Composing medical crews with equity and efficiency

Published version:

DOI:10.1007/s10100-009-0093-3

R. Aringhieri.

Composing medical crews with Equity and Efficiency.

Central European Journal of Operations Research, Special Issue on "Innovative Approaches for Decision Analysis in Energy, Health, and Life Science", 17(3):343-357, 2009.

DOI: 10.1007/s10100-009-0093-3

CEJOR manuscript No.
(will be inserted by the editor)

Composing Medical Crews with Equity and Efficiency

Roberto Aringhieri

Received: date / Revised version: date

Abstract The quality of the health care is directly connected to the equity and to the efficiency of the service delivered. Usually, the health care is delivered by crews composed of individuals working together sharing knowledge, experiences and skills. We consider the problem of composing medical crews in such a way that the health care service provided follows the principles of equity and efficiency. We present a general mathematical programming model for this problem and a solution algorithm based on Tabu Search methodology. Computational analysis proves the effectiveness of the proposed algorithm.

Keywords Health service · Manpower planning · Optimization

1 Introduction

The delivery of health care is a challenging problem concerning the quality of medical services. The equity and the efficiency of the services provided are two indicators assessing the quality of the health care system. Equity concerns the access to the health care system: the access should be guaranteed to all people regardless of their age, income, residence and also citizenship. For instance, this is the case of the Italian Emergency Medical Service. Moreover, the system pursues equity when the level of the service delivered is independent on the personnel offering the service at that time. On the other side, efficiency concerns how good is delivered the health care service.

Before discussing how to apply the principles of equity and efficiency to medical crews, we first introduce an example arising at the operation center of the Emergency Medical Service of Milano (Aringhieri 2008). The operation center manages all the demands from the instant in which the operator receives a call to the time an ambulance leave the hospital after the service. The Italian law states that, for urgent requests,

R. Aringhieri
Dipartimento di Informatica, Università degli Studi di Torino, Corso Svizzera 185, 10149 Torino, Italy,
Tel.: +39 011 6706755
Fax: +39 011 751603
E-mail: roberto.aringhieri@unito.it

the response has to be performed within a mandatory time of 8 minutes in the urban areas. Urgent requests are those having red or yellow code after the triage procedure. The statistical analysis of historical data indicates that a call last in the average more than 2 minutes before the operator is able to summon an ambulance (Aringhieri et al 2008a). Here, the efficiency is directly connected to the capability of the operation center to guarantee a fast response to each call associated to an urgent request. On the other side, the equity in terms of access is related to the fact that the same efficiency should be always guaranteed independently on the time of the day.

Usually, the health care is delivered by crews composed of individuals working together sharing knowledge, experiences and skills. In this paper, we consider the problem of composing medical crews in such a way that the health care service provide by themselves satisfy the principles of equity and efficiency discussed before. The problem commonly arises from the management of health care personnel. We have already discussed the case of the operation center at the Emergency Medical Service of Milano, which is the starting point of our research. Another case is related to the composition of heart surgery crews or crews for other specific surgery. Finally, this approach is commonly used in the health care research when a new project starts creating a team of researchers having different skills and knowledge, and – hopefully – high research efficiency.

To the best of our knowledge, this problem is not already considered in literature. The manpower planning literature is focused on the long-term supply of employees in the company adapted to the forecasted needs by recruitments, layoffs or retraining the current workforce (Feyter 2006, 2007). On the other side, huge research is devoted to short-term tactical planning level of the organization, i.e., the assignment of the available workforce to the different tasks that should be performed by the company or the classic rostering problem (Jiang et al 2004; Ernst et al 2004a; Cheang et al 2003; Burke et al 2004; Ernst et al 2004b; Kellogg and Walczak 2007). Integrated approach are also studied (Li et al 2007).

The paper is organized as follows. In Section 2 we propose a general mathematical programming model for our problem discussing also the possible extensions of this formulation. Moreover, we also present a graph formulation in order to prove its $\mathcal{NP}$-hardness. In Section 3, we described the Tabu Search algorithm developed to solve the problem. Section 4 reports the computational results obtained by solving a set of benchmark instances randomly generated: the analysis reported shows the capability of the algorithm to gain a substantial average improvement with respect to basic Local Search and Tabu Search algorithms. Moreover, we validate the quality of the solution provided by the algorithm through a comparison with a standard linear programming bound. Finally, Section 5 closes the paper.

2 A general mathematical model

We can formulate our problem as follows. Let P be the number of individuals ($p = 1, \ldots, P$) available in order to compose T crews ($t = 1, \ldots, T$). Each crew should have exactly M_t individuals in such a way that $\sum_{t=1}^{T} M_t \leq P$.

In order to model the efficiency, we introduce the values $e_p \in \mathbb{R}_+$ with $p = 1, \ldots, P$: it evaluates the capability of the individual p to do its job efficiently. For instance, e_p can represent the average time required to accomplish its task. In general, it measures the effectiveness of the service provided with respect to a given parameter of evaluation, i.e., accuracy, fast answer, and so on. The simplest way to measure the efficiency of a crew is the sum of all the individual efficiency. More accurate methods are discussed in (Aringhieri 2008).

