A Benders decomposition approach for planning home blood donations

Tiziano Bacci[1] · Ettore Lanzarone[2] · Sara Mattia[1] · Paolo Ventura[1]

Abstract

Blood donations are crucial for the health system. We consider the problem of planning blood donation services, where the donors are reached at home. The scope is to minimize the penalty for the unserved donors, while guaranteeing that the available resources for implementing the service are not exceeded and that the appointment preferences of the donors are met. We present an offline model for this setting, where the produced solution must be robust with respect to the availability of the donors, which is not known in advance and is managed in a stochastic way using scenarios. A Benders decomposition approach to solve this model is developed. The proposed algorithm is tested on real-life instances coming from the Milan department of the Associazione Volontari Italiani Sangue (AVIS).

Keywords Home blood donations · Donors preferences · Benders decomposition · Data uncertainty · Scenarios

1 Introduction

The availability of a sufficient blood supply is crucial for ensuring high-quality and fully-functional health services. Unfortunately, blood and its components are highly perishable: this forbids long term conservation and requires continuous donations by unpaid volunteers, as no successful artificial blood substitutes have been developed so far. Therefore, the policies used to implement and manage the so-called Blood Donation Supply Chain (BDSC) deeply affect the whole healthcare system. BDSC includes four main stages: collection, transportation, storage and final utilization

Extended author information available on the last page of the article

⛹ Springer

(Pierskalla 2004). Collection consists of finding donors, checking them for eligibility, collecting donations and controlling the produced blood units (Lanzarone and Yalçındağ 2019). Transportation faces the problem of moving blood from where donations are made to the storage place. Storage deals with the blood conservation, considering that the shelf life of blood products is very limited. Final utilization consists of distributing blood units to end-users, but also of predicting the amounts needed from the different health facilities.

Collection is the most critical stage of the whole process and, relying on voluntary donors, it structurally suffers from data uncertainty (Rahmani 2019). Moreover, it is particularly vulnerable to pandemics. In fact, it was calculated that, due to the SARS-CoV-2 pandemic, the production of red blood cells in Italy in April 2020 was reduced by 36.4% with respect to April 2019 (Centro Nazionale Sangue 2020). There were several concurrent reasons that caused this contraction: many donors became unavailable because they were infected; most places used as locations to perform blood donations (schools, churches, offices, clubs, …) were shut down because of the pandemic; many donors were scared of reaching hospitals or donation centers, where they could potentially meet SARS-CoV-2 positive individuals (Gupta et al. 2020; Pagano et al. 2020). This supports the policy of relocating health services on the territory instead of concentrating them into hospitals or centers (Lanzarone and Matta 2012; Carello et al. 2018; Govindan et al. 2020; Regis-Hernández et al. 2020), which includes home blood donations (Doneda et al. 2023). The possibility to donate at home can be seen as an incentive: if donating becomes easier for the donors, more donors can be convinced to start (or re-start) donating and to become regular donors

For this reason, this paper is devoted to the presentation of models and algorithms for a home blood donation problem that arises in an urban context. We show that the problem exhibits an underlying network structure that can be exploited to efficiently solve it by a branch-and-cut algorithm based on Benders decomposition. The approach is tested on instances derived from real-life data. In the rest of this section, after a brief review of the existing literature on the blood supply chain and on blood donations, we describe our contribution and provide information and references on data and techniques we use.

1.1 Literature review

While the medical aspects of blood donation are very well-studied, the logistics of the blood supply chain has received less attention in the literature, if compared to other supply chains (Baş Güre et al. 2016, 2018b). Recent surveys can be found in (Beliën and Forcé 2012; Osorio et al. 2015; Pirabán et al. 2019). Starting from Millard (1959), where industrial inventory models were applied to blood collection, several papers considered the blood supply chain, also including vehicles and mobile donations. Gunpinar and Centeno (2016) use vehicles to create temporary blood collection sites in regions that may be far from a collection center. The problem of managing a fleet of vehicles to transport blood from temporary locations to

the centers has been considered by Alfonso et al. (2015a), Gülizahinyazan et al. (2015), and Şahinyazan et al. (2015).

Scheduling problems arising in blood donations have also been investigated. Michaels et al. (1993) use simulation to compare donors scheduling strategies. Alfonso et al. (2012) analyze how different donor schedules and workforce planning strategies affect the efficiency of the system. A simulation-optimization approach for capacity planning and appointment scheduling is considered by Alfonso et al. (2015b). Baş Güre et al. (2018a) present a framework for scheduling appointments over a few weeks, considering both booked and non-booked donors, with the aim of balancing the production and reduce the queuing time. The approach was extended to stochastic arrivals by Yalçındağ et al. (2020). Finally, the problem of dimensioning a collection center based on the expected donors arrival patterns is studied by Testik et al. (2010).

However, the assumption that has been made so far in the blood collection literature is that the donors must come to the place of donation (either temporary or permanent) and not the other way around. Although the problem of collecting blood donations at home shares some aspects with the collection of biological samples from patients, which is a common practice in home care (Anaya-Arenas et al. 2021), home donations have been considered for the first time in the literature only recently (Doneda et al. 2023). The authors develop an appointment scheduling system for home donation, with a planning horizon of several weeks, taking into account donors preferences and balanced production. The problem is decomposed into three stages, that are sequentially solved: an offline procedure for selecting the appointments to be opened; an online phase for booking the appointments; an offline routing planner, where the routes are arranged. The online phase is required to obtain information on donors preferences, for which reliable statistical predictions are not available.

1.2 Our contribution

In this paper we consider a short time horizon (e.g., one week) and some statistical knowledge on donors availability and preferences can be exploited. This allows us to model the uncertainty using a set of scenarios that represents the availability of donors over the time horizon, thus deriving a fully offline mathematical model to handle home blood donation in an urban context. Using scenarios in order to manage data uncertainty, is a common approach, that has been already used in many contexts, including healthcare (Lanzarone et al. 2012; Guo et al. 2020). The proposed formulation represents the complex situation that a donation center faces, which includes: the assignment of appointments to donors; the presence of limited resources; the need to ensure some amounts of each blood type; the preferences of the donors and the uncertainty in the availability of the donors. To solve the proposed model, we develop an ad-hoc branch-and-cut (Padberg and Rinaldi 1991) algorithm, based on a Benders decomposition approach (Benders 1962). In particular, we show that an underlying network structure can be exploited to efficiently find violated Benders inequalities, if any.

Benders decomposition has been already proved to be effective in many fields, such as scheduling problems (Mattia et al. 2017), prevention of blackouts in power grids (Bienstock and Mattia 2007) and network design (Avella et al. 2007; Mattia 2012a, b, 2013; Mattia and Poss 2018). There also exist generalizations of this method (Hooker 2019), with application in operating room scheduling (Guo et al. 2020). Discovering and exploiting underlying structures in the solution of mixed integer programming problems, is also a well-known and successful technique. One possibility is to identify submatrices of the original constraint matrix that exhibit a combinatorial structure, that can be used to generate valid inequalities to strengthen the problem formulation. Inequalities from combinatorial structures include, among the others, *clique*, *(lifted) cover*, *network flows-based* (Achterberg and Raack 2010; Avella et al. 2013, 2021) and *mod-2 inequalities* (Avella et al. 2007; Arbib et al. 2017). Another possibility is to show that a given problem can be transformed into another one, for which efficient algorithms and theoretical results are available. Many applied problems have an underlying network structure that can be exploited, e.g., personnel scheduling (Mattia et al. 2017), pick-up and delivery (Bonomo et al. 2011), wireless network planning (Mannino et al. 2011), assignment and matching (Ahuja et al. 1993). Here we show that we can obtain Benders feasibility and optimality cuts by solving a min cost flow problem on a suitable graph. To our knowledge, this is the first time that this strategy has been used to tackle the home blood donation problem.

Data for the experiments come of the Milan department of the Associazione Volontari Italiani Sangue (AVIS), referred to as AVIS Milan. AVIS is the largest blood donors association in Italy, being responsible for the collection stage of a very large portion of the Italian blood supply chain (about 70% of the national blood demand). AVIS Milan manages about 50 whole blood donations per day and other few units produced by apheresis (AVIS 2020) and serves one of the largest hospitals in Milan, the Niguarda Hospital. This hospital must be replenished with a given amount of units for each blood type, which allows to set target levels for each planning period. It is worth remarking that this situation is common to several blood collection centers, thus making AVIS Milan and its setting of general validity in blood collection.

The rest of the paper is organized as follows. In Sect. 2 we describe the considered problem and present a first mathematical model. In Sect. 3 we provide theoretical results that allow to obtain a second model, based on Benders decomposition. The proposed solution approach is described in Sect. 4. In Sect. 5 we present the computational experience and discuss the results. Conclusions are given in Sect. 6.

2 The problem and the model

We address the problem of planning a home blood donation service over a given territory assigned to a single blood collection center, considering a fixed time horizon. Its main scope is to serve as many donors as possible, provided that some constraints on production and available resources are satisfied. We assume that the blood center is provided with blood storage facilities and vehicles, which are used to reach the

donors at home. It collects the blood from regular blood donors, according to given appointments. Below we detail the considered setting, while an Integer Programming (IP) formulation for the problem is given in Sect. 2.2.

2.1 The considered setting

The territory corresponding to the donation center is partitioned into cells of similar size and population. Each cell is assigned to one vehicle. Each vehicle can make at most two tours in the cell (one in the morning and one in the afternoon), starting from the center and coming back to the center at the end of the route. Each tour is, then, associated with a cell, a vehicle, a day, a part of the day (morning or afternoon) and a set of service stops (donation slots) that can be assigned to the donors (one donor per slot). Therefore, each donation slot has a pre-assigned cell, vehicle, tour, part of the day and time window for the donation. The locations within the cell where the stops must occur (i.e., the actual route that the tour must follow), instead, are not pre-set and depend on the actual location of the donors assigned to the slots, if any.

