J Comb Optim (2015) 29:750–780
DOI 10.1007/s10878-013-9620-1

Elevator dispatching problem: a mixed integer linear programming formulation and polyhedral results

Mirko Ruokokoski · Harri Ehtamo · Panos M. Pardalos

Published online: 8 May 2013

Abstract In the static elevator dispatching problem the aim is to design a route for each capacitated elevator to satisfy a set of transportation requests such that a cost function is minimized while satisfying a number of constraints. This problem is a crucial part in the control of an elevator group. So far, the problem has been formulated in various algorithmic-dependent forms, where part of the constraints have been given only verbally. In this paper we present a mixed-integer linear programming formulation of the problem where all constraints are given in explicit mathematical form. This allows, e.g., polyhedral analysis of the problem. We also present some new valid inequalities to strengthen the formulation. Furthermore, we study the polyhedral structure of the problem in a generic case arising in the down-peak traffic pattern. In particular, we show which equalities define a minimal equality system for the polytope of the problem, which is defined as the convex hull of the feasible solutions. In addition, we provide the dimension of the polytope and analyze which valid inequalities derived are facet inducing.

Keywords Elevator dispatching problem · Routing · Polyhedral results · Valid inequalities

Mathematics Subject Classification (2000) 90B06 · 90C11 · 90C27 · 90C57

M. Ruokokoski (✉) · H. Ehtamo
School of Science, Aalto University, P.O. Box 11100, 00076 Aalto, Finland
e-mail: mirko.ruokokoski@aalto.fi

H. Ehtamo
e-mail: harri.ehtamo@aalto.fi

P. M. Pardalos
University of Florida, 401 Weil Hall, P.O. Box 116595, Gainesville, FL 32611-6595, USA
e-mail: pardalos@ise.ufl.edu

1 Introduction

The purpose of this paper is to present a mixed integer linear programming formulation for the static elevator dispatching problem (EDP) and analyze the polyhedral structure of it in a generic case arising in the down-peak traffic pattern. The problem studied here can be formally described as follows. Given a group of capacitated elevators—one single-deck elevator per each elevator shaft—and a set of transportation requests, each involving a time-invariant demand as well as a known pickup and delivery vertex, the objective is to design a minimum-cost set of elevator routes accommodating all the requests. In addition, a set of constraints must be satisfied. For each elevator, the route must start at an origin depot vertex of it and end at a common terminal depot vertex, the load of it must not exceed the capacity at any vertex, and its direction of travel is allowed to change only when the elevator car is empty. The last requirement forbids movements, which are called *reversals*, in which at least one passenger travels in the opposite direction with respect to his destination floor after boarding the elevator. For each request, the pickup vertex must precede the delivery vertex, and both vertices must be visited by the same elevator. Requests, except fixed ones, can be served by any elevator. A fixed request must be served by the elevator for which it has been assigned to. For each elevator stop, first the leaving passengers exit the elevator on a last-in first-out basis, and then the entering ones board it in the ascending order of their arrival times, up to the capacity limit. Finally, the EDP may also comprise restrictions on the times at which each pickup and delivery vertex is visited by an elevator. The problem described above is an essential part in the control of an elevator group.

An elevator group in a building forms a vertical transportation system designed to transport passengers from their arrival floors to their destination floors safely, comfortably, and efficiently. In the basic form it consists of several capacitated single-deck elevators, one per each shaft, that reside physically close to each other. An elevator group responds to a common set up and down call buttons that are located in the vicinity of the elevators at each floor. A passenger gives a call by pressing either up or down button at the arrival floor. When an elevator arrives, the passenger boards the car and gives his destination floor by pressing a car call button. The EDP deals with the allocation of calls to elevators and planning the service order of them under the set of constraints. By its nature, the problem is a dynamic problem since requests are gradually received throughout time. It is stochastic as well since each passenger can be described by three random variables: time of arrival, arrival floor, and destination floor.

Several different variants of the EDP have been studied. In destination control systems, conventional up and down buttons in elevator lobbies are replaced with destination keypads. A passenger gives a *transportation request* on such a device by giving his destination floor (and possibly the number of passengers traveling along with him to the same destination floor) before entering a car (see, e.g., Schröder 1990; Koehler and Ottiger 2002, and Tanaka et al. 2005a). In some works the servicing elevator is shown to user on the request device immediately after the destination floor is given, and this assignment remains fixed once made (see, e.g., Hiller and Tuchscherer 2008). Another variant is an elevator system where elevators have two or more decks (see, e.g., Sorsa et al. 2003 and Hirasawa et al. 2008). In double-deck elevators two elevator cars are attached one on top of another so that passengers at two consecutive floors are served simultaneously. This variant is known as the multi-deck EDP. Recently, some authors have considered a system in which an elevator shaft contains several elevators that move independently from each other, although with the requirement of collision avoidance (see, e.g., Ikeda et al. 2006 and Yu et al. 2009). This variant is named the multi-car EDP.

Since the basic EDP and its all variants are very difficult to solve to optimality, also due to real-time requirement (the time to respond to a new request is limited to half a second), most of the research has focused on heuristic algorithms. Examples of the approaches used include artificial intelligence (Siikonen 1997), ant colony optimization (Liu and Liu 2007), fuzzy logic (Ho and Robertson 1994), genetic algorithms (Tyni and Ylinen 2001), neural networks (Markon et al. 2006), and local search Luh et al. (2008). To the best of our knowledge, only Levy et al. (1977); Pepyne and Cassandras (1997); Inamoto et al. (2003); Tanaka et al. (2005a, b), and Hiller (2009) consider exact algorithms.

The common component in all works on the EDP is that the problem is solved on a rolling time horizon basis: the static EDP with up-to-date information is solved repeatedly to adjust request assignments and/or elevator routes, either after a certain amount of time and/or immediately after the registeration of a new request.

Despite the fact that the EDP is well-known and old problem, none of the previous works presents a formulation for the EDP in which all the constraints are given in exact mathematical form (see, e.g., Tanaka et al. 2005a). In many papers, only a verbal description of the constraints is provided. Due to this, none of the works studies the polyhedral structure of the problem, the understanding of which may lead to efficient real-time algorithms. The aim of this paper is to fill in these gaps.

The rest of this paper is organized as follows. In Sect. 2 we present a mixed integer linear programming formulation for the static EDP, arising in elevator systems with destination keypads, and discuss how the presented formulation can further be improved. In Sect. 3 we present some new valid inequalities for the EDP that utilize specific characteristics of the problem. In Sect. 4 we study the polyhedral structure of the EDP in a generic case arising in the down-peak traffic pattern. Conclusion follows in Sect. 5.

2 Formulation of the EDP

2.1 Notation and definitions

We define the EDP on a loopless digraph $G = (V, A)$ with vertex set V and arc set $A \subset V \times V$. Let n be the number of transportation requests. We associate with each transportation request $i = 1, \dots, n$ a *pickup vertex* i and a *delivery vertex* $n + i$ corresponding to the arrival and destination floors of request i, respectively. Let P and D denote the sets of pickup and delivery vertices, respectively.

A group of identical elevators E, each with capacity Q, is available to serve the requests. Let m be the number of the elevators. Associated with each $e \in E$ is an *origin depot vertex* $2n + e$ representing the initial position of elevator e. The set of origin depot vertices is denoted by T. With all elevators is also associated a *common terminal depot vertex* 0. Each elevator ends its route at this vertex, practically meaning that after all requests assigned to it have been served, the elevator stays at its last floor. Vertex set V is thereby the union of disjoint sets P, D, T, and $\{0\}$. Define $f(i)$ to be the floor of vertex $i \in P \cup D$. Let $d(i)$ denote the direction of vertex $i \in V \setminus \{0\}$. For $i \in P \cup D$, the direction is either up or down, and for $i \in T$ the direction is either up, down, or not defined in the case the elevator is idle. For a set of objects S, $|S|$ denotes the cardinality of S whereas for a real number $r \in \mathbb{R}$, $|r|$ denotes the absolute value of r.

Suppose for now that the arc set is defined by $A = \{(i, j)|i \neq j, i, j \in P \cup D\} \cup \{(i, j)|i \in T, j \in P \cup D\} \cup \{(i, 0)|i \in P \cup D \cup T\}$. In Sect. 2.4 we define A in details. With each arc $(i, j) \in A$ is associated a *travel time* τ_{ij}. There are four cases to distinguish. (1) If $i, j \in P \cup D$ and $f(i) \neq f(j)$, travel time τ_{ij} consists of a stop time and constant door closing time at vertex i, a flight time between vertices i and j, and a constant door opening time at vertex j. The *stop time* at a vertex is proportional to the demand of the vertex. The *flight time* of an elevator from vertex i to vertex j includes the time to accelerate the elevator, to travel at nominal speed, and to decelerate to vertex j. The reader is referred to Roschier and Kaakinen (1979) for an exact formula for flight times. (2) If $i, j \in P \cup D$ and $f(i) = f(j)$, τ_{ij} just consists of the stop time at vertex i. (3) If the starting vertex of arc (i, j) is $2n + e$ for some $e \in E$ then the state of elevator e (moving at constant speed, accelerating, opening doors, etc.) is taken into account in the calculation of τ_{ij}. (4) If the ending vertex of (i, j) is the common depot vertex 0 then $\tau_{ij} = 0$. Because of the physical structure of the problem, the triangle inequality holds for travel times.

Define R as a subset of arcs A such that if an elevator traverses along an arc belonging to this subset, then the elevator changes its direction of travel either at the starting vertex or ending vertex of the arc. Note that this set also includes all arcs whose ending vertex is 0.

Transportation requests can be divided into three disjoint groups. An *on-board request* is a request whose arrival floor has been visited but the delivery has not, i.e., passenger(s) of the request is inside the elevator. An *assigned request* is a request that has an assignment to some elevator and the request cannot be reassigned. In addition, the arrival floor of the request has not been visited, i.e., the passenger is still waiting for the elevator at the arrival floor. A *non-assigned request* is a request that must be (re)assigned to some elevator and both arrival and destination floors are unvisited. We call on-board and assigned requests the *fixed requests* as they cannot be reassigned.