The equity principle is related to the idea that the efficiency of the service delivered should be independent on the personnel offering the service. This means that each crew should have the skills to deal with any kind of problem encountered during its work. Therefore, a crew with heterogeneous skills is better than the one having similar skills, from an equity point of view. To model this fact, for each pair p, q with $p, q = 1, \ldots, P$, we introduce the *diversity measure* d_{pq} such that $d_{pq} = d_{qp} \geq 0$ and $d_{pp} = 0$. The value d_{pq} models how much the skills of the individuals p and q are heterogeneous. Let D be the diversity threshold required for each crew $t = 1, \ldots, T$.

We now introduce the binary variable x_{pt}, with $p = 1, \ldots, P$ and $t = 1, \ldots, T$, which is equal to 1 if the individual p is assigned to crew t, 0 otherwise. A general mathematical model for our problem is the following:

$$\mathcal{P} : \max \quad \min_{t=1,\ldots,T} \sum_{p=1}^{P} e_p x_{pt} \tag{1}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} x_{pt} \leq 1\,, \qquad p = 1, \ldots, P \tag{2}$$

$$\sum_{p=1}^{P} x_{pt} = M_t\,, \qquad t = 1, \ldots, T \tag{3}$$

$$\sum_{p=1}^{P-1} \sum_{q=p+1}^{P} d_{pq} x_{pt} x_{qt} \geq D\,, \qquad t = 1, \ldots, T \tag{4}$$

$$x_{pt} \in \{0, 1\}\,, \qquad p = 1, \ldots, P\,, t = 1, \ldots, T$$

Constraints (2) and (3) assure the correct crew composition in terms of cardinality whilst constraint (4) models the fact that the skills should be enough distributed among crews. In terms of equity, the model aims at maximizing the efficiency of the crew having the minimum efficiency. The solution depicted by the model $\mathcal{P}$ selects a subset of individuals in such a way that they are composed in T crews having fixed cardinality and the skills are heterogeneously distributed within the crews.

Although $\mathcal{P}$ represents a general situation, it can be extended in order to model more accurately the rules for the crew composition. In the current formulation, the case in which two individuals p and q have the same skill is modeled by setting $d_{pq} = 0$; on the other side, a $d_{pq} > 0$ models how different are the skills of p and q. In such cases, this could be not enough to model the actual requirements for crew composition. Therefore, we need to take into account explicitly the different skills introducing the set of skills S and the new decision variable x^s_{pt} which is equal to 1 when the individual p having skill s is assigned to crew t. This detailed approach could be useful, for instance, when it is required that each crew should be composed of a given number of individuals having the same skill in order to allow a sort of turnover among them.

$\mathcal{NP}$-Hardness

In order to prove the $\mathcal{NP}$-hardness of the problem, we introduce the following graph formulation. Let $G = (V, E)$ be the following weighted undirected graph where: $V = \{1, \ldots, P\}$, $E = V \times V$, each vertex has a weight $w'_u = e_u$ and each edge has a weight $w''_{uv} = d_{uv}$. Let E_t be the set of edge in the subgraph induced by any subset of vertexes $V_t \subset V$. For any given T, a solution for $\mathcal{P}$ corresponds to finding $V_1, \ldots, V_T$ disjoint subsets of V in such a way that $|V_t| = M_t$ $(t = 1, \ldots, T)$, the value $h_t = \sum_{[u,v] \in E_t} w''_{uv}$ should be greater than or equal to D for any $t = 1, \ldots, T$, and the value $z = \min_{t=1,\ldots,T} \sum_{u \in V_t} w'$ is maximized. Since G is a complete graph, we observe that each V_t is a k-clique with $k = M_t$.

We now consider the following particular instance of this problem. First we set the vertex weights to 1, i.e., $w'_u = e_u = 1$ and $M_t = m$ for any $t = 1, \ldots, T$. This is equivalent to fix the optimal solution value $z^\star$ equal to m since for any possible solution value is $z = m$. The corresponding problem is therefore that of finding T edge weighted m-cliques in such a way that the weight of each clique should be greater than or equal to D. Setting T equal to 1, we obtain the decision version of the maximum edge subgraph problem which is known to be $\mathcal{NP}$-complete (see, e.g., (Garey and Johnson 1979; Ausiello et al 1999)). By consequence, our problem is $\mathcal{NP}$-hard.

3 The Tabu Search algorithm

The proposed algorithm is a quite standard Tabu Search (Glover and Laguna 1997) in which the initial solution, computed by a Greedy algorithm, is improved by a neighborhood search. This search is strengthened by adopting a short-term memory strategy and allowing the search to explore unfeasible solutions.

We denote a solution for our problem as a set of crews C^t having efficiency E^t and diversity D^t for any $t = 1, \ldots, T$. Moreover, the crew C^0 is composed of the individuals not selected by the algorithm, i.e., $C^0 = \{1, \ldots, P\} \setminus \{C^1 \cup \ldots \cup C^t\}$. We finally denote the diversity contribution of each individual p to a given crew C^t with the value D^t_p for any $p = 1, \ldots, P$ and $t = 1, \ldots, T$. D^t_p is set to

$$D^t_p = \sum_{q \in C_t} d_{pq}. \tag{5}$$

At the beginning of the algorithm, we have $C^0 = \{1, \ldots, P\}$, $C^1 = \ldots = C^T = \emptyset$, $E^1 = \ldots = E^T = 0$ and $D^1 = \ldots = D^T = 0$. Let z be equal to $\min\{E^1 = \ldots = E^T\}$. Finally, all the values D^t_p are set to 0.