Each donor can be assigned only to the slots that he/she likes, among those available for his/her cell. The routes are automatically determined given the appointments: each tour starts from the collection center, visits the donors in the appointment order for that tour and then goes back to the collection center. To go from location i to location j, the vehicle will follow an ij shortest path according to the current traffic conditions. The underlying assumption is that the visiting order has no critical effect on the total duration of the tours. This is true if the cells are small enough to guarantee that the travel time from any location i to any location j in the cell plus the donation time at j (which is a fixed time) does not exceed the duration slot.

As an example, consider the case of a cell that has at its disposal a single vehicle and consider a time horizon of one day. The vehicle can make at most one tour ($tour_1$) in the morning and at most one tour ($tour_2$) in the afternoon. Each tour contains three available stops, namely $slot_1^1, slot_2^1$ and $slot_3^1$ for $tour_1$ and $slot_1^2, slot_2^2$ and $slot_3^2$ for $tour_2$. Then, the total number of available slots is six, partitioned into two tours served by the same vehicle (tours are not time-overlapping as $tour_1$ is in the morning and $tour_2$ in the afternoon), that is, up to six donors living in the cell can potentially be served. Assume that there are four donors ($donor_1, \ldots, donor_4$) in the cell. Suppose that $donor_1$ likes only $slot_1^1$, $donor_2$ and $donor_3$ like all the available slots, while $donor_4$ likes $slot_1^1$ and $slot_1^2$. Consider the assignment ass_1: $donor_1$ assigned to $slot_1^1$; $donor_2$ assigned to $slot_1^2$, $donor_3$ assigned to $slot_2^1$, $donor_4$ assigned to $slot_2^2$. Assignment ass_1 corresponds to the activation of both $tour_1$ and $tour_2$. The route of $tour_1$ visits first $donor_1$ (assigned to $slot_1^1$) and then $donor_3$ (assigned to $slot_2^1$), before coming back to the center. The route of $tour_2$ visits first $donor_4$ and then $donor_2$. Consider now assignment ass_2: $donor_4$ assigned to $slot_1^1$; $donor_2$ assigned to $slot_2^1$, $donor_3$ assigned to $slot_3^1$, $donor_1$ not served, as no remaining slot matches the donor's preferences. Assignment ass_2 corresponds to the activation of $tour_1$ only, whose route visits first $donor_4$, then $donor_2$ and finally $donor_3$. Hence, a

tour can be seen as a set of slots (input), while the route that each tour must follow (possibly none) depends on the assignment of donors to slots (output).

Each tour needs some resources, the most important of which is the medical personnel, that is, a nurse and a physician are required to be on board of each vehicle. Therefore, at most a given percentage of the available tours can be activated. In our setting, it is enough to have one single constraint to limit the total number of activated tours over the considered time horizon. However, it is possible to impose further limits and side constraints on the tours (e.g., limiting the maximum number of vehicles that can simultaneously be routed), without affecting the theoretical results we present in Sect. 3. Since each tour is related to a very small portion of territory (cell), we assume that the working time of the onboard medical personnel, the fuel and the conservation time of the collected samples are never exceeded, independently of the actual assignment of the slots to the donors. We also suppose that the considered time windows always allow the vehicle to perform the donation and to reach the next donor or come back to the depot.

The center has a target level for each blood type that must be produced in the considered time horizon. Since the donors are unpaid, not serving a donor may discourage the donor from donating again. For this reason, when a donor is not served, a penalty cost must be paid and the corresponding penalty can be increased in the next round of assignments, so that the probability to serve that donor increases, as well. In principle, not all the donors may be available in the considered time horizon. Therefore, a set of scenarios is defined: in each scenario a given subset of the donors is supposed to be available and each scenario has an associated probability to happen. The problem consists of minimizing the mean penalty costs over the set of the considered scenarios, ensuring that the service does not exceeded the available resources and that, in each scenario, the prescribed amounts of blood are produced. Since the planning horizon (1 week) is much shorter than the shelf life of collected whole blood units (6 weeks), the blood amounts to be produced refer to the whole time horizon and there is no need to add production constraints for sub-periods as in Doneda et al. (2023).

2.2 The complete formulation

We consider a discrete time horizon T, where each $t \in T$ refers to a day of the week. Set B contains the blood types that must be produced (namely: A+, A−, B+, B−, AB+, AB-, 0+, 0−), while the set of all possible donors is denoted by J. Let R be the set of the tours and let I be the set of donation slots. For a slot i, we denote by $v(i)$ the tour to which i belongs. Let $0 \leq \beta \leq 1$ be the percentage of tours that can be activated, that is, due to limited resources, we can use only up to $\beta \mid R \mid$ tours.

We denote by $I_j \subseteq I$ the slots that are suitable for donor $j \in J$, based on his/her preferences and cell. A donor $j \in J$ is associated with a unique blood type, denoted by b_j. Let $J^b \subseteq J$ be the donors corresponding to blood type $b \in B$ and denote by LB_b and UB_b, respectively, the minimum and maximum number of blood units of type $b \in B$ to be produced in the considered time horizon T, that is, the number of donors of type b to be served. A set S of possible scenarios is

considered, each one with an associated probability p_s to occur. In a scenario, a donor may or may not be willing to donate. The set of the donors willing to donate in scenario $s \in S$ are denoted by $J_s \subseteq J$. Let w_j be the penalty to be paid if donor j is willing to donate, but not served.

Let z_v be a variable taking value 1 if tour $v \in R$ has been selected and 0 otherwise. Since i belongs to $v(i)$, if $z_{v(i)} = 1$, all slots j belonging to $v(i)$ became available. Let y_{ij}^s be a binary variable taking value 1 if donor $j \in J$ is assigned to slot $i \in I$ under scenario $s \in S$ and 0 otherwise. The problem can be modeled as in the *complete* formulation (CF) below.

$$\text{CF} \quad \min \sum_{s \in S} p_s \sum_{j \in J_s} w_j \left(1 - \sum_{i \in I_j} y_{ij}^s\right)$$

$$\sum_{v \in R} z_v \leq \beta \, | \, R \, | \tag{1}$$

$$\sum_{i \in I_j} y_{ij}^s \leq 1 \quad j \in J_s, s \in S \tag{2}$$

$$\sum_{j \in J_s : i \in I_j} y_{ij}^s \leq z_{v(i)} \quad i \in I, s \in S \tag{3}$$

$$LB_b \leq \sum_{j \in J^b} \sum_{i \in I_j} y_{ij}^s \leq UB_b \qquad b \in B, s \in S$$

$$\mathbf{z} \in \{0, 1\}^{|R|}$$

$$y_{ij}^s \in \{0, 1\} \qquad i \in I_j, j \in J_s, s \in S \tag{4}$$

The objective function to be minimized measures the expected penalty to be paid for not serving donors. In fact, if donor $j \in J_s$ is not served, then $\sum_{i \in I_j} y_{ij}^s = 0$ and cost w_j is paid. Otherwise, there exists a unique slot i assigned to j, $\sum_{i \in I_j} y_{ij}^s = 1$ and no penalty for not serving j is considered in the objective function. Constraints (1) ensure that we activate at most $\beta \, | \, R \, |$ tours. Constraints (2) guarantee that each donor j is assigned to at most one time slot in I_j. Constraints (3) force a slot to be unassigned (if the corresponding tour has not been selected) or to be assigned to at most one donor per scenario (if the corresponding tour has been activated). Constraints (4) ensure that the number of donations for each blood type is between the prescribed minimum and maximum values. We assume that $| \, J_s \cap J^b \, | \geq LB_b$ for any $b \in B$ and $s \in S$, otherwise the problem is trivially infeasible.

As common in scenario based approaches (Lanzarone et al. 2012; Bacci et al. 2023) the models usually become very difficult to solve when the number of scenarios and, then, the number of constraints and variables of the model, increases. This is true also for model CF. To overcome this issue, we present a second formulation (BF, that we call *projected* model), where a large part of the variables

and the related constraints are projected out according to a Benders decomposition scheme (Benders 1962).

3 The projected formulation

We reformulate the problem by projecting out the **y** variables and the related constraints, so that the remaining problem only includes the **z** variables, which do not depend on the scenarios. CF can be restated in a bilevel fashion as follows, where each inner min refers to a single scenario.

$$\min \sum_{s \in S} p_s \lambda_s$$

$$\sum_{v \in R} z_v \leq \beta \mid R \mid$$

$$\mathbf{z} \in \{0, 1\}^{|R|}$$

for any $s \in S$, $\quad \lambda_s = \min \sum_{j \in J_s} w_j \left(1 - \sum_{i \in I_j} y_{ij}^s \right)$

$$\sum_{i \in I_j} y_{ij}^s \leq 1 \qquad\qquad j \in J_s$$

$$\sum_{j \in J_s : i \in I_j} y_{ij}^s \leq z_{v(i)} \qquad\qquad i \in I$$

$$LB_b \leq \sum_{j \in J^b} \sum_{i \in I_j} y_{ij}^s \leq UB_b \qquad\qquad b \in B$$

$$y_{ij}^s \in \{0, 1\} \qquad\qquad i \in I_j, j \in J_s$$

We now prove that, for a given **z**, each inner min, that we call the *lower level problem* for s and **z**, can be solved as a linear programming problem.

3.1 Solving the lower level problem as a network flow problem

Minimizing the original objective function of the lower level problem is equivalent to optimize $\lambda_s = \sum_{j \in J_s} w_j - \max \sum_{j \in J_s} w_j \sum_{i \in I_j} y_{ij}^s$.

Theorem 1 *The lower level problem can be solved as a max-cost flow problem on an acyclic graph for any $s \in S$ and* **z**.