If a request is fixed then the corresponding vertices are fixed as well. If vertex i is fixed, then it must be reachable from origin depot vertex $2n + \hat{e}$ where $\hat{e}$ is the elevator for which vertex i has been fixed to. Note that if a passenger(s) related to some request is on-board, then the pickup vertex of that request is omitted. To impose *fixing constraints* we define the family $\mathcal{F}$ of all vertex subsets $F \subset V$ such that there are one fixed vertex i belonging to set F, $2n + \hat{e} \notin F$, $2n + e \in F \forall e \in E \setminus \{\hat{e}\}$, where $\hat{e}$ is the elevator for which vertex i has been fixed to. To impose precedence constraints (for each request, the pickup vertex must precede the delivery vertex, and both vertices must be visited by the same elevator), it is convenient to define the family

$\mathcal{S}$ of all vertex subsets $S \subset V$ such that $T \subset S, 0 \notin S$, and there is a unique request i for which $i \notin S$ and $n+i \in S$.

We associate with each vertex $i \in V$ a load ω_i satisfying $\omega_0 = 0, \omega_{2n+e} \geq 0$ ($e \in E$), and $\omega_i = -\omega_{n+i} (i = 1, \ldots, n)$. Hence, if $i \in P$ ($i \in D$), constant ω_i defines the number of passengers who are entering (exiting) the elevator at vertex i, whereas if $i \in T$, constant ω_i defines the initial load of the elevator related to vertex i. Denote $\Omega(S) = \sum_{i \in S} \omega_i$, where $S \subseteq V$. We assume that for each vertex $i \in V$ it holds that $\omega_i \leq Q$. Notice that in some vehicle routing problems splitting loads can reduce the cost value significantly compared to the non-splitting case, see e.g., Archetti et al. (2008). Here we, however, assume that this option is not allowed.

Let W and J denote the maximum waiting and the maximum journey time of passengers, respectively. The *waiting time* of a passenger is the time from the instant that the passenger registers the request, until the instant the passenger enters the elevator. The *journey time* of a passenger is the time from the instant the passenger registers the request, until the instant the passenger exits the elevator at the destination floor. Given with each vertex $i \in P$ is an *elapsed time* $\gamma_i \geq 0$, the time from the moment when request i was given to the current moment. We assume that requests are indexed in the ascending order of their arrival times, therefore for all request pairs i, j, if $i < j$ then $\gamma_i \geq \gamma_j$.

With each vertex $i \in V$ is also associated a time window $[a_i, b_i]$ where a_i and b_i are the earliest and latest time at which the service may begin at vertex i, respectively. For pickup vertex i, $b_i = W - \gamma_i$ whereas for delivery vertex $n+i$, $b_{n+i} = J - \gamma_i$. If a request device is not in the vicinity of the elevator group, for each request i the walking time from the device to the elevator group is taken into account in the lower bound a_i. In other cases a_i is set to 0. Observe that, in contrast to other routing problems such as the Dial-a-Ride Problem, in the EDP the time windows are defined by the system. A time window $[0, \infty)$ is called relaxed. We assume that the time windows of origin depot and terminal depot vertices are relaxed.

We next list three assumptions which are usually given, see e.g., Tanaka et al. (2005a), and which more or less reflect practical situations in elevator systems: (1) Waiting passengers at a floor cannot enter the servicing elevator car before all on-board passengers who are going to leave the elevator at that floor finish leaving the elevator. (2) If there is more than one passenger boarding the elevator at a floor, then they board the elevator in the ascending order of their arrival times. (3) If there is more than one passenger leaving the car at a floor, then they leave the elevator in the reverse order of their boarding.

In order to employ these assumptions, which we call the *service order constraints*, we define a subset of arcs O such that if an arc belonging to this set is used in a feasible solution, then at least one of the three assumptions is not satisfied. Formally O is defined as follows. (1) $i \in P, n+j \in D, f(i) = f(n+j)$. If $d(i) = d(n+j)$, then $(i, n+j) \in O$. (2) $i, j \in P, f(i) = f(j)$. If $d(i) = d(j)$ and $j < i$, then $(i, j) \in O$. (3) $n+i, n+j \in D, f(n+i) = f(n+j)$. Suppose that the direction of request i is upwards. If $d(n+i) = d(n+j)$ and $f(i) < f(j)$, then $(n+i, n+j) \in O$. Also, if $d(n+i) = d(n+j)$, $f(i) = f(j)$, and $i < j$, then $(n+i, n+j) \in O$. Similar implications can be derived in the case of downward direction.

It is also usually assumed that if an elevator starts loading a passenger at a floor, it must load the other passengers at that stop who are allocated to that elevator and whose direction coincide with the current direction of that elevator. In order to take this assumption into account, we define a set of forward paths $\mathcal{X}$, each of which are of the form $X = (2n + e, k_1, \dots, k_i, k_j, \dots, k_r)$ such that $k_i, k_r \in P, d(k_i) = d(k_r), f(k_i) = f(k_r) \neq f(k_j)$ and either (1) $\Omega(\{2n + e, k_1, \dots, k_i\}) < Q$ or (2) $\Omega(\{2n + e, k_1, \dots, k_i\}) = Q$ and $k_r < k_i$.

For each arc $(i, j) \in A$, let x_{ij} be a binary decision variable equal to 1 if and only if an elevator travels directly from vertex i to j, 0 otherwise. For each vertex $i \in V$, let continuous decision variables t_i and q_i be the time at which an elevator begins service at vertex i and the load of an elevator upon leaving vertex i, respectively.

In each feasible solution (x, t, q), vector x defines the feasible route for each elevator. A *feasible route* of $e \in E$ is a nonempty ordered subset $H_e \in V$ of vertices for which the induced subgraph $G(H_e)$ is a path from origin depot vertex $2n + e$ to the common terminal depot vertex, 0, that may visit at some vertices in between such that none of the constraints is violated. Each feasible elevator route can be divided into a set of subpaths such that each subpath contains successive vertices in one direction of travel. We call such subpaths the *elevator trips*. An m-route H is the union of m feasible routes $H_1, \dots, H_m$ such that each request belongs to exactly one route. Vector x is called the *characteristic vector* of H.

Finally, we give some definitions used throughout this paper to simplify the notation. For any arc set C, we define $x(C) = \sum_{(i,j)\in C} x_{ij}$. For any two vertex sets S, U, we define $(S, U) = \{(i, j) \in A | i \in S, j \in U\}$ and write $x(S, U)$ for $x((S, U))$. We also write $x(i, S)$ for $x(\{i\}, S)$. Define $\overline{S} = V \backslash S$, where $S \subseteq V$. For any forward path X, define $A(X)$ to be the arc set of X.

2.2 Mixed-integer linear formulation

Using the notation above, the EDP can be formulated as the following mixed integer program:

$$\min \sum_{i\in P} \frac{\omega_i}{\Omega(P)}(t_i + \gamma_i), \tag{1}$$

subject to

$$x(V, i) = 1, \quad \forall i \in P \cup D, \tag{2}$$

$$x(i, V) = 1, \quad \forall i \in V \setminus \{0\}, \tag{3}$$

$$x(\overline{S}, S) \geq 1, \quad \forall S \in \mathcal{S}, \tag{4}$$

$$x(\overline{F}, F) \geq 1, \quad \forall F \in \mathcal{F}, \tag{5}$$

$$t_j \geq t_i + \tau_{ij} - \max\{0, b_i + \tau_{ij} - a_j\}(1 - x_{ij}), \quad \forall (i, j) \in A, \tag{6}$$

$$a_i \leq t_i \leq b_i, \quad \forall i \in V, \tag{7}$$

$$\sum_{(i,j)\in A(X)} x_{ij} \leq |A(X)| - 1, \quad \forall X \in \mathcal{X}, \tag{8}$$

$$q_j \geq q_i + \omega_j - \min\{Q, Q + \omega_i\}(1 - x_{ij}), \quad \forall (i, j) \in A, \tag{9}$$

$$\max\{0, \omega_i\} \leq q_i \leq \min\{Q, Q + \omega_i\}, \quad \forall i \in P \cup D, \tag{10}$$

$$q_i = \omega_i, \quad \forall i \in T, \tag{11}$$

$$q_i = 0, \quad i = 0, \tag{12}$$

$$q_i \leq (1 - x_{ij}) \min\{Q, Q + \omega_i\}, \quad \forall (i, j) \in R, \tag{13}$$

$$x_{ij} = 0, \quad \forall (i, j) \in O, \tag{14}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A. \tag{15}$$

The cost function (1) minimizes the sum of average waiting time of passengers, which is one of the most common performance indices used. Constraints (2) and (3) are in-degree and out-degree constraints, respectively, and express that each vertex must be visited exactly once. Constraints 4 are precedence constraints which stipulate that for each request i, the pickup vertex i is visited before the delivery vertex $n + i$ and both vertices are visited by the same elevator. These constraints have originally been presented by Balas et al. (1995) in the context of the Precedence-Constrained Asymmetric Traveling Salesman problem (TSP) and by Ruland and Rodin (1997) in the context of the TSP with Pickup and Delivery but they apply directly to the EDP with multiple elevators. Constraints (5) ensure that if vertex i is fixed to elevator $\hat{e}$, then there must be a path from $2n + \hat{e}$ to i.

Consistency of the time and load variables, t_i and q_i, respectively, are ensured by constraints (6) and (9). Time window and elevator capacity constraints are imposed by (7) and (10). Constraints (8), which we refer to as the *boarding constraints*, mean that at each time an elevator leaves a floor, the load of it is full or all passengers that have been waiting there, assigned to that elevator, and whose direction coincides with the current direction of the elevator, have boarded the elevator. Constraints (11) and (12) set the initial and ending loads for elevators, respectively. Constraints (13) are reversal constraints meaning that at each time when an elevator changes its direction of travel, the elevator car is empty. Constraints (14) are the service order constraints. Finally, all the routing variables are binary (15).

Let us now consider the cost function used here in more details. Assume for now that the time windows are not present. The waiting time of each passenger i consists of a constant term, $\omega_i \gamma_i / \Omega(P)$, which, while affecting the optimal value, can be dropped without affecting the optimal solution. Thus, in the cost function each request is effectively considered to be arrived at the same time, which may result in overlong waiting and journey times for some passengers. In fact, one can even construct such a passenger flow in which the waiting times for some passengers increase to infinity. In many papers this issue have been disregarded (see e.g., Luh et al. 2008). This issue can be avoided by using nonlinear cost functions, like the sum of squared waiting times (Yu et al. 2010). Here we, however, want to preserve the linearity of the model. This can be done by associating a time window with each vertex that restricts the times at which the vertex can be visited and in which the elapsed time is taken into account, i.e., for $i \in P$, $b_i = W - \gamma_i$ and for $n+i \in D$, $b_{n+i} = J - \gamma_i$. In this way the waiting and journey times of a passenger never exceeds the values of W and J, respectively. The values of W and J should be chosen very carefully; too tight bounds of time windows make the problem infeasible, and too loose bounds make the problem very difficult to solve since the upper bounds of the time windows are present in the time consistency constraints (6).