3.1 Greedy initialization

At each iteration, the greedy algorithm considers the individuals having largest efficiency not already assigned to a crew. Among them, it builds a solution selecting the individual giving the largest improvement D^t_p to the efficiency of a crew among the ones having minimum cardinality. The pseudo-code of this procedure is given in Figure 1 in which is also highlighted the update of D^t_p values.

The assignment to a crew having minimum cardinality guarantees the feasibility of the constraints (2) and (3) while it is not guaranteed the feasibility of the constraint (4).

Fig. 1 Greedy initialization pseudo-code.

procedure `initGreedy( )`
initializationOf$(C^0; C^t; D^t; E^t; D^t_p)$;
repeat
  $E_{\max} = \{p \in C^0 : e_p = \max_{q \in C^0}\{e_q\}\}$;
  $C_{\min} = \{t = 1, \dots, T : |C^t| = \min\{|C^s| : s = 1, \dots, T\}\}$;
  select $p \in E_{\max}$ and $t \in C_{\min}$ in such a way to maximize D^t_p;
  $C^t = C^t \cup \{p\}$; $C^0 = C^0 \setminus \{p\}$; $E^t = E^t + e_p$; $D^t = D^t + D^t_p$;
  for each $q \in C^0$ **do** $D^t_q = D^t_q + d_{pq}$;
until $|C^1| = M_1$ && ... && $|C^T| = M_T$;
$z = \min\{E^1 = \dots = E^T\}$;
end-procedure

We actually update D^t_q also for $q \in C^t$, with the same formula, i.e., $D^t_q = D^t_q + d_{pq}$, since these values are required by the subsequent improvement phase. We observe that the loop is repeated $\sum_{t=1}^{T} M_t$ times which is $O(P)$. Each iteration is $O(P\,T)$ since $|E_{\max}|$ is $O(P)$ and $|C_{\min}|$ is $O(T)$. Therefore, the greedy initialization is $O(P^2\,T)$.

3.2 Improvement phase

Such a solution is then improved by the following neighbourhood search. At each iteration, the crew C^w having the worst efficiency is considered. Then, we find a pair (p, q) of individuals giving the best solution improvement in such a way that p belongs to C^w and q to any other crews. Notice that, the proposed neighbourhood maintains the feasibility of the constraints (2) and (3). The pseudo-code of this procedure is given in Figure 2 where z_{pq} denotes the value of the solution that the search could obtain exchanging p and q.

Fig. 2 Neighborhood Search pseudo-code.

procedure `neighborhoodSearch( )`
boolean improvedSolution;
repeat
  improvedSolution = FALSE; $z' = 0$;
  select w s.t. $E^w = \min\{E^t : t = 1, \dots, T\}$;
  for each $p \in C^w$ **do**
    for each $q \in \{1, \dots, P\} \setminus C^w$ **do**
      if $z_{pq} > z'$ && move (p, q) is feasible **then**
        $p' = p$; $q' = q$; $z' = z_{pq}$; $t = \text{crewOf}(q)$;
      end if
    end for
  end for
  if $z' > z$ **then**
    $z = z'$; $C^w = C^w \setminus \{p'\} \cup \{q'\}$; $C^t = C^t \setminus \{q'\} \cup \{p'\}$;
    update D^w and D^t using (6) and (7);
    update D^s_r using (8) and (9);
    improvedSolution = TRUE;
  end if
until improvedSolution;
end-procedure

In order to verify the feasibility of the constraint (4) of each move, we can use the values defined in (5). The new possible D^w is obtained by subtracting the total contribution of the old element p (that is D^w_p) and adding the total contribution of the new element t (that is $D^w_q - d_{pq}$). In a similar way, we obtain the new possible value of D^t. More formally, we have:

$$D^w = D^w - D^w_p + D^w_q - d_{pq} \tag{6}$$

and

$$D^t = D^t + D^t_p - D^t_q - d_{pq}. \tag{7}$$

The use of equations (6) and (7) makes the algorithm faster than the direct use of the diversity matrix. Actually, this computation is $O(1)$ while, without using the values D^t_p, it will be $O({M_t}^2)$. Since the composition of the crews w and t is changed, the algorithm should update also each individual contribution. With respect to the notation introduced in Figure 2, this can be done using the following equations:

$$D^w_r = D^w_r - d_{p'r} + d_{q'r}\,, \qquad r \in \{1, \ldots, P\}, \tag{8}$$

and

$$D^t_r = D^t_r + d_{p'r} - d_{q'r}\,, \qquad r \in \{1, \ldots, P\}. \tag{9}$$

Each iteration of the search is $O(P^2 + PT)$ since the double **for each** is $O(P^2)$ and the update of the values (5) is $O(PT)$.

3.3 Short-term memory and unfeasible solutions

The pseudo-code reported in Figure 2 continues its search until it finds an improving feasible solution. The Tabu Search methodology (Glover and Laguna 1997) relaxes the need of an improving solution to continue the exploration of the solution space. In order to avoid solution cycle, the Tabu Search plans to use a tabu list of already visited solutions.