Proof Consider the following direct acyclic graph $G(s, \mathbf{z})$ (see Fig. 1). The node set $V(s)$ has a source node σ, a destination node τ, an auxiliary node γ, a node for any donor $j \in J_s$, a node for any slot $i \in I$ and a node for any blood type $b \in B$. The arc set $E(\mathbf{z})$ includes: for any $i \in I$, an arc (σ, i) of cost 0 and capacity $z_{v(i)}$; an arc
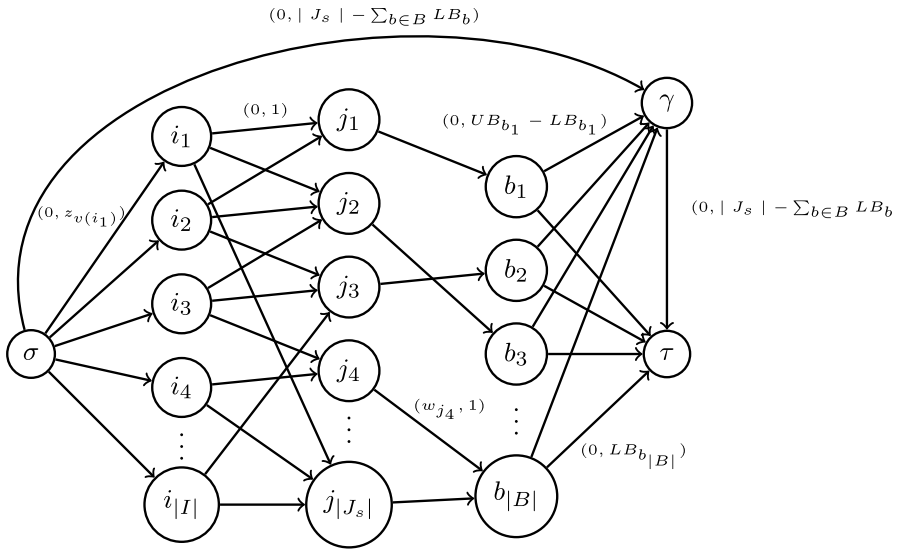
Fig. 1 Picture of the graph defined in Theorem 1. For some representative arcs, we added into brackets the corresponding cost and capacity values, respectively

(σ, γ) of cost 0 and capacity $\mid J_s \mid - \sum_{b \in B} LB_b$; an arc (γ, τ) of cost 0 and capacity $\mid J_s \mid - \sum_{b \in B} LB_b$; for any $b \in B$, an arc (b, τ) of cost 0 and capacity LB_b and an arc (b, γ) of cost 0 and capacity $UB_b - LB_b$; for any $j \in J_s$, an arc (j, b_j) with capacity 1 and cost w_j; for any $i \in I_j$ and $j \in J_s$, an arc (i, j) of cost 0 and capacity 1. Suppose that flow $\mid J_s \mid$ must be sent from σ to τ and that one wants to compute the max-cost flow. If all costs and capacities are integer, the flow will be integer. Since each $j \in J_s$ has only one outgoing arc of capacity 1, at most flow 1 can cross it and the flow will be 1 if j is served (i.e., a slot is assigned to it) and 0 otherwise. A flow 1 on arc (i, j) means that slot i is assigned to donor j. The capacities on the (σ, i) arcs guarantee that slot i is crossed by some flow (i.e., i is assigned to some j) only if the corresponding tour has been activated and the flow that can cross i is at most 1. This means that either i is unassigned or it is assigned to at most one donor. The flow can follow three paths in the graph from σ to τ: $\{(\sigma, \gamma), (\gamma, \tau)\}$; from σ to some b and then to τ using arcs (b, τ); from σ to some b and then to τ via the auxiliary node γ. Node τ must receive flow $\mid J_s \mid$ and it has to come either from nodes for $b \in B$ or from node γ. Since the sum of the capacities on the arcs entering τ is exactly $\mid J_s \mid$, they will be saturated by any feasible solution. Having flow LB_b on arc (b, τ) means that LB_b donors of type b have been assigned to some slots. Having a positive flow on edge (b, γ) means that more than LB_b donors (but less than UB_b) have been served. If there is some flow on arc (σ, γ), it means that there are some unassigned donors. Since we must maximize the cost of the flow, it would be preferable to follow the second or the third path, since the only arcs with a positive cost are the ones in $\{(j, b_j), j \in J_s\}$. If a feasible solution exists, then we can obtain a solution for the lower level problem by assigning slot i to donor j if the flow on arc (i, j) is 1. If the problem is infeasible, that is, it is not possible to send a flow from σ to τ respecting

the given upper and lower bounds on the capacities, this means that it is not possible to obtain the prescribed blood amounts and the lower level problem for the considered s and $\mathbf{z}$ is infeasible. □

The reformulation of the lower level problem as a max-cost flow problem and its dual are reported below. The primal problem for s and $\mathbf{z}$ is denoted by $low(s, \mathbf{z})$ and the dual by $dual(s, \mathbf{z})$. We use the convention that a negative supply/demand value for a node denotes a demand, while a positive one a supply. Variable f_{uv} represents the flow on arc $(u, v) \in E(\mathbf{z})$, being $E(\mathbf{z})$ the arc set defined in the proof of Theorem 1. Symbols on the left in $low(s, \mathbf{z})$ ($dual(s, \mathbf{z})$) are dual (primal) variables associated with the primal (dual) constraints.

$$low(s, \mathbf{z}) \quad \max \sum_{j \in J_s} w_j f_{jb_j}$$

$$(\delta_u) \quad \sum_{v:(u,v)\in E(\mathbf{z})} f_{uv} - \sum_{v:(v,u)\in E(\mathbf{z})} f_{vu} = \begin{cases} |J_s| & u = \sigma \\ -|J_s| & u = \tau \\ 0 & u \in V(s) \setminus \{\sigma, \tau\} \end{cases}$$

$$(\mu_{uv}) \quad f_{uv} \leq \begin{cases} LB_b & (u,v) \in \{(b, \tau), b \in B\} \\ UB_b - LB_b & (u,v) \in \{(b, \gamma), b \in B\} \\ |J_s| - \sum_{b\in B} LB_b & (u,v) \in \{(\sigma, \gamma), (\gamma, \tau)\} \\ z_{v(i)} & (u,v) \in \{(\sigma, i), i \in I\} \\ 1 & \text{otherwise} \end{cases}$$

$$\mathbf{f} \geq \mathbf{0}$$

The constraints corresponding to the dual variable δ are flow conservation constraints, ensuring that the required amount goes from σ to τ. The constraints corresponding to dual variables μ are capacity constraints that guarantee that the capacities on the arcs are not exceeded.

$$dual(s, \mathbf{z}) \quad \min |J_s| (\delta_\sigma - \delta_\tau) + \sum_{b \in B}(LB_b \mu_{b\tau} + (UB_b - LB_b)\mu_{b\gamma})$$

$$+ (|J_s| - \sum_{b \in B} LB_b)(\mu_{\sigma\gamma} + \mu_{\gamma\tau}) + \sum_{j \in J_s} \sum_{i \in I_j} \mu_{ij}$$

$$+ \sum_{i \in I} z_{v(i)} \mu_{\sigma i} + \sum_{j \in J_s} \mu_{jb_j}$$

$$(f_{jb_j}) \quad \delta_j - \delta_{b_j} + \mu_{jb_j} \geq w_j \quad j \in J_s$$

$$(f_{uv}) \quad \delta_u - \delta_v + \mu_{uv} \geq 0 \qquad (u, v) \in E(\mathbf{z}) \setminus \{(j, b_j), j \in J_s\}$$

$$\mu \geq \mathbf{0}$$

Let D_s be the dual feasible region and let $r(D_s)$ and $e(D_s)$ be its extreme rays and vertices, respectively. Note that D_s does not depend on $\mathbf{z}$, which appears only in the objective function of the dual problem. Then, both its rays and its vertices are

independent of the current $\mathbf{z}$. We can now reformulate CF using a Benders decomposition scheme as follows.

Theorem 2 *Problem CF can be reformulated as BF.*

$$BF \quad \min \sum_{s \in S} p_s \lambda_s$$

$$\sum_{v \in R} z_v \leq K$$

$$|J_s| (\delta_\sigma - \delta_\tau) + \sum_{b \in B}(LB_b \mu_{b\tau} + (UB_b - LB_b)\mu_{b\gamma})$$

$$+ (|J_s| - \sum_{b \in B} LB_b)(\mu_{\sigma\gamma} + \mu_{\gamma\tau}) + \sum_{j \in J_s} \sum_{i \in I_j} \mu_{ij} \tag{5}$$

$$+ \sum_{i \in I} \mu_{\sigma i} z_{v(i)} + \sum_{j \in J_s} \mu_{jb_j} \geq 0 \qquad (\boldsymbol{\delta}, \boldsymbol{\mu}) \in r(D_s)$$

$$\lambda_s \geq \sum_{j \in J_s} w_s - (|J_s| (\delta_\sigma - \delta_\tau))$$

$$\sum_{b \in B}(LB_b \mu_{b\tau} + (UB_b - LB_b)\mu_{b\gamma})$$

$$+ (|J_s| - \sum_{b \in B} LB_b)(\mu_{\sigma\gamma} + \mu_{\gamma\tau}) \tag{6}$$

$$+ \sum_{i \in I} \mu_{\sigma i} z_{v(i)} + \sum_{j \in J_s} \mu_{jb_j}) \qquad (\boldsymbol{\delta}, \boldsymbol{\mu}) \in e(D_s)$$

$$\mathbf{z} \in \{0, 1\}^{|R|}, \lambda \geq \mathbf{0}$$

Proof Problem $low(s, \mathbf{z})$ cannot be unbounded: it may admit a feasible solution or it may be infeasible. On the other hand, problem $dual(s, \mathbf{z})$ always admits the feasible solution where all the variables are set to 0, apart from μ_{jb_j}, which are set to w_j. Hence, it can be unbounded or it may admit an optimal solution, but it is never infeasible. If there exists at least a scenario $s \in S$ whose corresponding $low(s, \mathbf{z})$ is infeasible, then $\mathbf{z}$ is infeasible and must be cut off. Requiring that $low(s, \mathbf{z})$ is not infeasible for any s for the given $\mathbf{z}$, corresponds to requiring that $dual(s, \mathbf{z})$ is not unbounded. Hence, the dual objective function must assume a positive value for any extreme ray in $r(D_s)$. This is obtained by the Benders feasibility cuts (5). On the other hand, we must also ensure that the computed λ_s values correspond to optimal values for the lower level problem $low(s, \mathbf{z})$. The weak duality theory ensures that, for any feasible solution of the primal problem and any feasible solution of the dual problem, the primal objective value is lower than or equal to the dual objective value. This condition is implemented by the Benders optimality cuts (6). The strong duality theory ensures that there is at least one vertex (which may be different for any $\mathbf{z}$ and s) for which the constraint is satisfied with equality, guaranteeing that the corresponding primal and dual objective values are optimal. □

