalition size (one for each city). We report the averages in columns *Avg. maxB* (customers) and *Avg. maxS*(times), and the maximums in columns *Max. maxB* and *Max. maxS* for each coalition size. In columns *Avg. maxB* and *Avg. maxS* we can see that for all *Weekly-DB* and *Daily* schemes values are as close as possible to the daily bounds α^{DC} and α^{DT}

Table 7 Average and Maximum customer cumulative exchange balance (*maxFinB*) on day 7

Collaboration	α^{DC}	α^{DT}	α^{WC}	α^{WT}	a4c48		a10c100	
					Avg. <i>maxFinB</i> (%)	Max. <i>maxFinB</i> (%)	Avg. <i>maxFinB</i> (%)	Max. <i>maxFinB</i> (%)
<i>UC</i>	∞	∞	∞	∞	8.33	11.90	9.29	18.57
<i>Weekly</i>	∞	∞	0	0.05	0.00	0.00	0.00	0.00
<i>Weekly</i>	∞	∞	0	∞	0.00	0.00	0.00	0.00
<i>Weekly</i>	∞	∞	0.05	0.05	3.27	4.76	4.29	4.29
<i>Weekly</i>	∞	∞	0.05	∞	3.27	4.76	4.29	4.29
<i>Weekly</i>	∞	∞	0.1	0.05	5.65	9.52	8.57	10.00
<i>Weekly</i>	∞	∞	0.1	∞	6.85	9.52	7.14	10.00
<i>Weekly-DB</i>	0.2	0.2	0.1	0.1	6.25	8.33	7.50	10.00
<i>Weekly-DB</i>	0.4	0.4	0.2	0.2	5.95	11.90	15.36	20.00
<i>Daily</i>	0.1	0.1	∞	∞	4.76	5.95	5.36	7.14
<i>Daily</i>	0.2	0.2	∞	∞	6.85	9.52	10.36	14.29
<i>Daily</i>	0.4	0.4	∞	∞	7.14	10.71	16.43	20.00

Table 8 Average and Maximum time cumulative exchange balance (*maxFinS*) on day 7

Collaboration	α^{DC}	α^{DT}	α^{WC}	α^{WT}	a4c48		a10c100	
					Avg. <i>maxFinS</i> (%)	Max. <i>maxFinS</i> (%)	Avg. <i>maxFinS</i> (%)	Max. <i>maxFinS</i> (%)
<i>UC</i>	∞	∞	∞	∞	6.42	12.21	12.48	20.39
<i>Weekly</i>	∞	∞	0	0.05	2.86	4.83	2.83	4.90
<i>Weekly</i>	∞	∞	0	∞	3.42	4.06	4.54	5.46
<i>Weekly</i>	∞	∞	0.05	0.05	3.52	4.85	4.63	4.85
<i>Weekly</i>	∞	∞	0.05	∞	3.52	4.85	10.34	14.12
<i>Weekly</i>	∞	∞	0.1	0.05	3.53	4.97	4.66	4.94
<i>Weekly</i>	∞	∞	0.1	∞	5.96	12.21	10.86	13.92
<i>Weekly-DB</i>	0.2	0.2	0.1	0.1	6.13	8.87	7.30	9.95
<i>Weekly-DB</i>	0.4	0.4	0.2	0.2	8.46	16.14	14.68	16.81
<i>Daily</i>	0.1	0.1	∞	∞	4.04	5.02	4.01	6.37
<i>Daily</i>	0.2	0.2	∞	∞	5.07	7.21	10.57	12.52
<i>Daily</i>	0.4	0.4	∞	∞	9.62	15.80	15.22	18.63

(observe that in coalition size *a4c48* every single customer weighs 8.33% in the balance), proving that on all instances the bound is tight. A fortiori, also columns *Max. maxB* and *Max. maxS*, where the maximum is reported in place of the average, are close to the bound. Indeed, the *UC* scheme, which does not apply any bound on the balances, reports for maximum values always above 50%. Thus, the daily bound must become active in *Weekly-DB* and *Daily* schemes, where the applied daily bound is always below 40%. However, in the *Weekly* scheme, we observe that both maximum and average values are higher than those produced by the *UC* scheme when $\alpha^{WC} > 0$. This phenomenon can be

explained by the fact that, having to close the accounts at the end of the week, some balances are even more stretched during the week. When $\alpha^{WC} = 0$, the exchanges remain a little smaller, but as we have seen, at the expense of cost performance.