2.3 Improvements

Next we present several modeling techniques that can be used to improve the formulation (2)–(15). First we give some new notations. For any vertex set S, let $\pi(S) = \{i \in P | n+i \in S\}$, and $\sigma(S) = \{n+i \in D | i \in S\}$ denote the sets of *predecessors* and *successors* of S, respectively. Moreover, let $\Phi(S)$ denote a permutation of the vertices in S.

One way to improve the formulation is to reduce the number of decision variables. Consider first load variables $q_i, i \in V$. They are by nature "artificial" variables, and as a result, they can be eliminated by rewriting all constraints containing q variables only in x variables. This can be done by replacing (9)–(13) with *rounded capacity constraints* (which are often used in the context of the VRP (Naddef and Rinaldi 2002))

$$x(S, \overline{S}) \geq \max\left\{1, \left\lceil \frac{|\Omega(S)|}{Q} \right\rceil\right\}, \quad \forall S \subseteq V \setminus \{0\}, |S| \geq 2, \tag{16}$$

and by extending set $\mathcal{X}$, used in (8), to contain paths that are infeasible with respect to reversals. Thus, now $\mathcal{X}$ consists of also all the forward paths of the form $X = (2n+e, k_1, \ldots, k_{r-1}, k_r)$ where $(k_{r-1}, k_r) \in R$, and $\Omega(X \backslash \{k_r\}) > 0$. Such a path is clearly infeasible with respect to reversals since the load of elevator e is positive just before reversing its direction of travel either at vertex k_{r-1} or k_r.

As shown by Ropke and Cordeau (2009) in the context of the PDP, rounded capacity constraints (16) can be strengthened by taking into account the predecessors and successors of S as follows:

$$x(S, \overline{S}) \geq \max\left\{1, \left\lceil \frac{\Omega(\pi(S) \backslash S)}{Q} \right\rceil, \left\lceil \frac{-\Omega(\sigma(S) \backslash S)}{Q} \right\rceil\right\}. \tag{17}$$

Constraints (8) can be strengthened into *tournament constraints* (Ascheuer et al. 2000):

$$\sum_{j=1}^{r} x_{2n+e,k_j} + \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} x_{k_i,k_j} \leq r - 1. \tag{18}$$

Since for any infeasible path $X \in \mathcal{X}$, $X = (2n+e, k_1, \ldots, k_{r-1}, k_r)$, all the paths of the form $(2n+e, \Phi(U), k_r)$ are infeasible where $U = \{k_1, \ldots, k_{r-1}\}$, the tournament constraints (18) can be further strengthened into the following form

$$x(2n+e, U) + x(2n+e, k_r) + x(U, U) + x(U, k_r) \leq r - 1, \tag{19}$$

(Ascheuer et al. 2000).

Observe that time variables t_i cannot be eliminated since they are used in the cost function. Consider next routing variables x_{ij}, $(i, j) \in A$. Since with each arc (i, j) is associated routing variable x_{ij} and one time consistency constraint (6), an elimination of arcs that cannot belong to optimal solutions reduces the size of the problem. As the arc elimination is rather subtle task, we devote Sect. 2.4 for the detailed description of it.

Another way to improve the formulation is to disaggregate the time variables so that the connection between routing and time variables is stronger. The resulting formulation is called a *reformulation*. Following van Eijl (1995), we define the continuous decision variable t_{ij} as the time at which an elevator begins service at vertex i if $x_{ij} = 1$, and $t_{ij} = 0$ otherwise. The cost function now becomes

$$\min \sum_{i \in P} \left(\frac{w_i}{\Omega(P)} \left(\gamma_i + \sum_{(i,j) \in A} t_{ij} \right) \right), \tag{20}$$

and constraints (6) and (7) become

$$\sum_{(i,j) \in A} (t_{ij} + \tau_{ij} x_{ij}) \leq \sum_{(j,k) \in A} t_{jk} \quad \forall j \in P \cup D, \tag{21}$$

and

$$a_i x_{ij} \leq t_{ij} \leq b_i x_{ij} \quad \forall (i, j) \in A, \tag{22}$$

respectively. The connection to the original time variable t_i, $i \in V$, is as follows

$$t_i = \sum_{(i,j) \in A} t_{ij}. \tag{23}$$

The quality of this reformulation is assessed in the next theorem.

Theorem 1 *The reformulation with constraints* (2)–(5)*,* (8)–(15)*,* (21)*, and* (22) *is at least as strong as the formulation* (2)–(15).

Proof By summing constraints (22) over vertices j and substituting (23) and the equation $x(i, V) = 1$ into the resulting constraints we get constraints (7). By substituting (23) into (21) we have

$$\sum_{(i,j) \in A} (t_{ij} + \tau_{ij} x_{ij}) \leq t_j \quad \forall j \in P \cup D.$$

Since all the terms in the left-hand side are non-negative, it holds that

$$t_{ij} + \tau_{ij} x_{ij} \leq t_j \quad \forall (i, j) \in A.$$

By observing that $t_{ij} = x_{ij} t_{ij} = x_{ij} t_i$ we obtain

$$(t_i + \tau_{ij}) x_{ij} \leq t_j \quad \forall (i, j) \in A,$$

and by linearizing these constraints we get (6). □

As suggested by Desrochers et al. (1992) the width of time windows can be reduced as follows:

$$a_k = \max\{a_k, \min\{b_k, \min_{(i,k)\in A} \{a_i + \tau_{ik}\}\}\}; \tag{24}$$

$$b_k = \min\{b_k, \max\{a_k, \max_{(i,k)\in A} \{b_i + \tau_{ik}\}\}\}; \tag{25}$$

$$b_k = \min\{b_k, \max\{a_k, \max_{(k,j)\in A} \{b_j - \tau_{kj}\}\}\}. \tag{26}$$

The fourth rule presented by Desrochers et al. cannot be applied here since the cost function is the sum of the cumulative travel times, that is, it matters at which vertex an elevator arrives before the lower bound of the time window.

The third way to strengthen the formulation is to add valid inequalities into it. Valid inequalities for the EDP are described in Sect. 3.

Finally, consider an instance of the EDP in which there are a set of elevators which are in the same state, i.e., they all are empty, they reside at the same floor, their doors are in the same position, and none of them has fixed requests. We call such elevators *symmetrical*. For any feasible solution of the instance, a different solution with the same cost value can be obtained by just re-indexing the symmetrical elevators. Therefore, by eliminating this kind of symmetry the formulation can be improved.

Theorem 2 *Let Y be the set of symmetrical elevators. The symmetry caused by them is eliminated by replacing the set of origin depot vertices $2n + e$, $e \in Y$ with a single origin depot vertex, say k, and by replacing the set of out-degree equations of vertices $2n + e$, $e \in Y$ with the out-degree equation $x(k, V) = |Y|$. As a result, the number of integer feasible solutions is reduced by a factor of $|Y|!$.*

Proof When origin depot vertices $2n + e$, $e \in Y$ are replaced with a single vertex k and its out-degree equation is set to $x(k, V) = |Y|$, elevators residing at vertex k are considered as index-free, thus there is no symmetry anymore. Since elevators in Y can be re-indexed in $|Y|!$ different ways, the elimination of symmetry reduces the number of integer feasible solutions by a factor of $|Y|!$. □

2.4 Arc elimination

Above we defined the formulation, (1)–(15), on graph $G = (V, A)$ with arc set $A = \{(i, j) | i \neq j, i, j \in P \cup D\} \cup \{(i, j) | i \in T, j \in P \cup D\} \cup \{(i, 0) | i \in P \cup D \cup T\}$. Due to several constraints, some of the arcs are not included in any feasible solution. Here we present a set of rules to eliminate such arcs. In order to keep the presentation as clear as possible, we consider only cases where all elevators are idle doors open at some floors. Now $f(\cdot)$ is well-defined for $i \in T$. We call the resulting graph the *reduced graph* and denote it by $G_0 = (V, A_0)$.

The arc elimination rules are:

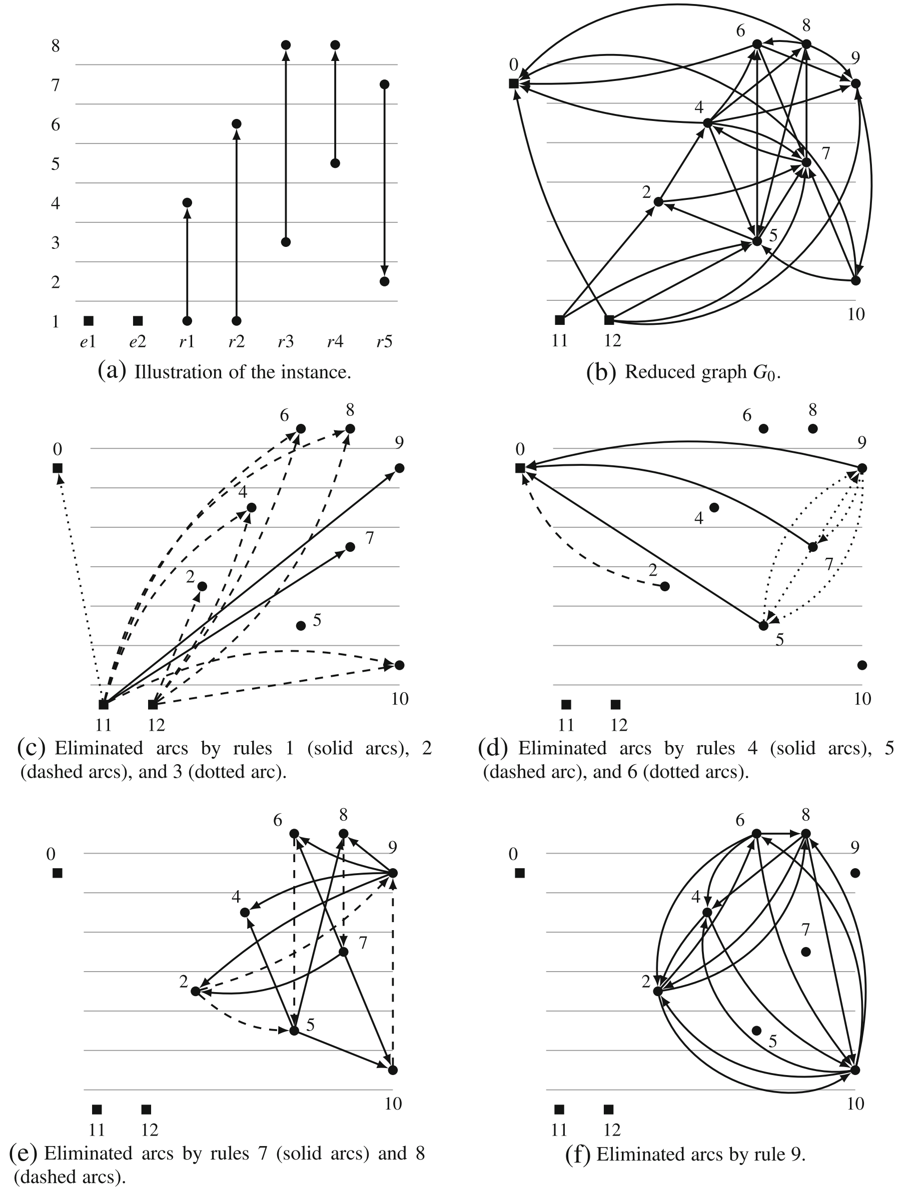
1) Arc $(2n+e, i)$, $e \in E$, $i \in P$. If request i has been fixed to some elevator, say $\hat{e}$, then all arcs $(2n+e, i)$, $e \neq \hat{e}$ are infeasible because only elevator $\hat{e}$ is allowed to visit at vertex i. Assume now that elevator $\tilde{e}$ has on-board requests and the direction of it is upwards. Let $D_{\tilde{e}} \subseteq D$ be the set of delivery vertices that are associated with the on-board request of $\tilde{e}$. If $d(i) \neq d(2n+\tilde{e})$ then $(2n+\tilde{e}, i)$ is infeasible since the delivery vertices $D_{\tilde{e}}$ of $\tilde{e}$ must be visited before $\tilde{e}$ can change its direction of travel. Let $n+k \in D_{\tilde{e}}$ be the vertex at which $\tilde{e}$ has to visit first in $D_{\tilde{e}}$. Suppose $d(i) = d(2n+\tilde{e})$. If now $f(i) < f(\tilde{e})$, or $f(i) \geq f(n+k)$, then arc $(2n+\tilde{e}, i)$ is infeasible. Similarly, some arcs are redundant in the case of downward direction.
2) Arc $(2n+e, n+i)$, $e \in E$, $n+i \in D$. Since for each request i the pickup vertex i must be visited before delivery vertex $n+i$, arc $(2n+e, n+i)$ is infeasible except the following case. Suppose that elevator $\hat{e}$ has on-board requests. Let $D_{\hat{e}} \subseteq D$ denote the set of delivery vertices associated with the on-board requests of $\hat{e}$. If $n+i$ is the vertex at which $\hat{e}$ has to visit first in $D_{\hat{e}}$, then arc $(2n+\hat{e}, n+i)$ is feasible.
3) Arc $(2n+e, 0)$, $e \in E$. If elevator e has at least one on-board or assigned request, then the vertices corresponding to those requests must be visited before going to the terminal depot vertex 0, and as a result of this, the arc $(2n+e, 0)$ cannot be the part of any feasible solution. Observe also that if an elevator group contains only a single elevator, $\hat{e}$, then arc $(2n+\hat{e}, 0)$ cannot belong to a feasible solution in any case.
4) Arc $(i, 0)$, $i \in P$. Arc $(i, 0)$ is infeasible for all $i \in P$ since delivery vertex $n+i$ must be visited after vertex i.
5) Arc $(n+i, 0)$, $n+i \in D$. Arc $(n+i, 0)$ is infeasible if vertex i has been visited by some elevator, say $\hat{e}$, and elevator $\hat{e}$ has still some passengers on-board after visiting at vertex $n+i$.
6) Arc (i, j), $i, j \in P$, $i \neq j$. If vertices i and j have been fixed to different elevators, then arc (i, j) can be eliminated. If the directions of requests i and j are not the same, e.g., request i is upwards and j is downwards, then arc (i, j) can also be eliminated. Assume now that the directions of i and j are upwards. If $f(i) > f(j)$, then arc (i, j) is infeasible. If $f(i) < f(j)$, and $f(n+i) \leq f(j)$, then arc (i, j) is infeasible as well. If $f(i) = f(j)$ and $j < i$, then arc (i, j) is infeasible, too. Similarly, some arcs are redundant in the case of downward direction.
7) Arc $(i, n+j)$, $i \in P$, $n+j \in D$. If vertices i and $n+j$ have been fixed to different elevators, then arc $(i, n+j)$ can be removed. If the directions of requests i and j are not the same, then arc $(i, n+j)$ is infeasible. Assume now that the directions of i and j are upwards. There are two cases to distinguish. (7.1) Request j is not an on-board request. Arc $(i, n+j)$ is infeasible if $f(i) \geq f(n+j)$. Arc $(i, n+j)$ is also infeasible if $f(n+i) \leq f(n+j)$. If $f(i) < f(j)$, then arc $(i, n+j)$ is infeasible as well. If $f(i) = f(j) < f(n+j) < f(n+i)$, and $i < j$, then arc $(i, n+j)$ is infeasible, too. (7.2) Request j is an on-board request. Let $\hat{e}$ be the elevator that has visited at vertex j. Arc $(i, n+j)$ is infeasible if $f(i) \geq f(n+j)$.

Arc $(i, n+j)$ is also infeasible if $f(n+i) \leq f(n+j)$. If $f(i) < f(2n+\hat{e})$, then arc $(i, n+j)$ is infeasible, too. Assume now $f(2n+\hat{e}) < f(i) < f(n+j) < f(n+i)$, and there is delivery vertex $n+k, k \neq j$, which is associated with the on-board requests of $\hat{e}$. If $f(i) < f(n+k) < f(n+j)$, then arc $(i, n+j)$ is infeasible. Also if $f(i) < f(n+k) = f(n+j)$, and k is visited later than j, then $(i, n+j)$ is infeasible. Similarly, some arcs are redundant in the case of downward direction.

8) Arc $(n+i, j), n+i \in D, j \in P$. If vertices $n+i$ and j have been fixed to different elevators, then arc $(n+i, j)$ can be excluded. Arc $(n+i, j)$ is infeasible if $j = i$, since a delivery vertex cannot be visited before the corresponding pickup vertex. Assume now that elevator $\hat{e}$ has on-board requests and the direction of it is upwards. Let $D_{\hat{e}} \in D$ denote the delivery vertices associated with the on-board request of $\hat{e}$. Suppose that vertices $n+i, n+k \in D_{\hat{e}}$, and vertex $n+k$ must be visited after $n+i$. If $d(j) \neq d(n+i)$, then arc $(n+i, j)$ is infeasible. Suppose now $d(j) = d(n+i)$. If $f(j) < f(n+i)$, or $f(j) \geq f(n+k)$, then arc $(n+i, j)$ is infeasible. Similarly, some arcs are redundant in the case of downward direction.
9) Arc $(n+i, n+j), n+i, n+j \in D$. If vertices $n+i$ and $n+j$ have been fixed to different elevators, then arc $(n+i, n+j)$ can be removed. If the directions of requests i and j are not the same, then arc $(n+i, n+j)$ is infeasible. Assume now that the direction of $n+i$ and $n+j$ is upwards. There are four cases to distinguish. (9.1) Request i is an on-board request and j is not. Let $\hat{e}$ be the elevator that has visited at vertex i. If $f(n+i) > f(n+j)$, then arc $(n+i, n+j)$ is infeasible. Also, if $f(j) < f(2n+\hat{e})$, then arc $(n+i, n+j)$ is infeasible. Arc $(n+i, n+j)$ is infeasible, too, if $f(n+i) = f(n+j)$. Arc $(n+i, n+j)$ is infeasible as well, if $f(j) \geq f(n+i)$. If $f(2n+\hat{e}) \leq f(j) < f(n+i) < f(n+j)$, but there is delivery vertex $n+k$ which is associated with the on-board requests of $\hat{e}$, and must be visited after vertex $n+i$ and before $n+j$, then arc $(n+i, n+j)$ is infeasible. (9.2) Request j is an on-board request and i is not. Let $\hat{e}$ be the elevator that has visited at vertex j. If $f(n+i) > f(n+j)$, then arc $(n+i, n+j)$ is infeasible. Also, if $f(i) < f(2n+\hat{e})$, then arc $(n+i, n+j)$ is infeasible. In other cases if there is delivery vertex $n+k$ which is associated with the on-board requests of $\hat{e}$, and must be visited after vertex $n+i$ and before $n+j$, then arc $(n+i, n+j)$ is infeasible. (9.3) Requests i and j are not on-board requests. If $f(n+i) > f(n+j)$, then arc $(n+i, n+j)$ is infeasible. Also, if $f(n+j) > f(n+i)$, and $f(j) \geq f(n+i)$, then arc $(n+i, n+j)$ is infeasible. Arc $(n+i, n+j)$ is infeasible, too, if $f(n+i) = f(n+j)$, $f(i) = f(j)$, and $i < j$. Arc $(n+i, n+j)$ is infeasible as well, if $f(n+i) = f(n+j)$, and $f(i) < f(j)$. (9.4) Requests i and j are on-board requests of $\hat{e}$. If $f(n+i) > f(n+j)$, then arc $(n+i, n+j)$ is infeasible. If $f(n+i) \leq f(n+j)$, but there is delivery vertex $n+k$ which is associated with the on-board requests of $\hat{e}$, and must be visited after vertex $n+i$ and before $n+j$, then arc $(n+i, n+j)$ is infeasible. Similarly, some arcs are redundant in the case of downward direction.

Remark 1 The arcs that belong to set O are also eliminated by these rules.

Example 1 Consider an instance with two elevators, $e1$ and $e2$, and five requests, $r1-r5$, of which $r1$ and $r2$ are on-board requests of $e1$. The floors at which the elevators



(a) Illustration of the instance.

(b) Reduced graph G_0.

(c) Eliminated arcs by rules 1 (solid arcs), 2 (dashed arcs), and 3 (dotted arc).

(d) Eliminated arcs by rules 4 (solid arcs), 5 (dashed arc), and 6 (dotted arcs).