Formulation BF has less variables than CF, but an exponential number of constraints. However, the problem can be solved via branch-and-cut, by generating Benders feasibility cuts (5) and Benders optimality cuts (6) dynamically. Usually, the optimization process terminates when only a very limited number of inequalities has been added, without the need to consider each vertex and ray of the dual polyhedron. In problem $dual(s, \mathbf{z})$, variables $\mathbf{z}$ are parameters, as they are fixed to given values, while $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ are the variables whose value must be computed. On the contrary, in constraints (5) and (6), $\mathbf{z}$ are variables, while $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ are fixed to given values and, hence, they are coefficients. Also note that, if we replace the single inequality (1) by any set of constraints $\mathbf{Qz} \leq \mathbf{q}$ that may limit the available tours and/or add objective costs for the $\mathbf{z}$ variables, these modifications do not affect the subproblems and, hence, the results used to derive reformulation BF.

3.2 Benders feasibility cuts

Feasibility has a natural interpretation by network flow arguments. Let a $\sigma\tau$ cut $\{W : V(s) \setminus W\}$ be a partition of the nodes $V(s)$ of graph $G(s, \mathbf{z})$, such that $\sigma \in W$ and $\tau \in V(s) \setminus W$. The arcs (u, v) of the cut are the ones having $u \in W$ and $v \in V(s) \setminus W$. The capacity of the cut is the sum of the capacities on the arcs of the cut.

Theorem 3 *The problem of checking if $low(s, \mathbf{z})$ is feasible, amounts to test if there is a $\sigma\tau$ cut in $G(s, \mathbf{z})$ whose capacity is strictly less than $|J_s|$.*

Proof By the max-flow min-cut theorem (Ford and Fulkerson 1956), each flow from the source to the destination must cross each cut that separates the source from the destination. If there exists one of such cuts whose capacity is less than the demand that must cross the cut, then the problem is infeasible. Since the costs play no role, as far as feasibility is concerned, suppose that $w_j = 0$ for all $j \in J$. Consider $dual(s, \mathbf{z})$. Assume that there exists a $\sigma\tau$ cut $\{W : V(s) \setminus W\}$ whose capacity is less than the demand. Consider the solution having $\mu_{uv} = 1$ for the arcs of this cut and 0 otherwise. Let $\delta_u = 1$ for $u \in V(s)$ and $\delta_u = 0$ otherwise. This solution is feasible for $dual(s, \mathbf{z})$ and, since the capacity of the considered cut is less than the demand, the corresponding objective function is negative. Moreover, it can be scaled by an arbitrary factor, leading to unboundedness. Hence, each unbounded ray corresponds to a $\sigma\tau$ cut with capacity less than the demand in $G(s, \mathbf{z})$. On the other hand, if each $\sigma\tau$ cut has a capacity greater than or equal to the demand, then the primal problem admits an optimal solution and so it is for the dual by linear duality. It follows that there is no unbounded ray. □

Then, each Benders feasibility cut corresponds to a $\sigma\tau$ cut of graph $G(s, \mathbf{z})$. We describe below some Benders feasibility cuts that can be added to the initial model to avoid separating trivially infeasible $\mathbf{z}$ vectors.

Theorem 4 *The source inequalities (7) are Benders feasibility cuts.*

$$\sum_{i\in \cup_{j\in J_s} I_j} z_{v(i)} \ge \sum_{b\in B} LB_b \quad s \in S \tag{7}$$

Proof Constraints (7) are valid for BF. In fact, if one does not have the slots to accommodate the minimum number of donors needed to reach the required blood mix, then $low(s, \mathbf{z})$ is infeasible. They correspond to the $\sigma\tau$ cuts $\{\{\sigma\} : V(s)\backslash\{\sigma\}\}$, for $s \in S$. This cut prescribes that the edges going from σ to any node $i \in I$, plus edge (σ, γ) have enough capacity to support an amount of flow corresponding to the number of donors of scenario s, that is, $| J_s |$. Since edge (σ, τ) can accommodate at most flow $| J_s | - \sum_{b\in B} LB_b$, then the edges from σ to $i \in I$ must have capacity at least $\sum_{b\in B} LB_b$, leading to inequality (7). The cut is illustrated in Fig. 2. ☐

According to the blood type, we can derive the inequalities below.

Theorem 5 *The blood type inequalities* (8) *are Benders feasibility cuts.*

$$\sum_{i\in \cup_{J^b \cap J_s} I_j} z_{v(i)} \ge LB_b \quad b \in B, s \in S \tag{8}$$

Proof Constraints (8) are valid for BF. If one does not have enough slots to accommodate the minimum number of donors needed to reach the required minimum amount for blood type b, then $low(s, \mathbf{z})$ is infeasible. For a given b and s, inequality (8) corresponds to the $\sigma\tau$ cut $\{V(s) \backslash \{\tau, b, J(b), I(b)\} : \{\tau, b, J(b), I(b)\}\}$, where $J(b) = J^b \cap J_s$ are the donors of blood type b and $I(b) = \cup_{j\in J(b)} I_j$ are the slots that can be assigned to these donors. This cut has capacity $\sum_{i\in I(b)} z_{v(i)}$ (edges from σ to
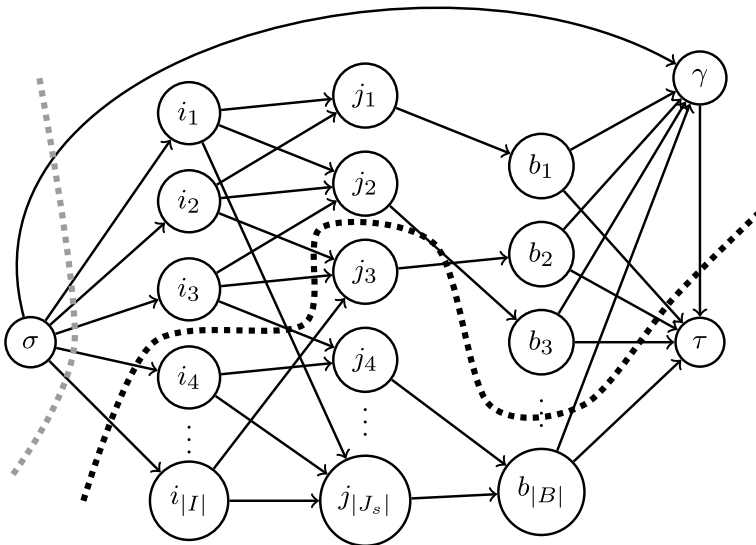

Fig. 2 In the picture, the dotted lines represents the cuts associated with an inequality of type (7) (in gray) and with an inequality of type (8) (in black)

$i \in I(b))$ $+\sum_{w \in B \setminus \{b\}} LB_w$ (edges from $w \in B \setminus \{b\}$ to τ) $+ \mid J_s \mid - \sum_{w \in B} LB_w$ (edge (γ, τ)). This capacity can support a total flow of amount $\mid J_s \mid$ only if $\sum_{i \in I(b)} z_{v(i)} \geq LB_b$, leading to inequalities (8). The cut is illustrated in Fig. 2. □

Algorithm 1 Processing a node of the branch-and-bound tree

---

    let $bestSol$ be the value of the best solution found so far
1: solve the linear relaxation of the problem at the node
    let $(\mathbf{z}, \boldsymbol{\lambda})$ and currentSol be the current solution and the corresponding value
2: **if** $currentSol > bestSol$ **then**
3:     prune the node and exit
4: **else**
5:     **if** $\mathbf{z}$ is integer or we are at the root node **then**
6:         $R = S$
7:         **repeat**
8:             choose $s \in R$
9:             $R = R \setminus \{s\}$
10:             **if** $low(s, \mathbf{z})$ is infeasible **then**
11:                 add inequality (5) for the unbounded dual ray
12:             **else**
13:                 **if** $low(s, \mathbf{z})$ admits optimal solution $\mathbf{f}^*$ **then**
14:                     **if** $\lambda_s < \sum_{j \in J_s} w_s - \sum_{j \in J_s} w_j f^*_{jb_j}$ **then**
15:                         add inequality (6) for the dual vertex for $\mathbf{f}^*$
16:                     **end if**
17:                 **end if**
18:             **end if**
19:         **until** $R = \emptyset$ or an inequality of type (5) is separated
20:         **if** no inequalities (5) or (6) are separated **then**
21:             **if** $(\mathbf{z}, \boldsymbol{\lambda})$ is integer **then**
22:                 $(\mathbf{z}, \boldsymbol{\lambda})$ is a feasible solution of BF
23:                 exit
24:             **else**
25:                 branch and exit
26:             **end if**
27:         **else**
28:             go to step 1
29:         **end if**
30:     **else**
31:         branch and exit
32:     **end if**
33: **end if**

---

4 The algorithm

Since formulation BF has an exponential number of constraints, the problem is solved by branch-and-cut, generating inequalities (5) and (6) dynamically. A branch-and-cut approach (Padberg and Rinaldi 1991) is a branch-and-bound (see, e.g., Wolsey (1998)) that starts from a suitable reduced formulation of the problem and where each node of the tree is processed by applying a user defined policy, which generates additional inequalities to be added to the model on the fly. In our case, the initial formulation that is used to solve BF does not include inequalities (5) and (6) and it only contains the constraint that limits the number of tours that can be activated (1) and inequalities (8), that are polynomially many in the input size.