In our algorithm, we introduce two different tabu lists. List L_1 forbids to select an individual moved in the last ℓ_1 iterations whilst list L_2 forbids to move an individual to its original crew before ℓ_2 iterations. Notice that it should be $\ell_1 < \ell_2$. The basic idea underlying the use of the lists L_1 and L_2 is not only to avoid cycles but also to lead the search: after a move, L_1 fixes p and q to allow the search to adjust the overall solution after their shift; then, since L_2 avoids the return to the original crew, it allows the search to make a sequence of shifts moving p (or q) from its original crew to a crew potentially more efficient when the direct exchange is not possible.

Short term memory provides the algorithm of a basic intensification and diversification strategy (Dell'Amico and Trubian 1998): after W_ℓ not improving iterations, ℓ_1 and ℓ_2 increase allowing the search to escape from a not promising region of the solution space; on the contrary, after I_ℓ improving iterations, ℓ_1 and ℓ_2 decrease allowing the search to intensify the search in a promising region. Therefore, during the search, the length of tabu lists starts from an initial value and it ranges between a minimum and a maximum values. Tabu Search requires also to introduce a termination condition: our algorithm stops its search after N_{it} iterations.

Allowing the search to visit unfeasible solutions usually improves the capability of the algorithm to explore the solution space finding better solutions (see, e.g., the algorithms for the Vehicle Routing Problem (Laporte et al 2000)). Moreover, this can improve also the performance of the algorithm (see, e.g., (Aringhieri and Dell'Amico 2005)). To deal with the infeasibility of the constraint (4), we introduced the following penalized objective function z_p,

$$z_p = z - \alpha I \tag{10}$$

where I is a measure of how much is unfeasible the solution, i.e.,

$$I = \sum_{t=1}^{T} \left(\max \left\{ 0, D - \sum_{p=1}^{P-1} \sum_{q=p+1}^{P} d_{pq} x_{pt} x_{qt} \right\} \right) ;$$

α is equal to $\dfrac{z^\star}{D}$ where $z^\star$ is the value of the best solution found during the search. We notice that α increases during the search as soon as $z^\star$ increases, i.e., the search finds new best solutions. This means that the search explores unfeasible solutions more easily at the beginning of the search when $z^\star$ has low value. On the contrary, it is less likely to explore unfeasible solutions at the end of the search since $z^\star$ – hopefully – tends to its optimal value.

4 Computational results

This section reports the computational results obtained by solving a set of benchmark instances randomly generated: the analysis reported shows the capability of the algorithm to gain a substantial average improvement with respect to basic Local Search and Tabu Search algorithm. Moreover, we validate the solution quality through a comparison with a bound based on a linear programming approach.

4.1 Setting up the computational experiments

Our algorithm is coded using the C standard 2 and runs on a Linux machine with G++ 3.4.6 compiler. The PC is an Intel Core 2 Duo T7200 2GHz with 2GB of main memory running a GNU/Linux Slackware Linux operating system (kernel 2.6.18). For our experiments, we use a set of 80 random generated instances such that:

- the number of individuals $P = \{100, 200, 300, 400, 500\}$,
- number of crews $T = \{5, 10\}$,
- number of individuals to be selected is equal to 60% or 80% of P,
- each crew has the same number of individuals M set to $\dfrac{0.6P}{T}$ or $\dfrac{0.8P}{T}$.

The diversity matrix is taken from the Silva's instances (Silva et al 2004) for the Maximum Diversity Problem (see, e.g., (Aringhieri et al 2008b)) while the efficiency values are randomly generated.

We heavily tested our algorithm in order to obtain a suitable parameters' settings, reported in Table 1. In our test, we have also considered a restricted version of the greedy algorithm depicted in Figure 1 in which the individual p is selected among all the individuals instead of the individuals having largest efficiency. On our benchmark, this allows to obtain always a feasible solution after the greedy computation.

Table 1 Parameters' settings.

ℓ_1			ℓ_2					
initial	min	max	initial	min	max	W_ℓ	I_ℓ	N_{it}
5	3	10	10	5	15	12	8	2000

4.2 Improvements analysis

In order to evaluate the improvement capability of the proposed algorithm, we introduce the following two values. The former is z_{LS} which is the result obtained by the Local Search depicted in Figure 2. The latter is related to the Tabu Search with fixed length tabu lists (i.e., without short-term memory) and without allowing the search to explore unfeasible solutions: the value z_{30} is the best value selected among 30 executions of the algorithm, each one with a different length settings. We notice that both tests use the restricted greedy algorithm to compute the initial feasible solution. Moreover, we consider three versions of the proposed algorithm:

- TS_1: our algorithm but starting with restricted greedy and allowing the search to explore only feasible solutions;
- TS_2: our algorithm but starting with restricted greedy and allowing the search to explore both feasible and unfeasible solutions;
- TS_3: our algorithm (the complete version).

In the following analysis, we do not discuss the results for each one of the 80 instances but only the aggregate results. In order to do not to leave anything out, we report all the computed results in Table 6 at the end of the paper.