In Tables 7 and 8 we analyze how imbalanced the most imbalanced company at the end of the week is, taking into account all the exchanges incurred in that period. The maximum of the cumulated percentage balance, realized on the last day of the week by companies in the tested schemes, is reported for customers (*maxFinB*) and times (*maxFinS*), respectively. In particular, for each company, we consider the maximum percentage of the cumulated balance realized at the end of the week with respect to the workload of the whole week, for each coalition size (e.g. *a4c48* and *a10c100*). We, thus, obtain 4 values for each coalition size (one for each city). We report the averages in columns *Avg. maxFinB* (customers) and *Avg. maxFinS*(times), and the maximums in columns *Max. maxFinB* and *Max. maxFinS* for each coalition size.

Furthermore, we immediately observe that, in general, these values are much smaller than the corresponding maximum values reported in Tables 5 and 6. This indicates that during the week each company realizes both positive and negative balances which, to some extent, compensate each other in the long run.

From the obtained results, we can also observe some significant difference in the values produced by the tested schemes. In particular, as expected, the *UC* scheme produces maximum values that are similar between customers and times and significantly high: around 12% and 20% for coalition sizes *a4c48* and *a10c100*, respectively. This suggests that, in the case of unconstrained collaboration, some company may be systematically overloaded (or devoid) and a certain imbalance is generated in exchanges from or towards one or more companies.

In terms of maximum and average cumulated exchange balance, the schemes *Weekly-DB* and *Daily* appear to be not well-performing with respect to the *UC* as for $\alpha^{WC} \geq 0.2$ they produce a similar average and maximum values of cumulated balance for both customers and times while they do not reach the savings of *UC* scheme. When $\alpha^{WC} = 0.1$ the exchanges improve a lot, but the *Savings* definitely deteriorate.

The scheme *Weekly* seems to perform better versus the *UC* scheme producing values almost always largely smaller; on the other side the *Savings* are comparable.

In Tables 9 and 10 we investigate how imbalanced the companies during the week are, both on single days and with respect to the cumulated balance up to any intermediate day of the week. In the previous tables, we investigated the maximum level of dissatisfaction; here we address the global satisfaction. In particular, for each coalition size, we consider the daily and cumulated balance on each day of the week for each company. As we have 4 instances, we end up with a sample of $4 \times 7 \times 10 = 280$ daily (and cumulated) balance values for the coalitions of size *a10c100* and with $4 \times 7 \times 4 = 112$ daily (and cumulated) balance values for the coalitions of size *a4c48*. We average the samples both for customer and time balance.

Given that every situation has its own peculiarities, many of the proposed schemes ensure a good trade-off in terms of *Savings* and balance. On the tested

Table 9 Average over daily (B) and cumulative ($cumB$) customer exchange balance

Collaboration	α^{DC}	α^{DT}	α^{WC}	α^{WT}	a4c48		a10c100	
					Avg. B (%)	Avg. $cumB$ (%)	Avg. B (%)	Avg. $cumB$ (%)
<i>UC</i>	∞	∞	∞	∞	13.10	5.92	17.21	9.01
<i>Weekly</i>	∞	∞	0	0.05	9.52	3.32	4.64	1.87
<i>Weekly</i>	∞	∞	0	∞	10.42	3.65	5.57	2.33
<i>Weekly</i>	∞	∞	0.05	0.05	14.58	6.42	17.36	7.22
<i>Weekly</i>	∞	∞	0.05	∞	14.58	6.42	15.93	7.51
<i>Weekly</i>	∞	∞	0.1	0.05	13.84	5.22	17.64	8.54
<i>Weekly</i>	∞	∞	0.1	∞	13.54	5.14	16.00	8.45
<i>Weekly-DB</i>	0.2	0.2	0.1	0.1	9.82	5.80	11.43	6.01
<i>Weekly-DB</i>	0.4	0.4	0.2	0.2	13.24	6.50	16.57	9.71
<i>Daily</i>	0.1	0.1	∞	∞	6.25	4.03	6.07	3.71
<i>Daily</i>	0.2	0.2	∞	∞	8.93	4.59	11.36	5.76
<i>Daily</i>	0.4	0.4	∞	∞	13.39	5.76	17.43	10.59