(e) Eliminated arcs by rules 7 (solid arcs) and 8 (dashed arcs).

(f) Eliminated arcs by rule 9.

Fig. 1 Illustration of the *arc* elimination for the instance with 2 elevators and 5 requests

are residing, and the arrival and destination floors of the requests, are illustrated in Fig. 1a. For requests, the arc shows the direction. Figure 1b illustrates the reduced graph G_0 of the instance. In this figure vertices associated with requests are indexed from 1 to 10, elevator origin vertices are indexed by 11 and 12, and the common terminal depot vertex is indexed by 0. Notice that vertices 1 and 3 are omitted since

$r1$ and $r2$ are on-board requests. The remaining four figures illustrate the eliminated arcs. In Fig. 1c solid, dashed and dotted arcs represent arcs that are eliminated by rules 1, 2, and 3, respectively. In Fig. 1d solid, dashed and dotted arcs represent arcs that are eliminated by rules 4, 5, and 6, respectively. In Fig. 1e solid and dashed arcs are the arcs that are eliminated by rules 7 and 8, respectively. Finally, in Fig. 1f solid arcs depict arcs that are eliminated by rule 9. In the original graph, $|A| = 82$, whereas in the reduced graph, $|A_0| = 30$, thus in total 52 arcs are eliminated by rules 1–9.

3 Valid inequalities

In this section we present several new families of valid inequalities for the EDP. In the rest of paper, we use the following additional notation. Let L_{ij}, $i, j \in V$, be the set of vertices that belongs to the longest path, in terms of the number of vertices, from i to j such that on this path an elevator does not change its direction of travel and the path is feasible at least for one elevator, and let $K_{ij} = L_{ij} \setminus \{i, j\}$.

It should be noted that since the EDP is very similar to other routing problems such as the Pickup and Delivery Problem and the Dial-a-Ride Problem, almost all the valid inequalities presented for them can be made valid for the EDP.

3.1 Lifted subtour elimination inequalities

Consider the following set of the simple subtour elimination constraints,

$$x(S, S) \leq |S| - 1 \quad \forall S \subseteq V, 2 \leq |S| \leq |V| - 1, \tag{27}$$

originally proposed by Dantzig et al. (1954) for the TSP. In the presence of precedence relationships, these inequalities can be lifted. Balas et al. (1995) proposed the following two families of lifted subtour elimination inequalities, called *π-inequalities* and *σ-inequalities*, for the Precedence-Constrained Asymmetric TSP:

$$x(S, S) + x(S, \overline{S} \cap \pi(S)) + x(S \cap \pi(S), \overline{S} \setminus \pi(S)) \leq |S| - 1, \tag{28}$$

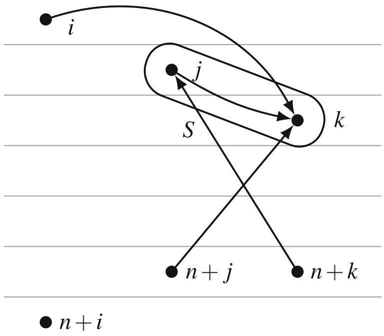
$$x(S, S) + x(\overline{S} \cap \sigma(S), S) + x(\overline{S} \setminus \sigma(S), S \cap \sigma(S)) \leq |S| - 1. \tag{29}$$

These inequalities also apply to the EDP for $S \subset P \cup D$. We next introduce another family of lifted subtour elimination inequalities.

Proposition 1 *Let $S = (s_1, \ldots, s_r) \subseteq P$ be an ordered set of pickup vertices that can be visited consecutively without changing the direction of travel, and let $U = \{(i, s_j) \in A_0 | i \in P, i \notin S, j > 1\}$ be a set of arcs such that for the starting vertex of an arc in this set it holds $K_{i,n+i} \supseteq S \cup \sigma(S)$. Then the following inequality is valid for the EDP:*

$$x(S, S) + x(U) + x(\sigma(S), S) \leq |S| - 1. \tag{30}$$

Fig. 2 Lifted σ-inequality for $S = \{j, k\}$



Proof We prove this by contradiction. Let H denote the union of arc sets (S, S), U, and $(\sigma(S), S)$. Observe that these sets are disjoint. Suppose now that inequality (30) is violated in a feasible solution (x, t). Then there is a set of pickup vertices S that satisfies the conditions of Proposition 1 such that $x(H) \geq |S|$. This means that it holds $x(H \cap (V, s_j)) = 1$ for each $s_j \in S$ since the ending vertex of each arc $(i, j) \in H$ is in S. In particular, it holds $x(\sigma(S) \cap (V, s_1)) = 1$ as $(S, S) \cap (V, s_1) = U \cap (V, s_1) = \emptyset$. The former condition comes from the fact that for each pickup vertex pair k and l, both arcs (k, l) and (l, k) are not present in the reduced graph A_0. Consider now the path where arc $(n + s_p, s_1)$, for some $n + s_p \in \sigma(S)$ is used. Track the path backwards and let s_k be the first vertex visited in S on this path. The incoming arc to that vertex is from U, otherwise $x(H \cap (V, s_k)) = 0$. Let (h, s_k) denote that arc. There are two cases to be considered. Suppose first that vertex $n + h$ is visited before $n + s_p$. Since $K_{h,n+h} \supseteq S \cup \sigma(S)$, there must be at least one vertex from S, say s_q, that is visited after $n + h$ and before $n + s_p$ on the path. This implies $x(H \cap (V, s_q)) = 0$, a contradiction. Suppose next that $n + h$ is visited after $n + s_p$. Since vertex s_1 can be visited before s_p, arc $(n + s_p, s_1)$ belongs to set R. However, at vertex $n + s_p$ passengers related to request h are still on-board, a contradiction. Consequently it must hold $x(S, S) + x(U) + x(\sigma(S), S) \leq |S| - 1$. $\square$

Remark 2 Since σ-inequality for any S that satisfies the conditions of Proposition 1 is $x(S, S) + x(\sigma(S), S) \leq |S| - 1$, inequalities (30) can be considered as lifted σ-inequalities.

Example 2 Consider an instance with three requests i, j and k such that $f(i) > f(j) > f(k) > f(n + j) = f(n + k) > f(n + i)$. Let $S = \{j, k\}$. Now $\sigma(S) = \{n + j, n + k\}$. The inequality (30) for this S is $x_{j,k} + x_{i,k} + x_{n+j,k} + x_{n+k,j} \leq 1$, and is illustrated in Fig. 2. Observe that for this S, the σ-inequality is $x_{j,k} + x_{n+j,k} + x_{n+k,j} \leq 1$.

3.2 Strengthened precedence constraints

The precedence constraints (4) can be strengthened by taking into account the fact that an elevator is permitted to change its direction of travel only when it is empty. This leads to the following proposition.

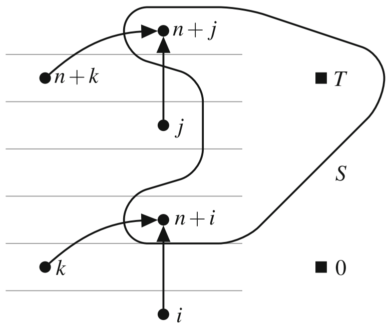


Fig. 3 Strengthened precedence constraint for $S = \{n+i, n+j, T\}$, and $U = \{i, j\}$

Proposition 2 *For any $S \subset V$, for which $1+|T| \leq |S| \leq |V|-2$ with $T \subset S, 0 \notin S$, and for which there is a set of pickup vertices U such that for each $i \in U, i \notin S, n+i \in S$, and any feasible path from i to $n+i$ does not contain any vertex from U other than i, the inequality*

$$x(\overline{S}, S) \geq |U| \tag{31}$$

is valid for the EDP.

Proof We prove this also by contradiction. Assume that the inequality is violated in an integer feasible solution x. Then there is a set of vertices $U \subseteq P$, and a set S that satisfy the conditions given in the proposition such that $x(\overline{S}, S) \leq |U|-1$. This means that at least two pickup vertices, $i, j \in U$, are visited by the same elevator, and both are visited before their delivery vertices $n+i$ and $n+j$. Assume now that vertex i is visited before j. By the definition of set U, any path from i to $n+i$ via vertex j is infeasible. This is a contradiction. Similar argument holds for the case where vertex j is visited before i. Consequently, $x(\overline{S}, S) \geq |U|$. □

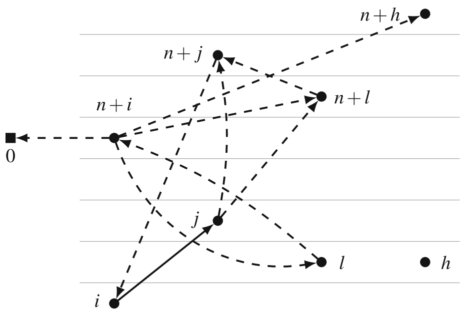
Example 3 Consider an instance with three requests, i, j and k such that $f(i) < f(k) < f(n+i) < f(j) < f(n+k) < f(n+j)$. The vertex sets $S = \{n+i, n+j, T\}$ and $U = \{i, j\} \subseteq P$ satisfy the condition given in Proposition 2. The resulting strengthened precedence constraint is $x_{i,n+i} + x_{k,n+i} + x_{j,n+j} + x_{n+k,n+j} \geq 2$. This is illustrated in Fig. 3.

3.3 Logical implications

In this subsection we introduce a set of valid inequalities which we call the logical implications. These inequalities, presented in the following four propositions, are obtained by first identifying arc pairs for which it holds that both arcs cannot be used in any feasible m-route, and then combining them, by taking into account the degree equations. Since the proofs of these propositions are very similar to each other, we only provide the proof for the first one.