First, we report the average and the maximum computing time of all the algorithms considered in the following. The computing time reported is the overall time needed by the algorithm to complete its execution, i.e., to perform all N_{it} iterations. Table 2

Table 2 Average and maximum computing time in seconds (column TS'_1 reports the results obtained setting $N_{it} = 4000$).

	z_{LS}	z_{30}	TS_1	TS'_1	TS_2	TS_3
avg.	0.15	10.99	0.38	0.68	2.76	2.64
max	0.33	35.44	1.21	2.78	8.05	7.58

shows that the proposed algorithm is really efficient from a computational point of view. Actually, the proposed algorithm, i.e., TS_3, finds a solution for the problem with an average computing time equal to 2.64 seconds and a maximum time of 7.58 seconds for an instance with $P = 500$, $T = 5$ and $M = 80$. We observe that the highest time is obviously obtained by the 30 repeated basic Tabu Search to compute the value z_{30}.

The following analysis is devoted to understand the capability of our algorithm, TS_3, to improve the quality of the solution performing a better exploration of the solution space. Table 3 reports the comparisons of TS_x algorithms with respect to the values z_{LS} and z_{30}. Our algorithm TS_3 shows the highest average improvements with respect to z_{LS} which is equal to 30.75%. Moreover, the number of instances improved

Table 3 Gaps (row TS_1' reports the results obtained setting $N_{it} = 4000$).

	gaps					
	w.r.t. z_{LS}			w.r.t. z_{30}		
	avg.	min	max	avg.	min	max
TS_1	26.71%	0.00%	65.19%	−0.47%	−13.69%	2.70%
TS_1'	27.37%	0.00%	65.19%	0.3%	−13.69%	3.72%
TS_2	12.64%	−10.97%	51.17%	−11.27%	−45.27%	0.00%
TS_3	30.75%	9.87%	75.28%	3.02%	−4.85%	75.28%

on at least the 30% are 34 over 80. With respect to the value z_{30}, TS_3 improves it on the 3.02%. Among the three TS_x algorithms, the worst improvement is given by TS_2. This is due to the fact that the algorithm is not able to improve z just allowing the search over unfeasible solutions. Furthermore, although it starts its search from the same initial solution, TS_1 obtains better average results than TS_2. The algorithm derives a substantial benefit from allowing the search over unfeasible solutions when also the initial solution is unfeasible, as shown in Table 4. The reported analysis proves

Table 4 Gaps: TS_3 vs. TS_1 and TS_2 (row TS_1' reports the results obtained setting $N_{it} = 4000$).

	gaps		
	avg.	min	max
TS_1	3.56%	−5.66%	75.28%
TS_1'	3.06%	−5.86%	75.28%
TS_2	16.80%	−2.65%	82.23%

the capability of the proposed algorithm to find good quality solutions with respect to basic algorithms. Unfortunately, we are not aware about how good are these solutions. To understand this fact, we provide a comparison with the bound discussed in the following.

4.3 Comparison with LP bound

The linear programming bound is based on the following reformulation of the original mathematical formulation (1)–(4). We introduce the binary variable y_{pqt}, with $p, q = 1, \ldots, P$ and $t = 1, \ldots, T$: it is equal to 1 if $x_{pt} = x_{qt} = 1$, 0 otherwise. We reformulate the constraint (4) adopting the following standard linearization:

$$y_{pqt} \leq x_{pt}\,, \qquad p = 1, \ldots, P,\ t = 1, \ldots, T \tag{11}$$

$$y_{pqt} \leq x_{qt}\,, \qquad q = 1, \ldots, P,\ t = 1, \ldots, T \tag{12}$$

$$x_{pt} + x_{qt} \leq y_{pqt} + 1\,, \qquad p, q = 1, \ldots, P,\ t = 1, \ldots, T \tag{13}$$

$$y_{pqt} = y_{qpt}\,, \qquad p, q = 1, \ldots, P,\ t = 1, \ldots, T. \tag{14}$$

Therefore, we consider the following linearized formulation of the original problem $\mathcal{P}$:

$$\begin{aligned} \mathcal{P}_L : \max \quad & \min_{t=1,\ldots,T} \sum_{p=1}^{P} e_p x_{pt} && (15) \\ \text{s.t.} \quad & (2),\, (3),\, (11) - (14) \\ & \sum_{p=1}^{P-1} \sum_{q=p+1}^{P} d_{pq} y_{pqt} \geq D\,, \qquad t = 1,\ldots,T && (16) \\ & x_{pt}\,, y_{pqt} \in \{0,1\} \qquad p,q = 1,\ldots,P\,, t = 1,\ldots,T \end{aligned}$$

We recall that each crew, in the graph formulation, is a clique with M_t vertexes. We exploit this fact to strengthen the above formulation adding the following two constraints. The former, depicted in (17), states that if individual p belongs to the crew t then the crew t will contain exactly other $M_t - 1$ individuals, i.e.,

$$\sum_{q=1, q \neq p}^{P} y_{pqt} = (M_t - 1) x_{pt}\,, \qquad p = 1,\ldots,P\,, t = 1,\ldots,T \tag{17}$$

while the former, depicted in (18), states that each crew should contain exactly $(M_t(M_t - 1))/2$ individuals, i.e.,

$$\sum_{p=1}^{P-1} \sum_{q=p+1}^{P} y_{pqt} = \frac{M_t(M_t - 1)}{2}\,, \qquad t = 1,\ldots,T. \tag{18}$$

Finally, the bound considered in the following is given by the solution of the linear relaxation of the model $\mathcal{P}_L$ strengthened with (17) and (18).