Indeed, it is well-known that Benders decomposition approaches can be very ineffective in the first iterations, when few constraints are present in the formulation and, hence, the generated solutions are often infeasible or far from the optimum. This effect can be mitigated by adding to the initial formulation a set of cuts, like constraints (8), that allows to generate more reasonable initial solutions, thus reducing the overall time and speeding up convergence. In Sect. 5.3 we discuss how effective these inequalities are, by showing what happens if we remove them from the initial formulation. The user defined policy (*separation*) is described below, where we discuss further strategies that have been adopted to improve the overall algorithm.

4.1 Separation

Consider a node of the branch-and-bound tree. If it is not the root node and the optimal solution of the linear relaxation $(\mathbf{z}, \lambda)$ is fractional, we just branch, as in a standard branch-and-bound. Instead, if we are at the root node or $(\mathbf{z}, \lambda)$ is integer, we check if $(\mathbf{z}, \lambda)$ violates some inequalities (5) or (6). Therefore, for each scenario $s \in S$, we solve $low(s, \mathbf{z})$. If $low(s, \mathbf{z})$ is infeasible, the dual is unbounded and a constraint (5) must be generated and added to the current formulation. If $low(s, \mathbf{z})$ is feasible, but λ_s does not correspond to the optimal value, a constraint (6) must be added to the current model. We add a single inequality (5) at a time but, possibly, multiple inequalities (6).

In fact, if a feasibility cut is violated by a solution $\mathbf{z}$ for a scenario s, then there is a high probability that it will be violated for other scenarios r. Indeed, if we cannot satisfy some lower bounds LB_b with $| J_s \cap J^b |$ potential donors for type b, it is unlikely to satisfy them with $| J_r \cap J^b | \leq | J_s \cap J^b |$ donors, despite the preferences of those donors could better match the activated tours. This observation was confirmed in the experiments, where, for a given current solution $(\mathbf{z}, \lambda)$, most of the times the different scenarios produced very similar (if not the same) feasibility cuts. Hence, at each iteration, we decided to add only the first violated feasibility cut that we generate among all scenarios. Moreover, fractional solutions are tested only at the root node. This avoids generating many inequalities that: although globally valid, are possibly violated only for one single node, whereas in the others are not needed and just increase the problem

size; are possibly violated by small amounts, leading to small changes between one fractional solution and the next, while branching would be more effective. All integer solutions are tested at each node to guarantee the correctness of the algorithm.

If $(\mathbf{z}, \lambda)$ is integer and it does not violate any inequality (5) or (6), then it is, indeed, a feasible solution that can be used to possibly update the incumbent solution, that is, the best solution computed so far. The node procedure is illustrated in detail in Algorithm 1, where we denote by *bestSol* the value of the best solution found so far and by *currentSol* the value of the current solution $(\mathbf{z}, \lambda)$. Step 10 of Algorithm 1 requires checking if $low(s, \mathbf{z})$ is feasible and, if the case, producing a violated inequality corresponding to an unbounded ray. As already discussed in Sect. 3, $low(s, \mathbf{z})$ can be modeled as a network flow problem, leading to the following result.

Theorem 6 *Checking if $low(s, \mathbf{z})$ is feasible or generating a violated inequality* (5) *if not, can be done in polynomial time.*

Proof Checking if $low(s, \mathbf{z})$ is feasible amounts to verify if there is a cut whose capacity is less than $|J_s|$. This can be done by applying a max-flow algorithm to the graph $G(s, \mathbf{z})$ corresponding to the current solution. The maximum flow in a graph can be computed in polynomial time (Ahuja et al. 1993). If the maximum flow is less than $|J_s|$, the corresponding minimum cut has capacity less than $|J_s|$ and it corresponds to an unbounded ray by Theorem 3. □

If $low(s, \mathbf{z})$ is not infeasible, step 13 requires producing an optimal solution of the problem. Again, by the network flow argument illustrated in Sect. 3, the following result holds.

Theorem 7 *Computing an optimal solution for $low(s, \mathbf{z})$ or generating a violated inequality* (6) *if any, can be done in polynomial time.*

Proof Problem $low(s, \mathbf{z})$ is a max cost flow problem on an acyclic graph. Max cost flows on acyclic graphs and corresponding dual variables can be computed in polynomial time Ahuja et al. (1993). □

4.2 The implementation

We implemented the code in Python and used the Gurobi (gurobipy) libraries 9.1.1 for solving the linear (integer) programming problems. Model BF is solved using Algorithm 1 at each node of the branch-and-bound tree. Model CF is solved using Gurobi with default settings. We ran the code on a 3 GHz Intel Xeon Gold 6136 machine, with 258 GB of RAM. For the solution of each instance, we set a time limit of 3600 s.

5 The computational experiments

Below we present and discuss the computational experiments that have been carried out using the models presented in Sects. 2.2 and 3.

5.1 The test bed

Since home blood donation has not been treated in practice and it was addressed only marginally in the literature, there are no reference instances that can be used as a benchmark. We started from data provided by AVIS Milan, which is not currently offering the possibility of home blood donations, and derived some instances for the home blood donation problem. These same data have been already used by Doneda et al. (2021) and Baş Güre et al. (2018a). According to AVIS Milan, we assume that the 60% of its donors are served by the home donation service, while the others continue to make the donations at the blood collection centers. Therefore, 60% of donation requests have been randomly extracted from the historical data and included in the simulations. In this way, the considered dataset includes a set J of 3865 donors who booked an appointment during a period of 120 days, starting from January 1st 2014. Therefore, in our experiments, we have on average $\pi = 3865/120 \approx 32$ donors to be served every day (equal to 60% of the about 50 daily donors of AVIS Milan).

AVIS Milan is located in the Eastern part of the city and collects blood from donors living or working in Milan. The city of Milan has an extension of 181.8 km^2 and an almost circular shape. The geodesic positions of the donors are uniformly generated in a circle with radius 7.6 km, where the uniform distribution is in agreement with the characteristics of Milan and the experience of AVIS Milan. The area is divided into 9 zones (cells): the city center, with a radius of 2.5 km and 8 suburban sectors of 45° each. We assume that each cell is visited twice a day by a tour, once in the morning and once in the afternoon. Each tour can serve 3 donors and, therefore, we can serve up to 54 donors per day. The geodesic position of each donor is used to place the donor in the corresponding cell. For the 9 cells we consider, the set R is made by $9 \times 2 = 18$ tours for each day and set I contains $18 \times 3 = 54$ slots per day. We set a penalty $w_j = 1$ for each slot $j \in J$.

The preferences of the donors for the slots (sets I_j) are randomly generated by adding to I_j any slot $i \in I$ belonging to a tour that visits the cell of donor j with an independent probability α. There are $| B | = 8$ blood types, whose distribution among the donors is illustrated in Table 1. For each $b \in B$, we let LB_b and UB_b be 15% and 150%, respectively, of the values reported in Table 1. For each scenario $s \in S$, we set a uniform probability $p_s = 1/ | S |$ to occur and we construct the set J_s of donors in s by randomly picking each donor in J with probability $7\pi/ | J |$.

Each group of instances is represented by the triple $\{| S |, \beta, \alpha\}$ and contains 10 instances with the same value of the parameters. Parameter $| S |$ represents the number of considered scenarios: it takes values in $\{200, 400, 600, 800, 1000\}$ and gives an indication of the size of the instances. Parameter β is the percentage of

Table 1 Distribution among the blood types of the donors in the dataset

Blood type		Percentage
AB	+	3.6
AB	–	0.5
A	+	34.1
A	–	5.8
B	+	9.6
B	–	1.6
0	+	38.0
0	–	6.8

the tours that can be activated: it takes values in $\{50\%, 70\%, 90\%\}$ and corresponds to the available resources. Parameter α is the probability that donor j likes slot i: it takes values in $\{25\%, 50\%, 75\%\}$ and represents a measure of the difficulty of satisfying the donors preferences. Hence, the considered test bed contains a total number of 450 instances. Observe that, as already mentioned, if we activate all available tours, we can serve up to 54 donors per day. On the other hand, the average number of daily donors of each scenario is $\pi \approx 32$. Therefore, any donor has, on average, almost two alternative dedicated slots for donating blood, when all tours are activated, and it is possible to serve a large portion of the donors, even with difficult donors preferences (small values of parameter α).

5.2 Comparing CF and BF

In Table 2 we present the computational results obtained by solving CF and BF on the instances with 200 scenarios described in Sect. 5.1. We limit to 200 scenarios because CF suffers when $|S|$ increases and cannot solve any instance with more than 200 scenarios and with a percentage of available tours less than 90%. Each row of the table is associated with the group of 10 instances corresponding to the

Table 2 Comparison between model CF and model BF on instances with 200 scenarios

$|S|$	β (%)	α (%)	CF		BF	
			Solved	Time	Solved	Time
200	50	25	0		5	671.7
		50	0		10	344.7
		75	0		10	525.9
	70	25	2	2753.6	5	234.5
		50	0		9	460.4
		75	0		9	451.7
	90	25	10	149.0	10	75.3
		50	10	190.3	10	141.4
		75	10	375.9	10	210.8

indicated parameter values $| S |$, β, and α. For each model, column *solved* denotes the number of instances of the group that are solved to optimality within the time limit, while column *time* reports the average computing time (in seconds) on the solved instances. The results show that model BF outperforms CF, both in terms of computing times and of solved instances. Also note that $\beta = 90\%$ means that the available slots are about 1.5 times the average number of donors in the scenarios. It follows that most **z** choices do not exceed the available resources and are sufficient to serve almost all the donors, reducing the importance and the effect of such variables in the model. As we discussed in Theorem 1, the **z** variables are the only true integer variables of the model, while the **y** ones can be treated as continuous variables. Therefore, it is to be expected that instances with $\beta = 90\%$ are much easier than the others for both approaches. In fact, both CF and BF can solve all the instances with $\beta = 90\%$, independently of the values of the other parameters.