Proposition 3 *Let $i, j \in P$, $(i, j) \in A_0$, and p denote the vertex in $\{n+i, n+j\}$ that is visited before $r = \{n+i, n+j\} \setminus \{p\}$ in each feasible route where arc (i, j) is used. In addition, let U be one of the following sets:*

Fig. 4 If decision variable x_{ij} (*solid arc*) is fixed to 1, then all the decision variables corresponding to *dashed arcs* can be set to 0



$$
\begin{aligned}
U_j^+ &= \{(j,k) \in A_0 | k \notin K_{jp} \setminus \{n+l | l \in P, l \in L_{ij}, n+l \in L_{jp}\}\},\\
U_p^- &= \{(k,p) \in A_0 | k \notin K_{jp} \setminus \{n+l | l \in P, l \in L_{ij}, n+l \in L_{jp}\}\},\\
U_p^+ &= \{(p,k) \in A_0 | k \notin K_{pr} \setminus \{n+l | l \in P, l \in L_{ij}, n+l \in L_{pr}\}\},\\
U_r^- &= \{(k,r) \in A_0 | k \notin K_{pr} \setminus \{n+l | l \in P, l \in L_{ij}, n+l \in L_{pr}\}\},\\
U_r^+ &= \{(r, p-n)\} \cup \{(r, n+l) \in A_0 | l \in P, l \in L_{ij}\},\\
U_i^- &= \{(k,i) \in A_0 | k \in P, n+k \in L_{ij}\}.
\end{aligned}
$$

Then for U, the following inequality is valid for the EDP:

$$
x_{ij} + \sum_{(k,l)\in U} x_{kl} \le 1. \tag{32}
$$

Proof Suppose that x is a feasible solution, but inequality (32) is violated for some U. Assume $U = U_j^+$. Since at most one arc can be used from U_j^+ due to the degree constraint (3), it holds that $x_{ij} + x_{jk} = 2$ for some $k \notin L_{jp} \setminus \{n+l | l \in P, l \in L_{ij}, n+l \in L_{jp}\}$. This means that both arcs (i,j) and (j,k) are used. Consider now a trip H in which these arcs are used. Since (j,k) is used, then either vertex p is passed by on trip H, or delivery vertex $n+l$ is visited on H, but l is not. These in turn mean that at least one precedence or reversal constraint is violated, a contradiction. Similarly arguments can be derived for other sets U_p^+, U_p^-, U_r^+, and U_r^-. □

Example 4 Consider an instance with four up-requests i, j, l, and h, such that $f(i) < f(l) = f(h) < f(j) < f(n+i) < f(n+l) < f(n+j) < f(n+h)$. This instance is illustrated in Fig. 4. In this figure each dashed arc corresponds to the decision variable whose value cannot be equal to 1 if $x_{ij} = 1$ (solid arc), and vice versa.

Proposition 4 *Let $i \in P$, $n+j \in D$, and $(i, n+j) \in A_0$. In addition, let U be one of the following sets:*

$$
\begin{aligned}
U_j^+ &= \{(j,k) \in A_0 | k \notin K_{ji} \setminus \{l | l \in P, l \in L_{ji}, n+l \in L_{i,n+j}\}\},\\
U_i^- &= \{(k,i) \in A_0 | k \notin K_{ji} \setminus \{l | l \in P, l \in L_{ji}, n+l \in L_{i,n+j}\}\},\\
U_{n+j}^+ &= \{(n+j,k) \in A_0 | k \notin K_{n+j,n+i} \setminus \{n+l | l \in P, l \in L_{i,n+j},\\
&\qquad n+l \in L_{n+j,n+i}\}\},
\end{aligned}
$$

$$\begin{aligned}
U^{-}_{n+i} &= \{(k, n+i) \in A_0 | k \notin K_{n+j,n+i} \setminus \{n+l | l \in P, l \in L_{i,n+j},\\
&\qquad n+l \in L_{n+j,n+i}\}\},\\
U^{+}_{n+i} &= \{(n+i, j)\} \cup \{(n+i, n+l) \in A_0 | l \in P, l \in L_{i,n+j}\},\\
U^{-}_{j} &= \{(k, j) \in A_0 | k \in P, n+k \in L_{i,n+j}\}.
\end{aligned}$$

Then for U, the following inequality is valid for the EDP:

$$x_{i,n+j} + \sum_{(k,l)\in U} x_{kl} \leq 1. \tag{33}$$

Proposition 5 *Let $n+i, n+j \in D$, $(n+i, n+j) \in A_0$, and p denote the vertex in $\{i, j\}$ that is visited before $r = \{i, j\} \setminus \{p\}$ in each feasible route where arc $(n+i, n+j)$ is used. In addition, let U be one of the following sets:*

$$\begin{aligned}
U^{-}_{p} &= \{(k, p) \in A_0 | k \in P, n+k \in L_{n+i,n+j}\},\\
U^{+}_{p} &= \{(p, k) \in A_0 | k \notin K_{pr} \setminus \{l | l \in P, l \in L_{pr}, n+l \in L_{n+i,n+j}\}\},\\
U^{-}_{r} &= \{(k, r) \in A_0 | k \notin K_{pr} \setminus \{l | l \in P, l \in L_{pr}, n+l \in L_{n+i,n+j}\}\},\\
U^{+}_{r} &= \{(r, k) \in A_0 | k \notin K_{r,n+i} \setminus \{l | l \in P, l \in L_{r,n+i}, n+l \in L_{n+i,n+j}\}\},\\
U^{-}_{n+i} &= \{(k, n+i) \in A_0 | k \notin K_{r,n+i} \setminus \{l | l \in P, l \in L_{r,n+i}, n+l \in L_{n+i,n+j}\}\},\\
U^{+}_{n+j} &= \{(n+j, i)\} \cup \{(n+j, n+l) \in A_0 | l \in P, l \in L_{n+i,n+j}\}.
\end{aligned}$$

Then for U, the following inequality is valid for the EDP:

$$x_{n+i,n+j} + \sum_{(k,l)\in U} x_{kl} \leq 1. \tag{34}$$

Proposition 6 *Let $n+i \in D$, $j \in P$, and $(n+i, j) \in A_0$. In addition, let U be one of the following sets:*

$$\begin{aligned}
U^{-}_{i} &= \{(k, i) \in A_0 | k \in P, n+k \in L_{n+i,j}\},\\
U^{+}_{i} &= \{(i, k) \in A_0 | k \notin K_{i,n+i} \setminus \{l | l \in P, l \in L_{i,n+i}, n+l \in L_{n+i,j}\}\},\\
U^{+}_{i} &= \{(k, n+i) \in A_0 | k \notin K_{i,n+i} \setminus \{l | l \in P, l \in L_{i,n+i}, n+l \in L_{n+i,j}\}\},\\
U^{+}_{j} &= \{(j, i) \cup (j, n+i)\} \cup \{(j, n+l) \in A_0 | l \in P, l \in K_{n+i,j}, n+l \in K_{j,n+j}\},\\
U^{-}_{n+j} &= \{(n+l, n+j) \in A_0 | l \in P, l \in K_{n+i,j}, n+l \in K_{j,n+j}\},\\
U^{+}_{n+j} &= \{(n+j, i) \cup (n+j, n+i)\} \cup \{(n+j, n+l) \in A_0 | l \in P, l \in K_{n+i,j}\}.
\end{aligned}$$

Then for U, the following inequality is valid for the EDP:

$$x_{n+i,j} + \sum_{(k,l)\in U} x_{kl} \leq 1. \tag{35}$$

Remark 3 One could obtain more complex logical implications by considering multiple arcs simultaneously.

3.4 Passing equalities

The following three equalities are obtained by identifying arc pairs for which it holds that either both arcs are used, or neither arcs is used in any feasible m-route. We do not provide proofs for propositions 8 and 9 since they are very similar to that of Proposition 7.

Proposition 7 *Let $i, j \in P$, $(i, j) \in A_0$, and let p denote the vertex in $\{n+i, n+j\}$ that is visited before $r = \{n+i, n+j\} \backslash \{p\}$ in each feasible route. If $L_{pr} \subset D$, $L_{ij} \subset P$, and $v \in L_{ij}$ implies that $n + v \in L_{pr}$ and vice versa, then the following equality,*

$$x_{ij} - x_{pr} = 0, \tag{36}$$

is valid for the EDP.

Proof Suppose first that $x_{ij} = 1$. Then in any feasible solution there must be a path from p to r. Consider a trip H in which the arc (i, j) is used. Since $v \in L_{ij}$ implies that $n + v \in L_{pr}$ and vice versa, and $L_{pr} \subset D$, none of the vertices in L_{pr} can be visited on H, thus $x_{pr} = 1$. Similarly, $x_{pr} = 1$ implies $x_{ij} = 1$. Suppose next that $x_{ij} = 0$. Now there are three cases: (1) requests i and j are serviced by different elevators; (2) vertices i and j are serviced by the same elevator, but not on the same trip; and (3) both vertices are visited on the same trip and at least one vertex in L_{ij} is visited on that trip. If i and j are visited by different elevators, then p and r must be visited by different elevators as well, thus $x_{pr} = 0$. If i and j are visited by the same elevator but on different trips, then vertices p and r are also visited on different trips, hence $x_{pr} = 0$. If at least one vertex from L_{ij} is visited on a trip that contains a path from i and j, then the corresponding delivery vertices must be visited on that trip. Since those delivery vertices belong to L_{pr}, it holds that $x_{pr} = 0$. Similarly $x_{pr} = 0$ implies $x_{ij} = 0$. Consequently, $x_{ij} - x_{pr} = 0$.

Corollary 1 *If $L_{pr} = \emptyset$ and $L_{ij} \neq \emptyset$, then (36) reduces to*

$$x_{ij} \leq x_{pr}. \tag{37}$$

Example 5 Consider an instance with three requests i, j, and k such that $f(i) = f(j) = f(k) < f(n+i) < f(n+j) < f(n+k)$. The passing equalities are the following: $x_{ij} - x_{n+i,n+j} = 0$, $x_{jk} - x_{n+j,n+k} = 0$, and $x_{ik} - x_{n+i,n+k} = 0$. These are illustrated in Fig. 5 where solid arcs correspond to the first equality, dashed arcs to the second one, and dotted arcs to the last one.

Proposition 8 *Let $i, j \in P$, $(i, j) \in A_0$, and let p denote the vertex in $\{n+i, n+j\}$ that is visited before $r = \{n+i, n+j\} \backslash \{p\}$ in each feasible route. If $L_{jp} \subset D$, $L_{ij} \subset P$, and $v \in L_{ij}$ implies that $n + v \in L_{jp}$, and vice versa, then the following equality,*

$$x_{ij} - x_{jp} = 0, \tag{38}$$

is valid for the EDP.

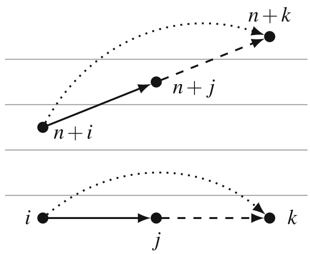


Fig. 5 Passing equalities for the instance with three up-requests

Proposition 9 *Let* $i \in P$, $n+j \in D$, *and* $(i, n+j) \in A_0$. *If* $L_{n+j,n+i} \subset D$, $L_{i,n+j} \subset P$, *and* $v \in L_{i,n+j}$ *implies that* $n+v \in L_{n+j,n+i}$, *and vice versa, then the following equality,*

$$x_{i,n+j} - x_{n+j,n+i} = 0, \tag{39}$$

is valid for the EDP.