To compute the bounds we used Cplex 8.1 with standard settings. We tested only the instances having P equal to 100 since they are the only instances solved within the time limit of 24 hours. Table 5 reports the results computed by Cplex. The average bound is 6.73% and it attests the quality of the solution computed by our algorithm. Moreover, we observe that the average bound of TS_1 and TS_2 is respectively 19.82 and 33.02. We notice that the bound can be strengthened adopting more refined linearization techniques even if this study is out of the scope of the paper.

5 Conclusions

In this paper, we have introduced the problem of composing medical crews in such a way that the health care service provided by themselves follows the principle of equity and efficiency. To the best of our knowledge, this problem has never been discussed in literature.

We have proposed two mathematical formulations of the problem. The first one is a general mathematical program in which the principles of equity and efficiency are introduced. We discuss also possible extension of the model in order to take into account more accurate rules for composing crews. The second one is a graph model which allow us to prove the $\mathcal{NP}$-hardness of our problem. We have also developed a Tabu Search algorithm to compute a solution of the problem. The proposed algorithm has been proved to be efficient from a computational point of view. Moreover, a comparison with a linear programming bound shows the quality of the solution computed.

Table 5 Comparing TS_3 with LP bound

	bound	secs.	TS_3	secs.	gaps
01-P100T10M6	449.70	18189.21	399	0.16	12.71%
01-P100T10M8	508.30	17174.76	458	0.23	10.98%
01-P100T5M12	899.40	707.12	871	0.18	3.26%
01-P100T5M16	1016.60	10459.12	999	0.25	1.76%
02-P100T10M6	449.70	9817.35	398	0.16	12.99%
02-P100T10M8	508.30	16032.57	469	0.22	8.38%
02-P100T5M12	899.40	3237.71	889	0.19	1.17%
02-P100T5M16	1016.60	7883.09	1008	0.24	0.85%
03-P100T10M6	449.70	14078.59	397	0.17	13.27%
03-P100T10M8	508.30	25855.01	475	0.22	7.01%
03-P100T5M12	899.40	1274.91	867	0.18	3.74%
03-P100T5M16	1016.60	8520.69	1002	0.24	1.46%
04-P100T10M6	449.70	13939.99	393	0.15	14.43%
04-P100T10M8	508.30	22554.53	467	0.22	8.84%
04-P100T5M12	899.40	4380.95	863	0.17	4.22%
04-P100T5M16	1016.60	10795.57	990	0.24	2.69%
				average gap	6.73%

Ongoing research is mainly interested in the study of more accurate methods to assess the efficiency of a crew. Although it is a valid measure, the objective function described in (1) does not take into account how individuals collaborate in order to accomplish their work. For instance, this can be done introducing properly a stochastic process in the optimization model, as discussed in (Aringhieri 2008), or developing a combined simulation and optimization approach (see, e.g., (Fu 2002)). All these approaches will be – hopefully – tested and applied to a real case study.

Acknowledgements The authors wish to thank the Emergency Service Management of Milano for fruitful discussions. Besides, the author wish to thank the student F. Sangiovanni for its help for the numerical results.

References

Aringhieri R (2008) Models for the efficient team planning at emergency medical service of milano. In: Xie X, Lorca F, Marcon E (eds) Operations Research for Health Care Delivery Engineering, Proceeding of the 33rd International Conference on Operational Research Applied to Health Service (ORAHS 2007), pp 281–288

Aringhieri R, Dell'Amico M (2005) Comparing Metaheuristic Algorithms for the SONET Network Design Problems. J Heuristics 11(1):35–57

Aringhieri R, Carello G, Morale D (2008a) A simulation based tool for ambulances management evaluation. In: Xie X, Lorca F, Marcon E (eds) Operations Research for Health Care Delivery Engineering, Proceeding of the 33rd International Conference on Operational Research Applied to Health Service (ORAHS 2007), pp 379–392

Aringhieri R, Cordone R, Melzani Y (2008b) Tabu Search vs. GRASP for the Maximum Diversity Problem. 4OR 6(1):45–60

Ausiello G, Crescenzi P, Gambosi G, Kann V, Marchetti-Spaccamela A, Protasi M (1999) Complexity and Approximation. Springer Verlag

Burke EK, Causmaecker PD, Berghe GV, Landeghem HV (2004) The state of the art of nurse rostering. J Sched 7:441–499

Cheang B, Li H, Lim A, Rodrigues B (2003) Nurse rostering problems – a bibliographic survey. Eur J Oper Res 151:447–460

Dell'Amico M, Trubian M (1998) Solution of large weighted equicut problems. Eur J Oper Res 106:500–521

Ernst A, Jiang H, Krishnamoorthy M, Owens B, Sier D (2004a) An Annotated Bibliography of Personnel Scheduling and Rostering. Ann Oper Res 127:21–144