Table 10 Average over daily (S) and cumulative ($cumS$) time exchange balance

Collaboration	α^{DC}	α^{DT}	α^{WC}	α^{WT}	a4c48		a10c100	
					Avg. S (%)	Avg. $cumS$	Avg. S (%)	Avg. $cumS$ (%)
<i>UC</i>	∞	∞	∞	∞	15.28	7.38	20.59	11.44
<i>Weekly</i>	∞	∞	0	0.05	10.59	4.74	5.00	2.79
<i>Weekly</i>	∞	∞	0	∞	11.97	4.30	6.51	3.43
<i>Weekly</i>	∞	∞	0.05	0.05	16.60	8.04	17.73	8.25
<i>Weekly</i>	∞	∞	0.05	∞	16.60	8.04	19.25	9.23
<i>Weekly</i>	∞	∞	0.1	0.05	15.49	6.69	19.99	8.78
<i>Weekly</i>	∞	∞	0.1	∞	15.06	6.68	20.33	10.75
<i>Weekly-DB</i>	0.2	0.2	0.1	0.1	9.60	6.31	10.06	5.05
<i>Weekly-DB</i>	0.4	0.4	0.2	0.2	14.61	7.66	17.67	11.67
<i>Daily</i>	0.1	0.1	∞	∞	4.62	3.52	4.48	2.46
<i>Daily</i>	0.2	0.2	∞	∞	8.97	4.58	10.82	5.84
<i>Daily</i>	0.4	0.4	∞	∞	15.24	8.18	18.17	10.31

instances, the *Weekly* scheme performs similarly to the *UC* one and controls very well the balance over the time horizon. In fact, if no daily bounds are required, the *Weekly* produces the best trade-off results in terms of fairness and efficiency.

4.4.3 Average savings versus balances

We depict, in Figs. 6 and 7, some results already reported numerically in Tables 4-8. In Fig. 6 we consider the average values obtained by a sample of collaboration

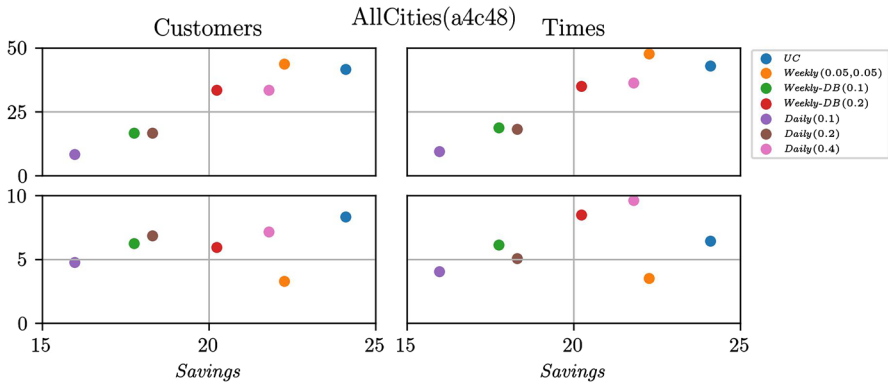


Fig. 6 Average *Savings* versus *maxB* (up-left), *maxS* (up-right), *maxFinB* (bottom-left) and *maxFinS* (bottom-right)

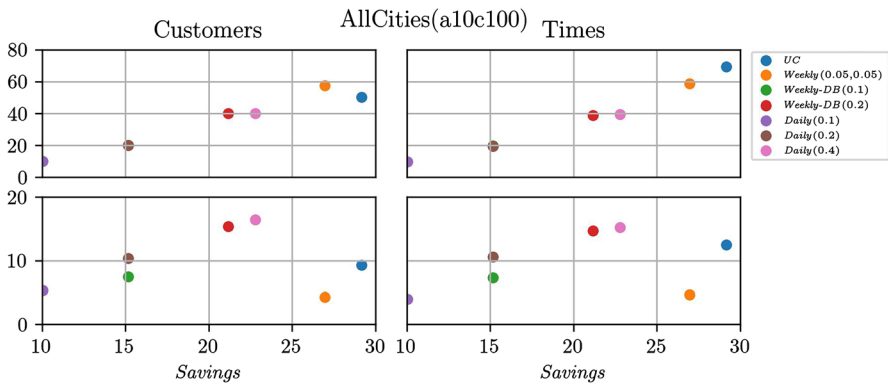


Fig. 7 Average *Savings* versus *maxB* (up-left), *maxS* (up-right), *maxFinB* (bottom-left) and *maxFinS* (bottom-right)