4 Polyhedral analysis

4.1 Description of the case studied

Polyhedral analysis for the EDP is very difficult to carry out in a general case since, for example, the number of routing variables is very dependent on how the given requests are located in a building. Therefore we have to restrict ourselves to special cases. Possible choices are up-peak and down-peak traffic conditions. During *down-peak*, which happens for example in office buildings typically at the end of the day, most (all) of the passengers leave the floors of a building and travel primarily to the lobby in order to exit the building whereas during *up-peak*, which happens for example in the morning, the traffic is just to the opposite direction. Natural choice would be the up-peak traffic since it is considered as the most critical traffic situation from the elevator group sizing point of view; if an elevator group can handle up-peak, it can handle the other traffic peaks as well. Yet, here we study a generic case arising in the down-peak traffic pattern because down-peak is more challenging than up-peak from the polyhedral analysis standpoint, see for example Theorems 3 and 4.

In the considered case, the following five assumptions hold. (1) There are n non-assigned requests such that $f(1) > \ldots > f(n)$, $f(n+1) = \ldots = f(2n) = 1$, and $[a_i, b_i] = [0, \infty)$ for $i = 1, \ldots, 2n$. (2) The exit floor of a building is at the first (lowest) floor. (3) The number of elevators is equal to n. (4) Each elevator has an unlimited capacity, $Q = \infty$. (5) All elevators are symmetrical.

We use the approach presented in Sect. 2.3 to eliminate the symmetric elevators. Let $+0$ denote the merged vertices. If we now required that each elevator ended its route at vertex 0, there would be n arcs from $+0$ to 0. Instead, we require that if an elevator serves at least one vertex then it ends its route at 0, otherwise it stays at $+0$. This can be done by excluding the out-degree equation of $+0$ and removing all arcs

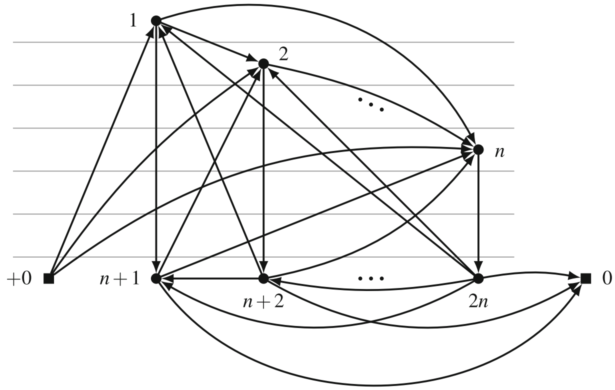


Fig. 6 Reduced graph for the case studied

from $+0$ to 0. Now the out-degree equations, (3), become

$$x(i, V) = 1, \quad \forall i \in P \cup D. \tag{40}$$

Figure 6 illustrates the graph of the considered case where all redundant arcs eliminated and elevator origin vertices are merged into a single vertex.

The next theorem gives the number of different feasible solutions projected onto x space for the case studied. The reason why we count the number of feasible solutions in x space instead of (x, t) is that the number of them in the latter space is clearly infinite. Let $S(n, k)$ denote the Stirling number of the second kind,

$$S(n, k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^j \binom{k}{j} (k - j)^n.$$

Theorem 3 *The number of feasible solutions in x space in the down-peak case is*

$$\sum_{m=1}^{n} \sum_{k=m}^{n} S(n, k) * k! * \binom{k-1}{m-1} / (m!), \tag{41}$$

Proof Let us first consider an m-route where the elevators leave the first floor to serve requests exactly $k \geq m$ times. Such m-routes can be obtained by first dividing the set of n requests into exactly k nonempty ordered clusters, and then allocating the clusters to m elevators with the requirement that each elevator serves the clusters allocated to it in an ascending order and it must serve all requests in a cluster before it can start serving requests in the next cluster. The number of ways to generate k ordered clusters is $S(n, k) * k!$. As we consider m-routes, the clusters should be allocated so that each elevator $1, \ldots, m$ gets at least one cluster. There are

$$\binom{k-1}{m-1}$$

Table 1 The number of feasible solutions for the TSP, 1-PDP, and 1-EDP-up

n	TSP	1-PDP	1-EDP-DOWN	1-EDP-UP
1	2	1	1	1
2	24	6	3	2
3	720	90	13	5
4	49.320	2.520	75	15
5	3.628.800	113.40	541	52
6	479.001.600	7.484.400	4.683	203
7	87.178.291.200	681.080.400	47.293	877
8	20.922.789.888.000	81.729.648.000	545.835	4.140

ways to do that (repetitions are allowed). According to the multiplicative principle we have

$$S(n,k) * k! * \binom{k-1}{m-1}$$

different m-routes where there are exactly k clusters. The fact that clusters are ordered implies that the elevators are ordered as well. To eliminate that, we divide the number of m-routes by $m!$, which is the number of ways to label m elevators. Summing the resulting expression over $m = 1, \ldots, n$ and $k = m, \ldots, n$ we obtain the result. □

Corollary 2 *If all elevators are asymmetrical but still idle at some floors and the number of them is finite, say l, then the number of feasible solutions in x space is*

$$\sum_{m=1}^{\min\{l,n\}} \sum_{k=m}^{n} S(n,k) * k! * \binom{k-1}{m-1}. \tag{42}$$

For comparison reasons we also provide the number of feasible solutions in the up-peak case, which is obtained from the considered case by swapping the arrival and destination floors for each request.

Theorem 4 *The number of feasible solutions in x space in the up-peak case is*

$$\sum_{k=1}^{n} S(n,k). \tag{43}$$

Proof Since now the arrival floor of each request is the same, the lowest floor, and all elevators are uncapacited, the number of feasible solutions equals to Bell number. □

In Table 1 we report the number of solutions in the case of a single elevator for both traffic patterns, up-peak (1-EDP-UP) and down-peak (1-EDP-DOWN), as a function of the number of requests, n. For comparison reasons we report the number of feasible solutions for a single vehicle Pickup and Delivery Problem (1-PDP) (which are taken from Carrabs et al. 2007), and for TSP, where the number of cities to be visited is $2n$. From this table we can see that down-peak is more challenging traffic pattern than up-peak, and the problem specific constraints of the EDP reduce the number of feasible solutions significantly compared to the PDP and TSP.

4.2 Dimension of the EDP polytope

In this subsection we aim at determining the dimension of the EDP polytope, P_{EDP}, defined as the convex hull of the feasible solutions of the problem. That is,

$$\begin{aligned} P_{EDP} = \text{conv}\{(x,t) \in \mathbb{R}^{|A_0| \times (2n+2)} | (x,t) \text{ satisfies} \\ (2), (4) - (7), (14), (15), (17), (19), \text{ and } (40)\}, \end{aligned} \tag{44}$$

where $|E| = n$, $Q = \infty$, $[a_i, b_i] = [0, \infty)$ for $i = 1, \ldots, 2n$, and A_0 is the set of arcs in the reduced graph G_0. We next give two lemmas needed in the determination of the dimension of P_{EDP}.

Lemma 1 *The number of arcs, $|A_0|$, is*

$$2n^2 + n. \tag{45}$$

Proof Since $|(\{j\}, \overline{\{j\}})| = n - j + 1$ for $j = 1, \ldots, n$, $|(\{n+j\}, \overline{\{n+j\}})| = n + j - 1$ for $j = 1, \ldots, n$, and $|(\{+0\}, \overline{\{+0\}})| = n$, hence $|A_0| = n + 1/2n(n+1) + n^2 + 1/2n(n-1) = 2n^2 + n$. □

Lemma 2 *The rank of the matrix induced by equality constraints* (2)*,* (36)*, and* (38) *–* (40) *is*

$$\frac{1}{2}(n^2 + 5n). \tag{46}$$

Proof Observe first that for each $i, j \in P$, $i < j$, there exist L_{ij} and L_{pr} that satisfy the conditions of Proposition 7 whereas for any $i, j \in P \cup D$ there are no sets that satisfy the conditions of Proposition 8 and 9. The number of equations in (2) and (40) are both $2n$. Since it holds that $x(n, V) = x(V, 2n) = x_{n,2n}$, equation $x(n, V) = 1$ is a linear combination of $x(V, 2n) = 1$, and vice versa. For each $i \in P, i < n$, equation $x_{i,n} - x_{2n,n+i} = 0$ is a linear combination of $x(i, V) = 1$, $x(V, n+i) = 1$, and $x_{ij} - x_{n+j,n+i} = 0$ for $j = i+1, \ldots, n-1$ with coefficients $1, -1$, and -1 for $j = i+1, \ldots, n-1$. Let us now consider the following family of equations:

$$x(V, i) = 1, \quad \forall i \in P \cup D, \tag{47}$$

$$x(i, V) = 1, \quad \forall i \in P \setminus \{n\} \cup D, \tag{48}$$

$$x_{ij} - x_{n+j,n+i} = 0, \quad \forall i \in P \setminus \{n, n-1\}, j \in P \setminus \{n\}, i < j. \tag{49}$$

We next prove that (47)–(49) are linearly independent. This is accomplished by providing for each equation $ax = a_0$ of the family (47)–(49) a point $x \in \mathbb{R}^{|A_0|}$