We also observe that, when neither CF nor BF can solve an instance to optimality, the best solution produced by CF at the time limit serves, on average, about 4% less donors than the corresponding BF solution in the instances with $| S |= 200$. The difference increases to about 17% on the instances with $| S |= 400$.

5.3 Statistics on solving BF

In Tables 3, 4, we focus on the model BF and present in detail the computational results for it, using instances up to 1000 scenarios. The former refers to the instances solved to optimality, the latter to the ones not solved to optimality within the time limit.

In Table 3, for each group of instances, the number of solved instances is reported in column *solved*. For such instances, columns *time*, *nodes*, *F-cuts* and *O-cuts* give, respectively, the average computing time (in seconds), the average number of explored nodes, the average number of generated feasibility cuts (5), the average number of generated optimality cuts (6). Observe that BF can solve most of the instances even when the number of scenarios $| S |$ goes up to 1000. Because of the cut generation procedure we adopted, the number of feasibility cuts is very limited with respect to the number of optimality cuts.

In Table 4, we analyze the performances of BF over the instances that it could not solve to optimality within the time limit of 3600 s. Here, column *unsolved* reports the number of the unsolved instances of each group. Column *gap (%)* is the average percentage optimality gap when the procedure stops and column *nodes* is the average number of branch-and-bound nodes. These values show that, even when the solution provided within the time limit is not proved to be optimal, it is, nevertheless, definitively a good solution. In fact, with only one exception on an instance with 1000 scenarios, the optimality gap values are always below 0.5%.

Table 5 measures the effect of inequalities (8) in solving BF. It reports a comparison on the instances with 200 scenarios between two settings of the algorithm: inequalities (8) added in the initial formulation (column *BF with* (8)); inequalities (8) separated as feasibility cuts (column *BF without* (8)). We report the number of instances that are solved to optimality within the 1 h time limit in both

Table 3 Results on the solved instances for BF

$|S|$	β (%)	α (%)	BF				
			Solved	Time	Nodes	F–cut	O–cut
200	50	25	5	671.7	98388.6	0.8	2486.4
		50	10	344.7	365.3	1.7	2728.2
		75	10	525.9	121513.9	2.2	2961.8
	70	25	5	234.5	32437.0	0.4	2080.8
		50	9	460.4	35047.7	1.4	1877.8
		75	9	451.7	777.1	1.8	2057.4
	90	25	10	75.3	26.8	0.6	913.0
		50	10	141.4	128.7	1.3	761.0
		75	10	210.8	52.3	1.8	1112.9
400	50	25	7	942.9	14823.1	0.6	5405.6
		50	9	757.5	412.4	1.6	6459.4
		75	10	1172.2	581.6	1.8	6882.7
	70	25	7	697.1	29819.3	0.4	4287.4
		50	8	777.9	1377.8	1.1	3855.4
		75	10	1038.6	583.6	1.4	4043.4
	90	25	10	200.0	174.2	0.5	1869.4
		50	10	365.9	52.2	1.2	2031.9
		75	10	531.1	84.7	1.4	2634.9
600	50	25	6	1472.5	2404.3	0.5	8226.7
		50	9	1591.6	584.3	1.8	10997.2
		75	10	2060.9	591.4	2.0	10571.1
	70	25	4	847.2	539.3	0.0	6983.8
		50	6	1284.4	525.5	1.0	7468.7
		75	10	1463.1	596.6	1.5	6508.6
	90	25	10	275.0	1.0	0.3	3008.5
		50	10	647.2	124.1	1.0	3660.7
		75	10	805.6	52.6	1.5	3950.3
800	50	25	5	1751.4	1397.6	0.0	11186.2
		50	8	1621.8	860.5	1.3	12018.9
		75	8	2060.9	989.0	1.4	11630.6
	70	25	7	1394.0	1956.7	0.6	9546.9
		50	6	2190.2	3803.0	0.8	10058.2
		75	10	2196.1	955.9	1.4	9443.2
	90	25	10	442.3	158.5	0.4	4363.7
		50	10	752.3	52.6	0.8	5071.6
		75	10	1086.5	155.2	1.4	6287.6

Table 3 (continued)

$|S|$	β (%)	α (%)	BF				
			Solved	Time	Nodes	F–cut	O–cut
1000	50	25	4	2423.4	1595.3	0.0	17212.8
		50	8	2137.6	591.3	0.9	17335.1
		75	7	2775.8	1216.3	2.0	15654.6
	70	25	8	1795.7	4625.4	0.4	12319.4
		50	7	2356.7	834.0	0.9	12259.0
		75	8	2500.3	560.9	1.1	11161.5
	90	25	10	482.3	56.7	0.2	5687.9
		50	10	1049.3	565.1	0.6	6212.5
		75	10	1383.3	256.8	1.3	6245.1

Table 4 Results on the unsolved instances for BF

| $|S|$	β (%)	α (%)	BF				
			Unsolved	Nodes	Gap (%)	F–cut	O–cut
200	50	25	5	491162.2	0.04	0.4	2838.6
	70	25	5	1585928.6	0.06	0.6	2187.0
		50	1	1358491.0	0.08	2.0	1783.0
		75	1	1285237.0	0.06	3.0	1889.0
400	50	25	3	63924.7	0.04	0.3	6184.0
		50	1	128545.0	0.01	1.0	5456.0
	70	25	3	361729.3	0.07	0.7	4208.7
		50	2	312602.0	0.14	1.5	3426.5
600	50	25	4	19890.3	0.03	0.5	13664.3
		50	1	165788.0	0.45	1.0	7393.0
	70	25	6	206902.5	0.12	0.3	6329.8
		50	4	133572.0	0.11	1.3	5682.3
800	50	25	5	6193.6	0.06	1.2	14790.8
		50	2	52366.5	0.34	1.0	28490.0
		75	2	258.5	0.10	4.5	26329.0
	70	25	3	131134.3	0.06	0.0	9674.0
		50	4	55173.0	0.09	0.8	10150.0
1000	50	25	6	6890.5	0.03	0.8	18883.8
		50	2	11041.0	0.01	3.0	22743.0
		75	3	180.3	18.51	1.7	28111.0
	70	25	2	69308.5	0.07	0.5	11795.5
		50	3	38541.3	0.14	0.3	11717.3
		75	2	37043.0	0.23	2.0	10389.0
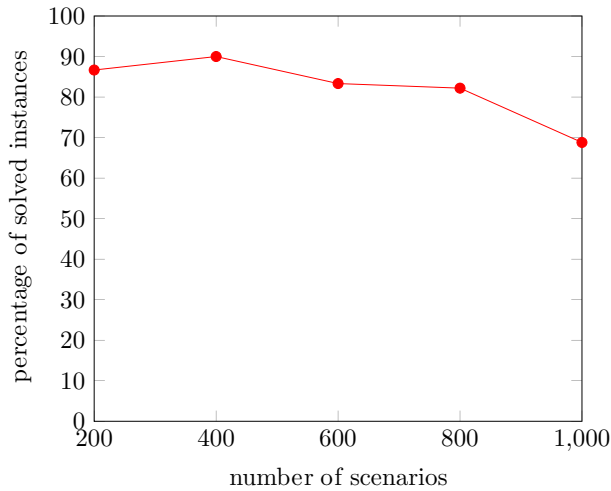
Fig. 3 Percentage of solved instances depending on the number $|S|$ of scenarios for BF
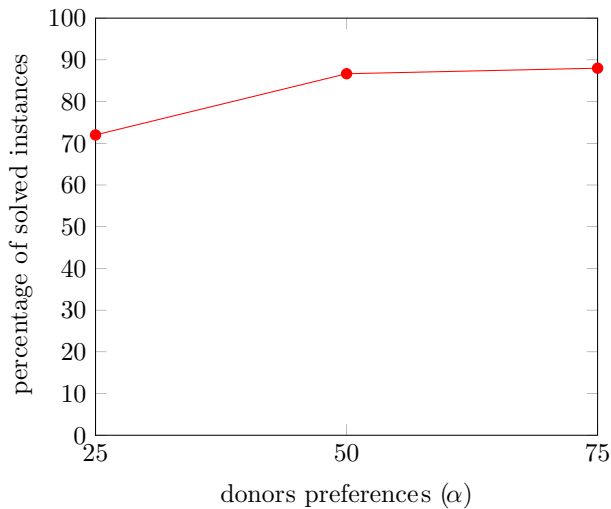

Fig. 4 Percentage of solved instances depending on parameter α for BF

settings. If the second setting is used for BF, the number of instances solved to optimality dramatically decreases, even in the easy case where $\beta = 90\%$.