schemes on all cities for the coalition size *a4c48*. In the top row, we plot the average *Savings* versus the average of *maxB* and *maxS* (Customers and Times) reporting column *Savings* of Table 4 vs. columns *Avg.maxB* and *Avg.maxS* of Tables 5 and 6, respectively. In the bottom row, we plot the average *Savings* versus the average of *maxFinB* and *maxFinS* (Customers and Times) reporting column *Savings* of Table 4 vs. columns *Avg.maxFinB* and *Avg.maxFinS* of Tables 7 and 8, respectively. Analogous results are reported in Fig. 7 for coalition size *a10c100*.

We observe that, while scheme *Weekly(0.05,0.05)* attains the *Savings* closest to *UC* scheme, in general it provides the worst (highest) values of *maxB* and *maxS*, but definitely excellent values of *maxFinB* and *maxFinS*. Thus, it is an excellent candidate for an actual implementation of collaboration.

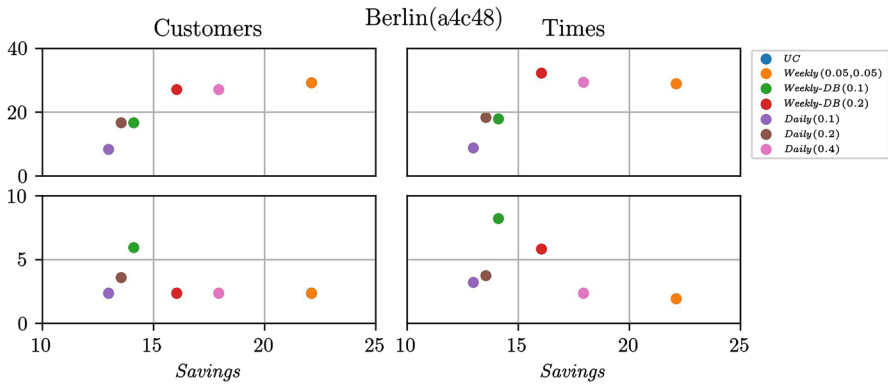


Fig. 8 Average *Savings* versus $maxB$ (up-left), $maxS$ (up-right), $maxFinB$ (bottom-left) and $maxFinS$ (bottom-right)

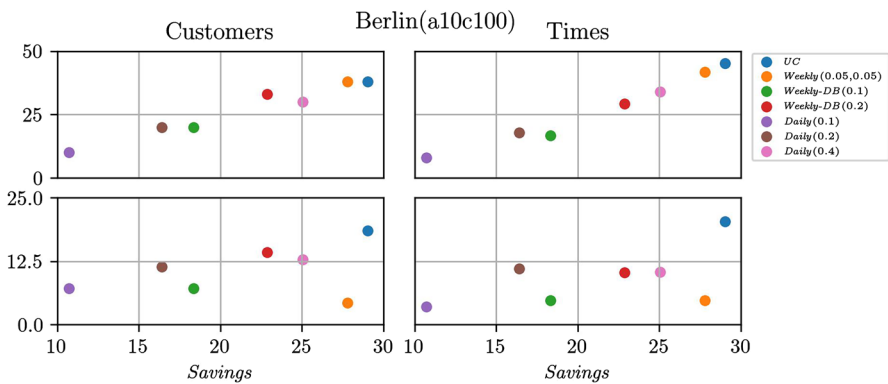


Fig. 9 Average *Savings* versus $maxB$ (up-left), $maxS$ (up-right), $maxFinB$ (bottom-left) and $maxFinS$ (bottom-right)

The same pattern is followed in Figs. 8 and 9, where the case of a single instance (*Berlin*) is presented for coalition size *a4c48* and coalition size *a10c100*.

All Figs. 6, 7, 8 and 9 should be read in terms of a multi-objective problem where we want the points on the graphs to be as far to the right, as the more right the point is the higher the saving is, and as low as possible, as the lower the point is the lower the balance is.