Ernst A, Jiang H, Krishnamoorthy M, Sier D (2004b) Staff scheduling and rostering: A review of applications, methods and models. Eur J Oper Res 153:3–27

Feyter TD (2006) Modeling heterogeneity in manpower planning: dividing the personnel system into more homogeneous subgroups. Appl Stoch Model Bus 22:321–334

Feyter TD (2007) Modeling mixed push and pull promotion flows in Manpower Planning. Ann Oper Res 155:25–39

Fu MC (2002) Optimization for simulation: Theory vs. practice. INFORMS J Comput 14(3):192–215

Garey M, Johnson D (1979) Computers and Intractability. A guide to the Theory of NP-completeness. V.H. Freeman and Company, New York

Glover F, Laguna M (1997) Tabu Search. Kluwer Academic Publishers, Boston

Jiang H, Krishnamoorthy M, Sier D (eds) (2004) Staff Scheduling and Rostering: Theory and Applications, Part I and II, Ann Oper Res, vol 127

Kellogg DL, Walczak S (2007) Nurse scheduling: From academia to implementation or not? Interfaces 37(4):355–369

Laporte G, Gendreau M, Potvin J, Semet F (2000) Classical and modern heuristics for the vehicle routing problem. Int T Oper Res 7:285–300

Li Y, Chen J, Cai X (2007) An integrated staff-sizing approach considering feasibility of scheduling decision. Ann Oper Res 155:261–390

Silva GC, Ochi LS, Martins SL (2004) Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. In: Proceedings of the 3rd International Workshop on Efficient and Experimental Algorithms (WEA 2004), Lectures Notes on Computer Science (LNCS), vol 3059, Springer–Verlag, pp 498–512

Table 6 Best solution value and computing time for each instance.