In Figs. 3, 4 and 5 we show how the average number of solved instances changes when, respectively, the number of considered scenarios, the value of parameter α and the value of parameter β change. The difficulty of the problem grows with the number of scenarios and decreases with the increasing of parameters α and β. Indeed, the greater is the value of these two parameters, the easier is to assign donors to slots.
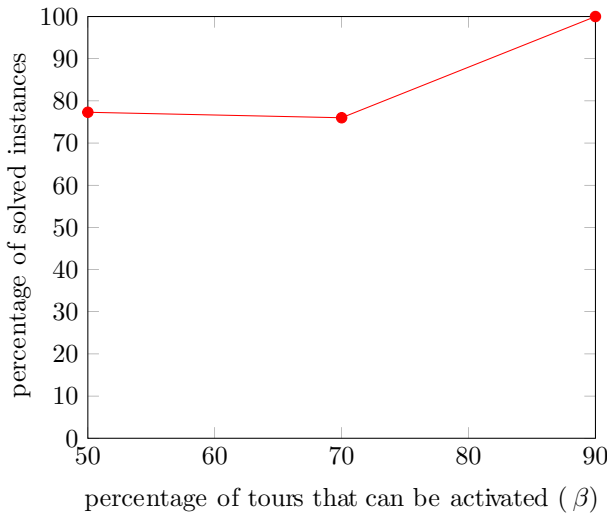
Fig. 5 Percentage of solved instances depending on parameter β for BF

Table 5 The role of constraints (8) for BF

$|S|$	β (%)	α (%)	BF with (8) Solved	BF without (8) Solved
200	50	25	5	0
		50	10	3
		75	10	1
	70	25	5	0
		50	9	0
		75	9	1
	90	25	10	10
		50	10	4
		75	10	10

5.4 Served donors

The main objective of the blood donation planning described in the present paper is to serve as many donors as possible, provided that some constraints are satisfied. In Tables 6 and 7 we report the average number of served donors (column *percentage of served donors*), respectively, for the solved instances and for the unsolved ones (served donors in the best solution found). The results show that the number of served donors is always very large, also for the instances not solved to optimality. The only instance where the served donors are less than 81% is the unique unsolved instance with 1000 scenarios with gap around 18% that was discussed before.

In Figs. 6, 7 and 8 we summarize how the percentage of served donors change, depending on the number of scenarios, the donors preferences (α) and the percentage

Table 6 Served donors on the solved instances for BF (optimal solution)

$|S|$	β (%)	α (%)	BF Percentage of served donors
200	50	25	82.0
		50	81.9
		75	81.9
	70	25	97.8
		50	97.8
		75	97.8
	90	25	99.8
		50	99.8
		75	99.8
400	50	25	81.8
		50	81.8
		75	81.8
	70	25	97.8
		50	97.7
		75	97.7
	90	25	99.8
		50	99.8
		75	99.8
600	50	25	81.7
		50	81.8
		75	81.8
	70	25	97.7
		50	97.7
		75	97.7
	90	25	99.8
		50	99.8
		75	99.8
800	50	25	81.7
		50	81.7
		75	81.8
	70	25	97.7
		50	97.7
		75	97.7
	90	25	99.8
		50	99.8
		75	99.8

Table 6 (continued)

$|S|$	β (%)	α (%)	BF Percentage of served donors
1000	50	25	81.8
		50	81.7
		75	81.8
	70	25	97.7
		50	97.7
		75	97.7
	90	25	99.8
		50	99.8
		75	99.8

Table 7 Served donors on the unsolved instances for BF (best available solution)

$|S|$	β (%)	α (%)	BF Percentage of served donors
200	50	25	81.9
	70	25	97.7
		50	97.6
		75	97.7
400	50	25	81.9
		50	81.9
	70	25	97.6
		50	97.6
600	50	25	81.9
		50	81.7
	70	25	97.7
		50	97.7
800	50	25	81.8
		50	81.8
		75	81.7
	70	25	97.7
		50	97.7
1000	50	25	81.7
		50	81.8
		75	74.9
	70	25	97.7
		50	97.7
		75	97.7
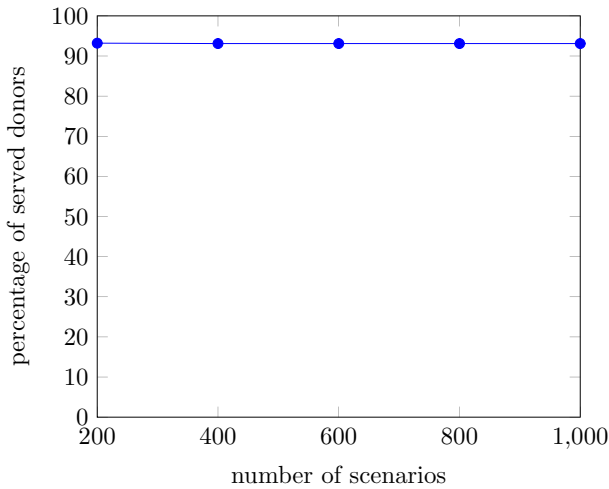
Fig. 6 Percentage of served donors depending on the number $|S|$ of scenarios for BF
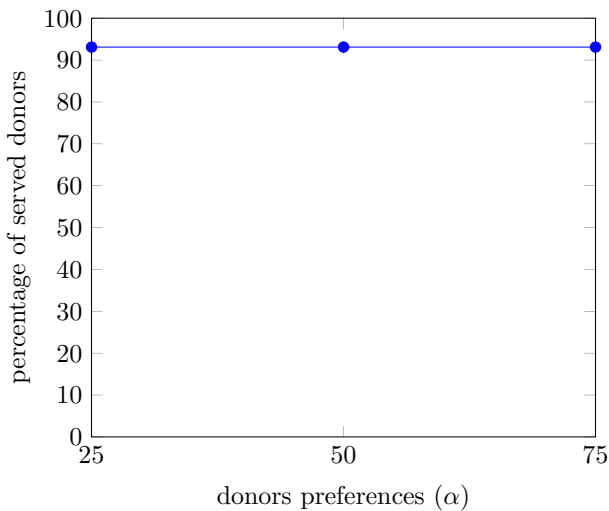
Fig. 7 Percentage of served donors depending on parameter α for BF

of tours that can be activated, given the available resources (β). Although the preferences of the donors cannot be controlled, it is easy to see that they are not crucial and that it is possible to serve a large number of donors even when each of them only likes 25% of the proposed slots. On the other hand, as expected, the value of parameter β largely affects the number of served donors. However, even with a small percentage of available tours ($\beta = 50\%$) it is possible to serve more than 80% of the considered donors. Observe that, when $\beta = 50\%$, the number of available slots is about 84% of the average number of donors in each scenario. An increase in the
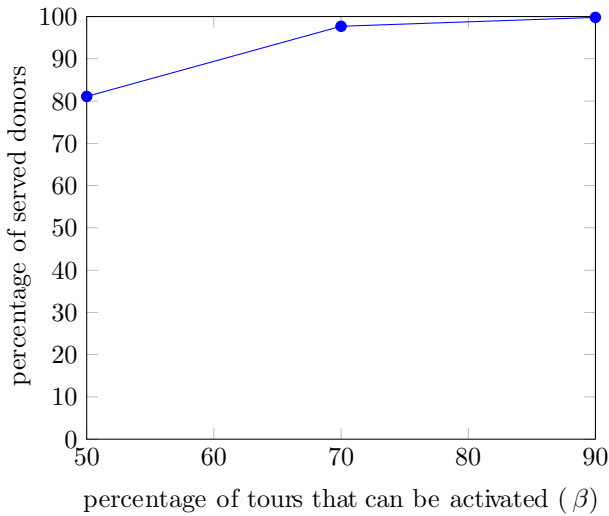
Fig. 8 Percentage of served donors depending on parameter β for BF

percentage of available tours from 50 to 70% allows to serve almost all the donors (about 98%). Observe that, further increasing the resources dedicated to the service, that is, increasing β from 70 to 90% allows to increase the number of served donors by a maximum of 2%. Also note that, in principle, it may not be possible to serve all the donors, due to the upper bounds on the blood production, independently of the available resources.

6 Conclusions

We considered the problem of planning home blood donations, which has received only a very limited attention in the literature so far. The uncertain availability of the donors is modeled in a stochastic way, by scenarios. The aim was to serve as many donors as possible, while respecting a bound on the available resources, that limits the number of tours that can be activated, and ensuring that prescribed amounts of each blood type are produced in each scenario. For this problem, we proposed an integer programming formulation that is solved using a Benders decomposition approach, via a network flow argument. The results show that the proposed framework can produce good solutions, where a large portion of the donors are served. They also show how to control the resources that are needed to produce the desired level of service. In particular, activating about 50% of the tours already ensures to serve more than 80% of the donors.

References

Achterberg T, Raack C (2010) The MCF-separator: detecting and exploiting multi-commodity flow structures in MIPs. Math Program Comput 2:125–165

Ahuja R, Magnanti T, Orlin J (1993) Network flows: theory, algorithms and applications. Prentice Hall, Englewood Cliffs

Alfonso E, Xie X, Augusto V et al (2012) Modeling and simulation of blood collection systems. Health Care Manag Sci 15:63–78

Alfonso E, Augusto V, Xie X (2015) Mathematical programming models for annual and weekly bloodmobile collection planning. IEEE Trans Autom Sci Eng 12:96–105

Alfonso E, Xie X, Augusto V (2015b) A simulation-optimization approach for capacity planning and appointment scheduling of blood donors based on mathematical programming representation of event dynamics. In: 2015 IEEE international conference on automation science and engineering (CASE), pp 728–733

Anaya-Arenas AM, Prodhon C, Renaud J et al (2021) An iterated local search for the biomedical sample transportation problem with multiple and interdependent pickups. J Op Res Soc 72:367–382

Arbib C, Servilio M, Ventura P (2017) An improved integer linear programming formulation for the closest 0–1 string problem. Comput Oper Res 80:94–100

Avella P, Mattia S, Sassano A (2007) Metric inequalities and the network loading problem. Discret Optim 4:103–114

Avella P, Boccia M, Mattia S (2013) Mixed integer lifted cover inequalities for knapsack problems with a single continuous variable. In: Proceedings of 5th international conference on modeling, simulation and applied optimization (ICMSAO) 2013

Avella P, Boccia M, Mattia S et al (2021) Weak flow cover inequalities for the capacitated facility location problem. Eur J Oper Res 289:485–494

AVIS (2020) Statistiche: la presenza di AVIS nel territorio. www.avis.it/dati-donazioni, Accessed Sept 2023

Bacci T, Mattia S, Ventura P (2023) The realization-independent reallocation heuristic for the stochastic container relocation problem. Soft Comput 27(7):4223–4233