With respect to the cumulated balances at the end of the week (bottom rows of the figures) we observe that the point representing *Weekly*(0.05,0.05) is efficient with respect to the point representing the *UC* scheme: a slightly smaller savings, but a smaller unfairness as well. Moreover, *Weekly* (0.05,0.05) produces the second best saving after *UC* while having the least cumulated balances at the end of the week. The same can not be said of *UC*, which is more unfair than several other collaboration schemes.

On the contrary, with respect to the maximum daily balance over the week, the *Weekly* scheme seems to perform often worse than other schemes. However, collaboration schemes able to perform much better the *Weekly* in this regard, have a very poor performance in terms of savings.

Thus, from a managerial viewpoint, the collaboration initiative can be exploited to obtain great savings with a very small level of cumulated unfairness at the end of the period, or to obtain smaller savings with a day-by-day high level of unfairness.

4.4.4 Day-by-day balances

In Fig. 10 we provide detailed results obtained by the *Weekly*(0.05, 0.05) scheme, on each day of the horizon $F = \{1, 2, \dots, 7\}$ for each company on instance *Berlin* in coalition size *a4c48*.

In particular, we show the percentage cumulated customer balance (top row) and percentage cumulated time balance (bottom row) as determined by the collaboration scheme. The bar charts report day-by-day the percentage balance, while in the line charts we see the cumulated percentage balance. We can observe that all companies are involved in the exchange almost every day and with alternating signs. As expected, exchanges present a peak (on day 5), but are quite limited for the rest of the week. Not surprisingly, the cumulated percentage balance tends to be smoother as time goes on in the week. At the end of the week, *Weekly*(0.05, 0.05) manages to keep the most imbalanced company at the end of the week quite under control, as required by the model. In Appendix B we report the same kind of figures concerning all other schemes we reported in Fig. 6 (see Figs. 11, 12, 13, 14, 15 and 16).

5 Conclusions

In this paper, we have presented an optimization model that controls the level of fairness in the assignment of customers to shared taxi companies engaged in a collaboration initiative. The model assigns the customers with the objective of minimizing the

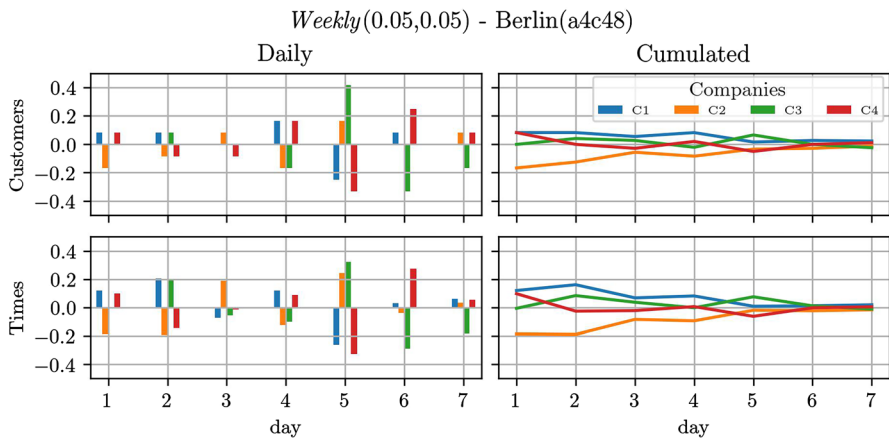


Fig. 10 Daily and cumulated customer and time balance for the given scheme

total routing cost while bounding the number of customers and the traveled time of each company over a pre-defined time horizon and/or in a single day. Different approaches to control the fairness are compared on a set of benchmark instances. The computational results show the flexibility of the multi-day approach and its potential to further reduce the operational costs compared with the single-day optimization and, at the same time, to be fair to all companies, and, thus, to make the collaboration sustainable.

Further research should be devoted to the application of the approach to companies engaged in different transportation services and to experiments on real data. Other approaches may be designed that have the same goal, that is to guarantee to each company involved in a collaboration that it will always gain from the collaboration (daily certainly or over a period of time). Only with this guarantee the collaboration initiative can be sustainable and exploit its potential to reduce the environmental impact of transportation services.

Appendix A. Example

In the following, we list the details of route plans obtained for each scheme in the introductory example provided in Sect. 1.

In Fig. 1, the route plans of the two companies for the two days in a NC scheme are shown. The red company is servicing its own customers with a daily cost of 19 while the blue company has a daily cost of 25. The total routing cost is 88. In this case, no exchange among companies has been performed and, hence, the assignment is fair as each company is serving its own customers.