	z_{LS}		z_{30}		TS_1		TS_2		TS_3	
	$z^\star$	secs.	$z^\star$	secs.	$z^\star$	secs.	$z^\star$	secs.	$z^\star$	secs.
01-P100T10M6	297	0.01	297	0.59	297	0.02	297	0.18	399	0.16
01-P100T10M8	360	0.01	360	0.61	360	0.02	360	0.23	458	0.23
01-P100T5M12	602	0.01	810	1.18	803	0.04	690	0.20	871	0.18
01-P100T5M16	830	0.01	1004	2.02	1011	0.06	968	0.25	999	0.25
02-P100T10M6	326	0.01	415	0.77	412	0.03	326	0.16	398	0.16
02-P100T10M8	375	0.01	488	1.02	487	0.02	384	0.23	469	0.22
02-P100T5M12	700	0.01	840	1.39	725	0.03	725	0.18	889	0.19
02-P100T5M16	851	0.01	1006	1.88	1005	0.06	992	0.24	1008	0.24
03-P100T10M6	306	0.02	372	0.85	372	0.03	320	0.15	397	0.17
03-P100T10M8	271	0.01	271	0.96	271	0.03	266	0.18	475	0.22
03-P100T5M12	661	0.01	843	1.43	816	0.05	600	0.18	867	0.18
03-P100T5M16	912	0.01	988	1.80	998	0.07	861	0.23	1002	0.24
04-P100T10M6	333	0.02	409	0.82	398	0.03	333	0.16	393	0.15
04-P100T10M8	299	0.01	490	1.14	447	0.03	452	0.23	467	0.22
04-P100T5M12	661	0.02	830	1.43	789	0.05	658	0.17	863	0.17
04-P100T5M16	849	0.01	988	1.76	908	0.06	849	0.24	990	0.24
05-P200T10M12	501	0.05	779	3.51	800	0.13	652	0.66	781	0.61
05-P200T10M16	719	0.05	925	4.50	938	0.15	854	0.86	903	0.83
05-P200T5M24	1211	0.06	1524	5.50	1549	0.18	1216	0.67	1645	0.69
05-P200T5M32	1509	0.06	1868	7.39	1872	0.25	1663	1.00	1870	0.94
06-P200T10M12	630	0.07	788	3.04	797	0.12	574	0.72	781	0.62
06-P200T10M16	711	0.06	924	3.84	931	0.15	807	0.96	905	0.87
06-P200T5M24	1245	0.06	1598	5.65	1594	0.20	1379	0.72	1626	0.67
06-P200T5M32	1574	0.06	1874	7.30	1876	0.24	1791	1.04	1864	0.94
07-P200T10M12	567	0.06	782	2.69	788	0.12	567	0.68	749	0.61
07-P200T10M16	764	0.06	928	3.67	936	0.15	907	0.88	883	0.91
07-P200T5M24	1255	0.06	1591	5.38	1597	0.20	1380	0.71	1623	0.70
07-P200T5M32	1581	0.06	1881	7.40	1884	0.23	1785	0.95	1860	0.93
08-P200T10M12	474	0.06	771	2.49	783	0.09	422	0.63	769	0.61
08-P200T10M16	750	0.06	928	3.49	923	0.15	833	0.86	887	0.88
08-P200T5M24	1221	0.06	1608	4.50	1598	0.16	1277	0.71	1641	0.69
08-P200T5M32	1584	0.06	1882	6.27	1883	0.24	1786	0.97	1871	0.94
09-P300T10M18	865	0.14	1228	5.72	1218	0.20	988	1.40	1245	1.38
09-P300T10M24	1143	0.14	1465	7.77	1468	0.27	1295	2.18	1440	2.06
09-P300T5M36	1790	0.13	2380	9.56	2363	0.33	2115	1.64	2551	1.60
09-P300T5M48	2431	0.14	2929	12.76	2926	0.43	2744	2.70	2932	2.12
10-P300T10M18	865	0.14	1189	5.52	1182	0.20	1034	1.78	1233	1.39
10-P300T10M24	1132	0.12	1432	7.02	1434	0.24	1308	2.61	1433	2.26
10-P300T5M36	1859	0.13	2449	9.40	2445	0.32	2136	1.95	2559	1.87
10-P300T5M48	2353	0.12	2940	12.85	2947	0.43	2824	2.70	2938	2.63
11-P300T10M18	942	0.14	1214	5.60	1219	0.19	1077	1.73	1265	1.78
11-P300T10M24	1207	0.13	1466	7.33	1471	0.26	1390	2.38	1455	2.32
11-P300T5M36	1945	0.14	2522	9.94	2526	0.34	2341	2.00	2556	1.93
11-P300T5M48	2332	0.13	2957	13.73	2957	0.48	2868	2.72	2937	2.67
12-P300T10M18	857	0.12	1210	5.54	1241	0.19	1080	1.77	1242	1.68
12-P300T10M24	1261	0.14	1466	7.22	1469	0.24	1351	2.50	1438	2.32
12-P300T5M36	1842	0.14	2425	9.34	2417	0.32	2140	2.00	2542	1.90
12-P300T5M48	2407	0.14	2950	12.89	2955	0.44	2790	2.60	2921	2.62
13-P400T10M24	1193	0.24	1567	9.59	1502	0.32	1218	3.36	1707	3.09
13-P400T10M32	1562	0.26	1993	12.98	1998	0.45	1894	4.28	1979	5.02
13-P400T5M48	2360	0.19	3344	16.65	3291	0.57	2995	3.65	3471	3.28
13-P400T5M64	3499	0.23	4004	22.27	3997	0.75	3797	4.94	4000	4.81
14-P400T10M24	1141	0.21	1653	10.19	1669	0.35	1454	2.91	1697	2.93
14-P400T10M32	1558	0.21	1992	13.10	2000	0.45	1879	4.26	1961	4.40
14-P400T5M48	2498	0.20	3211	16.17	3132	0.54	2643	3.70	3466	3.52
14-P400T5M64	3430	0.20	4008	22.78	4004	0.76	3788	5.19	3998	4.81
15-P400T10M24	1210	0.19	1652	9.88	1655	0.34	1432	3.13	1702	3.02
15-P400T10M32	1645	0.20	1998	13.17	1999	0.45	1878	4.38	1974	4.24
15-P400T5M48	2425	0.19	3341	16.64	3347	0.57	2898	3.93	3473	3.48
15-P400T5M64	3358	0.20	4018	22.74	4018	0.77	3883	4.82	4001	4.71
16-P400T10M24	1325	0.20	1620	9.78	1633	0.34	1453	3.19	1685	2.94
16-P400T10M32	1661	0.21	1981	12.73	1989	0.43	1898	4.79	1952	4.36
16-P400T5M48	2540	0.20	3355	16.89	3352	0.56	2981	3.36	3470	3.66
16-P400T5M64	3330	0.19	4023	23.43	4024	0.78	3832	5.10	3993	4.62
17-P500T10M30	1518	0.32	2010	16.07	1995	0.55	1801	5.22	2145	5.39
17-P500T10M40	2023	0.33	2466	20.58	2473	0.70	2202	7.20	2483	6.67
17-P500T5M60	3209	0.31	4001	25.79	3928	0.87	3601	5.73	4332	5.64
17-P500T5M80	4231	0.32	4946	34.35	4877	1.16	4428	7.83	4989	7.58
18-P500T10M30	1413	0.30	2016	15.97	2013	0.55	1746	5.32	2132	5.03
18-P500T10M40	2012	0.32	2474	20.71	2486	0.70	2226	7.08	2461	6.75
18-P500T5M60	2876	0.30	4015	25.53	4020	0.88	3399	5.60	4354	5.64
18-P500T5M80	3956	0.30	4972	34.93	4976	1.18	4771	7.86	5000	7.54
19-P500T10M30	1332	0.32	2021	16.12	2026	0.54	1699	5.22	2133	5.44
19-P500T10M40	2015	0.33	2469	20.25	2467	0.69	2260	7.87	2470	6.95
19-P500T5M60	3185	0.31	4126	26.34	4113	0.90	3600	5.80	4335	5.58
19-P500T5M80	4243	0.32	4966	35.43	4970	1.21	4776	7.48	4987	7.58
20-P500T10M30	1428	0.31	2122	16.54	2131	0.56	1922	5.16	2114	4.96
20-P500T10M40	1976	0.33	2503	21.61	2504	0.73	2378	7.51	2446	6.56
20-P500T5M60	3364	0.32	4109	25.97	4137	0.88	3581	6.06	4346	5.51
20-P500T5M80	4214	0.32	4992	35.44	4972	1.21	4767	8.05	4995	7.40