Baş Güre S, Carello G, Lanzarone E et al (2016) Management of blood donation system: literature review and research perspectives. In: Matta, A, Sahin, E, Li, J, Guinet, A, Vandaele, N (eds) Health care systems engineering for scientists and practitioners. Springer proceedings in mathematics & statistics, vol 169. Springer, Cham, pp 121–132

Baş Güre S, Carello G, Lanzarone E et al (2018) An appointment scheduling framework to balance the production of blood units from donation. Eur J Oper Res 265(3):1124–1143

Baş Güre S, Carello G, Lanzarone E et al (2018) Unaddressed problems and research perspectives in scheduling blood collection from donors. Prod Plan Control 29(1):84–90

Beliën J, Forcé H (2012) Supply chain management of blood products: a literature review. Comput Oper Res 217:1–16

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. Numer Math 4:238–252

Bienstock D, Mattia S (2007) Using mixed-integer programming to solve power grid blackout problems. Discret Optim 4:115–141

Bonomo F, Mattia S, Oriolo G (2011) Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem. Theoret Comput Sci 45:6261–6268

Carello G, Lanzarone E, Mattia S (2018) Trade-off between stakeholders' goals in the home care nurse-to-patient assignment problem. Op Res Health Care 16:29–40

Centro Nazionale Sangue (2020) Globuli rossi: monitoraggio dati SISTRA produzione e consumo. https://centronazionalesangue.it/wp-content/uploads/2020/08/Monitoraggio-mensile-GR-per-sito_aprile-2020.pdf. Accessed Dec 2020

Doneda M, Yalçındağ S, Lanzarone E (2023) A three-stage matheuristic for home blood donation appointment reservation and collection routing. Flex Serv Manuf J. https://doi.org/10.1007/s10696-023-09518-6

Doneda M, Yalçındağ S, Marques I et al (2021) A discrete-event simulation model for analyzing and improving operations in a blood donation center. Vox Sang 116:1060–1075

Ford L, Fulkerson D (1956) Maximal flow through a network. Can J Math 8:399–404

Govindan K, Mina H, Alavi B (2020) A decision support system for demand management in healthcare supply chains considering the epidemic outbreaks: a case study of coronavirus disease 2019 (covid-19). Transp Res Part E Log Transp Rev 138:101967

Gülizahinyazan F, Kara B, Taner MR (2015) Selective vehicle routing for a mobile blood donation system. Eur J Oper Res 245:22–34

Gunpinar S, Centeno G (2016) An integer programming approach to the bloodmobile routing problem. Transp Res Part E Log Transp Rev 86:94–115

Guo C, Bodur M, Aleman D et al (2020) Logic-based benders decomposition and binary decision diagram based approaches for stochastic distributed operating room scheduling. INFORMS J Comput 33(4):1259–1684

Gupta A, Ojha S, Poojary M et al (2020) Organization of the outdoor blood donation drives amid novel coronavirus pandemic and national lockdown: an experience from a tertiary care oncology institution in India. Transfus Apheresis Sci 59(5):102878

Hooker JN (2019) Logic-based benders decomposition for large-scale optimization. arxiv:1910.11944 pdf, accessed Jun 2023

Lanzarone E, Matta A (2012) The nurse-to-patient assignment problem in Home Care services. In: Tànfani E, Testi A (eds) Advanced decision making methods applied to health care. International series in operations research & management science, vol 173. Springer, Milan, pp 121–139

Lanzarone E, Yalçındağ S (2020) Uncertainty in the blood donation appointment scheduling: key factors and research perspectives. In: Bélanger V, Lahrichi N, Lanzarone E, Yalçındağ S (eds) Health care systems engineering. ICHCSE 2019. Springer proceedings in mathematics & statistics, vol 316. Springer, Cham, pp 293–304

Lanzarone E, Matta A, Sahin E (2012) Operations management applied to home care services: The problem of assigning human resources to patients. IEEE Trans Syst Man Cybern Part A Syst Hum Inst Elect Electron Eng 42:1346–1363

Mannino C, Mattia S, Sassano A (2011) Planning wireless networks by shortest path. Comput Optim Appl 48:533–551

Mattia S (2012) Separating tight metric inequalities by bilevel programming. Oper Res Lett 40(6):568–572

Mattia S (2012) Solving survivable two-layer network design problems by metric inequalities. Comput Optim Appl 51(2):809–834

Mattia S (2013) A polyhedral study of the capacity formulation of the multilayer network design problem. Networks 62(1):17–26

Mattia S, Poss M (2018) A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation. Comput Optim Appl 69:753–800

Mattia S, Rossi F, Servilio M et al (2017) Staffing and scheduling flexible call centers by two-stage robust optimization. Omega 72:25–37

Michaels J, Brennan J, Golden B et al (1993) A simulation study of donor scheduling systems for the American red cross. Comput Op Res 20:199–213

Millard D (1959) Industrial inventory models as applied to the problem of inventorying whole blood. Master's thesis, Ohio State University

Osorio A, Brailsford S, Smith H (2015) A structured review of quantitative models in the blood supply chain: a taxonomic framework for decision-making. Int J Prod Res 53:7191–7212

Padberg M, Rinaldi G (1991) A branch-and-cut algorithm for the resolution of large-scale traveling sales-man problems. SIAM Rev 33:60–100

Pagano M, Hess J, Tsang H et al (2020) Prepare to adapt: blood supply and transfusion support during the first 2 weeks of the 2019 novel coronavirus (COVID-19) pandemic affecting Washington State. Transfusion 60(5):908–911

Pierskalla W (2004) Supply chain management of blood banks. In: Brandeau ML, Sainfort F, Pierskalla WP (eds) Operations research and health care. A handbook of methods and applications, vol 5. Springer, Belin, pp 103–145

Pirabán A, Guerrero W, Labadie N (2019) Survey on blood supply chain management: models and methods. Comput Op Res 112:104756

Rahmani D (2019) Designing a robust and dynamic network for the emergency blood supply chain with the risk of disruptions. Ann Oper Res 283:613–641

Regis-Hernández F, Carello G, Lanzarone E (2020) An optimization tool to dimension innovative home health care services with devices and disposable materials. Flex Serv Manuf J 32:561–598

Şahinyazan F, Kara B, Taner M (2015) Selective vehicle routing for a mobile blood donation system. Eur J Oper Res 254:22–34

Testik M, Ozkaya B, Aksu S et al (2010) Discovering blood donor arrival patterns using data mining: A method to investigate service quality at blood centers. J Med Syst 36:579–594

Wolsey L (1998) Integer Programming. Wiley-interscience series in discrete mathematics and optimization, Wiley, London

Yalçındağ S, Baş Güre S, Carello G et al (2020) A stochastic risk-averse framework for blood donation appointment scheduling under uncertain donor arrivals. Health Care Manag Sci 23:535–555

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

Tiziano Bacci is a researcher in optimization and discrete mathematics at the National Research Council of Italy. He was an adjunct professor in operations research at University of Rome Tor Vergata. His main research interests are the development of exact and heuristic algorithms for combinatorial optimization and mixed-integer nonlinear optimization problems. The main applications are in energy, logistics, and healthcare.

Ettore Lanzarone received the Ph.D. in Bioengineering from the Politecnico di Milano, Italy, in 2008. He is an associate professor at the Department of Management, Information and Production Engineering of the University of Bergamo, Italy. He is also a research associate at the Institute for Applied Mathematics and Information Technology (IMATI) of the National Research Council (CNR), Italy, where he worked as a researcher from 2011 to 2020. He is a member of the CIRRELT (Centre Interuniversitaire de Recherche sur les Reseaux d'Entreprise, la Logistique et le Transport), Canada. His research activities focus on bioengineering, optimization and operations research, and stochastic models. Applications of the last two topics include healthcare (e.g., home care, blood donation system, ambulance tracking) and manufacturing (scheduling). He recently chaired the 48th annual meeting of the EURO working group on Operational Research Applied to Health Services (ORAHS 2022).

Sara Mattia holds a Ph.D. in Operations Research from the University of Rome La Sapienza and is currently senior researcher at the Institute for Systems Analysis and Computer Science (Istituto di Analisi dei Sistemi ed Informatica - IASI) of the National Research Council of Italy (Consiglio Nazionale delle Ricerche - CNR). S. Mattia is director of the research unit of the National Institute for High Mathematics of Italy (Istituto Nazionale di Alta Matematica - INdAM) hosted at IASI-CNR, leader of the IASI research group on Optimization, Discrete Mathematics and Applications (OPTIMA) and member of the board of the Ph.D. program in Computer Science, Control and GeoInformation of University of Rome Tor Vergata. Her main research interests are operations research, integer programming, robust optimization and bilevel programming, with application to management decision making problems arising in healthcare, network design, personnel scheduling and logistics.

Paolo Ventura is a senior researcher at the Institute of Systems Analysis and Computer Science "Antonio Ruberti" of the Italian National Research Council (CNR-IASI). His main research interests are Integer Programming and Combinatorial Optimization with applications in logistics and transportation. He is author of many articles in the most relevant international journals of the area. Since 2004, he teaches Operations Research at the University "Tor Vergata" of Rome. He is member of the organizing committee of the yearly "Cargese Workshop of Combinatorial Optimization" and, in the odd years, of the "Aussois Combinatorial Optimization Workshop".

Authors and Affiliations

Tiziano Bacci[1] · Ettore Lanzarone[2] · Sara Mattia[1] · **Paolo Ventura[1]**

✉ Sara Mattia
   sara.mattia@iasi.cnr.it

   Tiziano Bacci
   tiziano.bacci@iasi.cnr.it

   Ettore Lanzarone
   ettore.lanzarone@unibg.it

   Paolo Ventura
   paolo.ventura@iasi.cnr.it

[1] Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", Consiglio Nazionale delle Ricerche, via dei Taurini, 19, 00185 Roma, Italy

[2] Department of Management, Information and Production Engineering, University of Bergamo, viale Marconi, 5, 24044 Dalmine, BG, Italy