As a lower bound, in Fig. 4, the *UC* case, in which no bounds on daily and cumulated customer balance are imposed, is depicted. We observe that the total cost is 68 with a 22.7% saving. As shown in Table 1, savings for companies could be very different. In fact, along a huge saving for the blue company (72%), a strong increase (around 42%) in costs is observed for the red company. In terms of customer balance (both daily and cumulated), the proposed route plans are poor performing as blue company serves 2 customers less than in the NC scheme every day and 4 less at the end of the two days and, hence, it scores 40% in both daily and cumulated customer balance.

In Fig. 2, we show the solution when we impose that the daily customer balance, in absolute value, has to be less than 20%, meaning that both companies could serve daily up to 6 customers and not less than 4 customers. As a result, the blue company is conceding its customer 5 to the red company in both days. At the end of the time horizon, there is a high cumulated customer balance (20%) as the blue company conceded two customers while the red one acquired the two. The total routing cost is 84 and, hence, a saving of 4.5% is obtained. As shown in Table 1, individual savings for the blue company is around 16% with a moderate increase in costs, around 10%, for the red company.

Finally, in Fig. 3, we depict the route plans produced by bounding the final cumulated customer balance, in absolute value, to no more than 20%. In this case, the total route length drops to 70 with a 20.45% saving. This value is very close to the one obtained in the *UC* case, making the price of being fair quite acceptable, i.e. losing around 2% of the saving. Even though the maximum daily customer balance resembles

the *UC* one, the company being at a loss in the first day becomes the one being in profit in the second day, making the average daily customer balance equal to zero. In fact, the imbalance in terms of served customers at the end of the two days is zero as the number of customers served by each company is the same they had in the *NC* scheme. According to data in Table 1, both companies are having an advantage in terms of costs.

Appendix B. Figures

See Figs. 11, 12, 13, 14, 15 and 16.

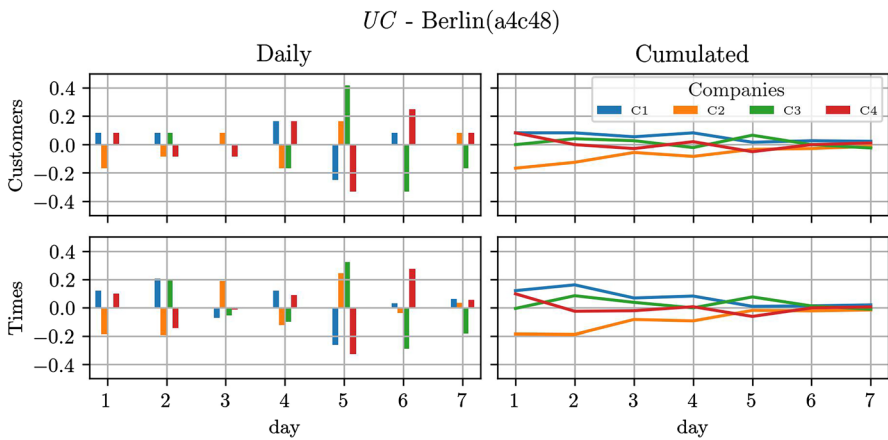


Fig. 11 Daily and cumulated customer and time balance for the given scheme

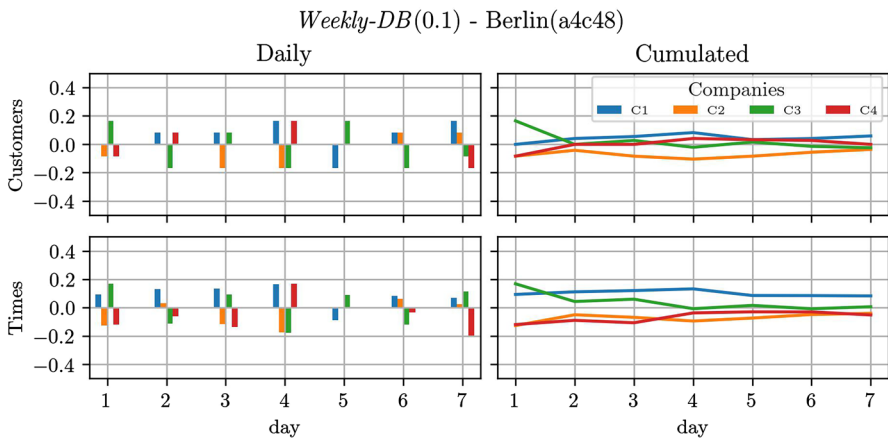


Fig. 12 Daily and cumulated customer and time balance for the given scheme

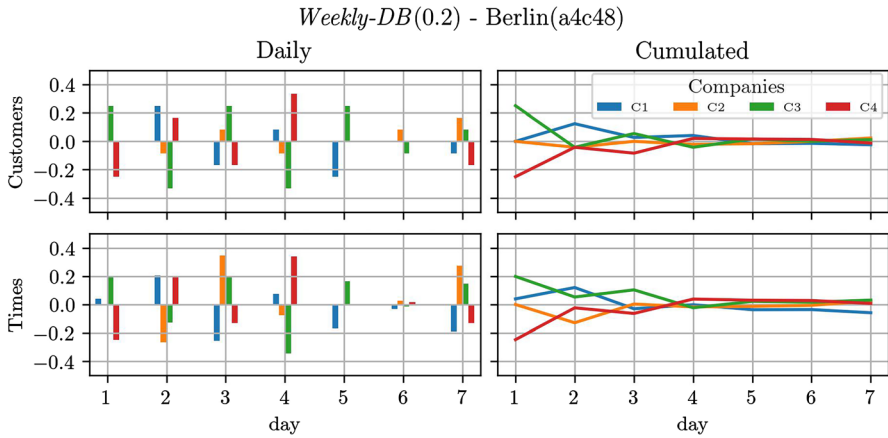


Fig. 13 Daily and cumulated customer and time balance for the given scheme

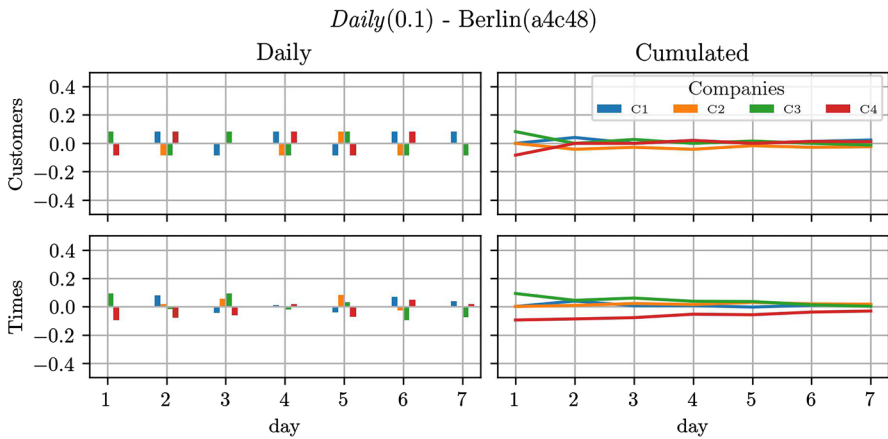


Fig. 14 Daily and cumulated customer and time balance for the given scheme

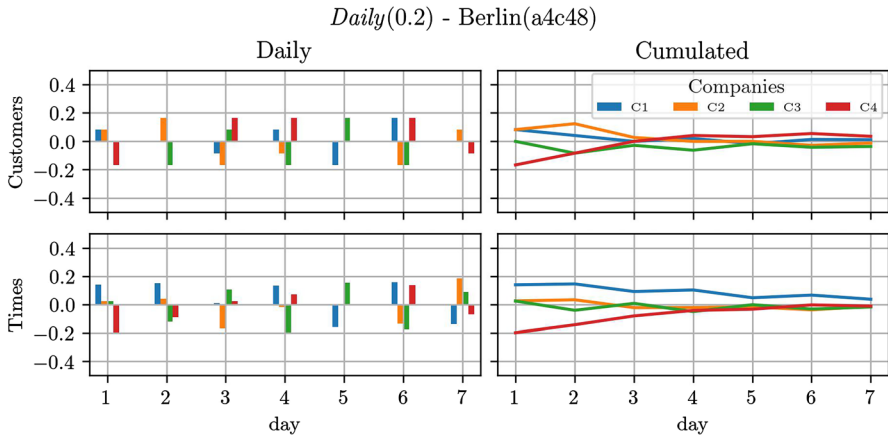


Fig. 15 Daily and cumulated customer and time balance for the given scheme

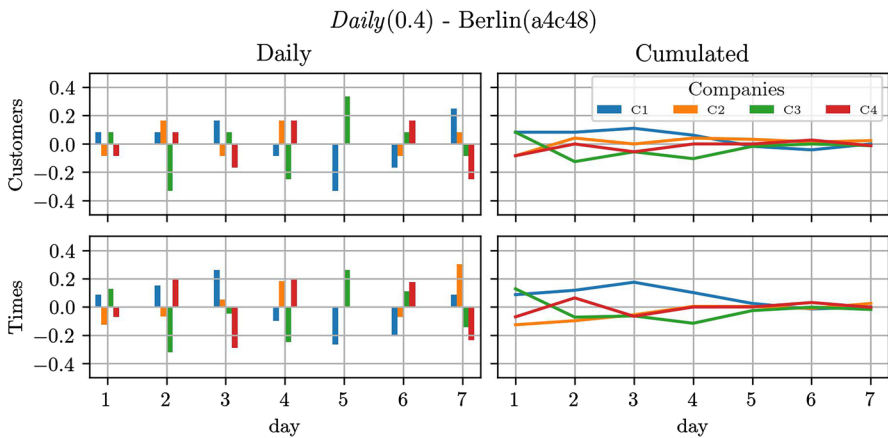


Fig. 16 Daily and cumulated customer and time balance for the given scheme

Acknowledgements The authors express their appreciation for the comments and suggestions of the anonymous reviewers which allowed them to substantially improve a previous version of this paper.

Funding Open access funding provided by Università degli Studi di Brescia within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose and have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative

Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aloui A, Hamani N, Derrouiche R, Delahoche L (2021) Systematic literature review on collaborative sustainable transportation: overview, analysis and perspectives. *Transport Res Interdiscip Perspect* 9:100291
- Angeles E, Morandi V, Speranza MG (2022) Optimization models for fair horizontal collaboration in demand-responsive transportation. *Transport Res Part C Emerg Technol* 140:103725
- Basso F, D'Amours S, Rönnqvist M, Weintraub A (2019) A survey on obstacles and difficulties of practical implementation of horizontal collaboration in logistics. *Int Trans Oper Res* 26:775–793
- Cordeau J-F (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Oper Res* 54:573–586
- Crujssen F, Cools M, Dullaert W (2007) Horizontal cooperation in logistics: opportunities and impediments. *Transport Res Part E Logist Transport Rev* 43:129–142
- Fernández E, Fontana D, Speranza MG (2016) On the collaboration uncapacitated arc routing problem. *Comput Oper Res* 67:120–131
- Gansterer M, Hartl RF (2018) Collaborative vehicle routing: a survey. *Eur J Oper Res* 268:1–12
- Gansterer M, Hartl R (2021) The prisoners' dilemma in collaborative carriers' request selection. *Cent Eur J Oper Res* 29:1–15
- Gansterer M, Hartl RF, Wieser S (2020) Assignment constraints in shared transportation services. *Ann Oper Res* 305:1–27
- Mladenović N, Hansen P (1997) Variable neighborhood search. *Comput Oper Res* 24:1097–1100
- Molenbruch Y, Braekers K, Caris A (2017) Benefits of horizontal cooperation in dial-a-ride services. *Transport Res Part E Logist Transport Rev* 107:97–119
- Mourad A, Puchinger J, Chu C (2019) A survey of models and algorithms for optimizing shared mobility. *Transport Res Part B Methodol* 123:323–346
- Muller LF (2009) An adaptive large neighborhood search algorithm for the resource-constrained project scheduling problem. In: *Proceedings of the VIII metaheuristics international conference (MIC)*
- Nelson JD, Wright S (2021) Flexible transport services. In: *The Routledge handbook of public transport*. Routledge, pp 224–235
- Pisinger D, Ropke S (2010) Large neighborhood search. In: *Handbook of metaheuristics*. Springer, pp 399–419
- Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp Sci* 40:455–472
- Vansteenwegen P, Melis L, Aktaş D, Montenegro BDG, Vieira FS, Sörensen K (2022) A survey on demand-responsive public bus systems. *Transport Res Part C Emerg Technol* 137:103573

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.