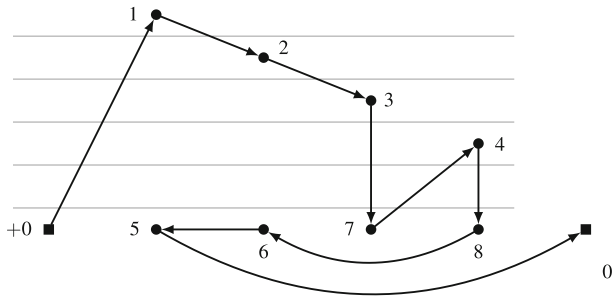


Fig. 7 A construction for the proof of Lemma 2

satisfying all the equations in the family except $ax = a_0$. For each Eq. (47), $i \in P$, let x be the characteristic vector of route $(i, n+i, 1, 2 \ldots, i-1, i+1, \ldots, n-1, n, 2n, 2n-1, \ldots, n+i+1, n+i-1, \ldots, n+1, 0)$. For each Eq. (47), $n+i \in D$, let x be the characteristic vector of route $(n+i, 1, 2, \ldots, i-1, i+1, \ldots, n-1, 2n-1, 2n-2, \ldots, n+i+1, n+i-1, \ldots, n+1, i, n, 2n, 0)$. For each Eq. (48), $i \in P \setminus \{n\}$, let x be the characteristic vector of the union of routes $(+0, i)$ and $(+0, 1, n+1, 2, n+2, \ldots, i-1, n+i-1, i+1, n+i+1, \ldots, n, 2n, n+i, 0)$. For each Eq. (48), $n+i \in D$, let x be the characteristic vector of route $(+0, 1, \ldots, i-1, i+1, \ldots, n, 2n, 2n-1, \ldots, n+i+1, n+i-1, \ldots, n+1, i, n+i)$. For each Eq. (49), $i \in P \setminus \{n, n-1\}$, $j \in P \setminus \{n\}, i < j$, let x be the characteristic vector of route $(+0, 1, \ldots, i, j, n+j, j+1, n+j+1, \ldots, n, 2n, n+i, n+i-1, \ldots, n+1, 0)$. This last route is illustrated in Fig. 7 for the case of four requests where $i = 2$ and $j = 3$. Therefore, the rank of the matrix induced by equality constraints (2), (36), and (38)–(40) is $2n + 2n - 1 + 1/2(n-1)(n-2) = 1/2(n^2 + 5n)$, and (47)–(49) define an equality system for P_{EDP}. □

Theorem 5 *The dimension of* P_{EDP}, $\dim(P_{EDP})$, *is*

$$\frac{1}{2}(3n^2 + n + 4). \tag{50}$$

Proof The number of variables x_{ij} and t_i are $|A_0|$ and $2n+2$, respectively. Lemmas 1 and 2 imply that $\dim(P_{EDP}) \leq |A_0| + 2n + 2 - 1/2(n^2+5n) = 1/2(3n^2+n+4)$. We next show that this bound is tight by exhibiting $1/2(3n^2+n+4)+1$ affinely independent vertices (x, t) of P_{EDP}. Set $H = \{(i, j) \in A_0 | i, j \in P, i < j\} \cup \{(n+i, j) \in A_0 | i, j \in P, i \neq j\}$. Let h^{ij} be a feasible solution (x, t) where x is the characteristic vector of m-route such that (i, j) is the only arc traveled from set H, and for t it holds that $t_{+0} = 0$, and $t_l = t_k + \tau_{kl}$ for $l \in P \cup D \cup \{0\}$, where k is the immediate predecessor of l in some elevator route induced by x. Similarly, let h be a feasible solution (x, t) where x is the characteristic vector m-route such that none of the arcs from H is traveled, and t is defined in the same way as in the previous case. Let z be a feasible solution (x, t) where x is the characteristic vector of 1-route, denoted by p, and t is defined as before. Let z^i_ϵ be a solution where x and t are the same as in z but the values of t_i and t_j for all the successors j of i in p are increased by ϵ. Clearly, all solutions h^{ij}, $(i, j) \in H, h$, and $z^i_\epsilon, i \in V$ are affinely independent, and the number of them is $1/2(3n^2 - 3n) + 1 + 2n + 2 = 1/2(3n^2 + n + 4) + 1$. □

4.3 Facets of the EDP polytope

In this subsection we show which valid inequalities derived are facet defining. Since $2n + 2$ affinely independent solutions can be constructed from any feasible solution (x, t) by delaying the arrival times of elevators at some vertices, see for example the proof of Theorem 5, and t variables increase the dimension of the EDP polytope by $2n + 2$, we may consider for simplicity only the x space. The dimension of the EDP polytope restricted onto x space is $1/2(3n^2 + n + 4) - (2n + 2) = 3/2(n^2 - n)$. Furthermore, in the rest of the paper we do not make any difference between a feasible m-route and the characteristic vector x of it.

We begin with the upper and lower bound constraints of x.

Theorem 6 *Let $n \geq 3$, then inequality $x_{ij} \leq 1$, $(i, j) \in A_0$ defines a facet of P_{EDP} only if $i = n - 1$ and $j = 2n - 1$.*

Proof Put $H = \{(i, j) \in A_0 | i, j \in P, i < j\} \cup \{(n+i, j) \in A_0 | i, j \in P, i \neq j\}$. Let $h^{ij}, i \neq n - 1, j \neq 2n - 1$ be a feasible m-route in which only arcs $(n - 1, 2n - 1)$ and (i, j) are used from H. In addition, let h be a feasible m-route in which only $(n - 1, 2n - 1)$ is used from H. Since solutions h^{ij} and h are affinely independent and the number of them is $3/2(n^2 - n)$, inequality $x_{n-1,2n-1} \leq 1$ defines a facet of P_{EDP}. We next show that $x_{ij} \leq 1$ is not facet inducing for any other i and j. Inequality $x_{n,2n} \leq 1$ is not facet inducing since the face defined by it is not proper, i.e., it is satisfied at equality in every feasible m-route. Note that for any other arc (i, j) either the out-degree of i or the in-degree of j is at least 3, implying that $\dim(\{x \in P_{EDP} | x_{ij} = 1, (i, j) \in A_0 \setminus \{(n-1, 2n-1), (n, 2n)\}\})$ is at most $3/2(n^2-n) - 2$. □

Theorem 7 *Let $n \geq 4$, then inequality $x_{ij} \geq 0$ defines a facet of P_{EDP} for all $(i, j) \in A_0 \setminus \{(n - 2, 2n - 2), (n - 1, 2n - 1), (n, 2n)\}$.*

Proof We first show that inequalities $x_{n-2,2n-2} \geq 0$, $x_{n-1,2n-1} \geq 0$, and $x_{n,2n} \geq 0$ are not facet inducing. Since $x_{n,2n} \leq 1$ is satisfied at equality in every feasible m-route, $x_{n,2n} \geq 0$ is not facet inducing. If $x_{n-1,2n-1} = 0$ then $x_{n-1,n} = 1$, which in turn implies that $x_{k,n} = 0$ for all $(k, n) \in A_0, k \neq n - 1$. Thus, $x_{n-1,2n-1} \geq 0$ is not facet inducing either. Inequality $x_{n-2,2n-2} \geq 0$ is not facet inducing since $x_{n-2,2n-2} = 0$ implies that $x_{2n-1,n} = 0$. To see this let $x_{n-2,2n-2} = 0$. Now either $x_{n-2,n-1} = 1$, or $x_{n-2,n} = 1$. Suppose first $x_{n-2,n-1} = 1$. Due to the passing equality, it must hold that $x_{2n-1,2n-2} = 1$, which in turn implies that $x_{2n-1,n} = 0$. Suppose next $x_{n-2,n} = 1$. Due to the in-degree equations, this implies $x_{2n-1,n} = 0$ as well.

Now let us consider the lower bound inequalities for the remaining arcs. It is easy to see that if $x_{ij} = 0$ for any remaining arc, then the dimension is decreased by one, which proves that they are facet inducing. □

We will next prove that lifted σ-inequalities are facet inducing. Here, we need some concise notation for elevator routes. Let $\{S_1, \ldots, S_k\}$ be a family of mutually disjoint sets of pickup and delivery vertices such that for each $i = 1, \ldots, k$ it holds that if $|S_i| \geq 2$, then $j \in S_i$, $j \in P$, implies $n + j \in S_i$, and vice versa. We denote by $(+0, S_1, \ldots, S_k, 0)$ any feasible elevator route that starts at $+0$, ends at 0, and visits all vertices in S_i before those in S_{i+1} for $i = 1, \ldots, k-1$, such that for each pickup vertex l in S_i, the immediate successor of it is $n + l$.

Theorem 8 *Let $n \geq 3$, and let $S = (s_1, \ldots, s_r) \subseteq P \setminus \{n\}$, $|S| \geq 2$ be an ordered set of pickup vertices that can be visited consecutively without changing the direction of travel, and $U = \{(i, s_j) \in A_0 | i \in P, i \notin S, j > 1\}$ be a set of arcs such that for the starting vertex of an arc in this set it holds $K_{i,n+i} \supseteq S \cup \sigma(S)$. Then the lifted σ inequality (30) defines a facet of P_{EDP}.*

Proof In this proof we need a requirement that each elevator leaves the vertex $+0$. We employ this requirement by first adding an auxiliary arc between vertices $+0$ and 0, then by associating with the new arc a non-negative integer variable,

$$x_{+0,0} \in \{0, \ldots, n\}, \tag{51}$$

expressing how many elevators out of n go directly from $+0$ to 0 without serving any requests, and finally by adding the following equation:

$$x(+0, V) = n. \tag{52}$$

Observe that this modification has no impact on the dimension since the addition of $x_{+0,0}$ increases the dimension by one whereas the addition of (52) decreases the dimension by one as (52) is linearly independent of (47)–(49). Now the EDP polytope in x space is:

$$\begin{aligned} P_{EDP} = \mathrm{conv}\{x \in \mathbb{R}^{|A_0|+1} | x \text{ satisfies} \\ (2), (4), (5), (14), (15), (17), (19), (40), (51), \text{ and } (52)\}. \end{aligned} \tag{53}$$

Let K denote the face of P_{EDP} induced by (30), i.e., $K = \{x \in P_{EDP} | x(S, S) + x(U) + x(\sigma(S), S) = |S| - 1\}$, and let $M = (S, S) \cup U \cup (\sigma(S), S)$. Obviously, K is proper. Indeed, let $(+0, s_1, n + s_1, 0)$ and $(+0, s_2, n + s_2, 0)$ be two elevator routes which clearly are feasible and do not consist of any arc from M. Now the remaining elevator routes can use up to $|S| - 2$ arcs from M.

Let $H = V \setminus \{n, 2n\}$. By summing the out-degree equations over vertices in $H \setminus \{0\}$ and then by subtracting $x(V, n) = 1$ from the resulting equation we obtain $x(H, H) = 3n - 3$.

We prove that $\dim(K) = \dim(P_{EDP}) - 1$ by showing that (47)–(49) and (52) along with the equation

$$x(S, S) + x(U) + x(\sigma(S), S) = |S| - 1 \tag{54}$$

constitute a minimal equality system for K. This in turn is accomplished by showing that any equation $ax = a_0$, satisfied by all $x \in K$, is a linear combination of (47)–(49), (52), and (54).

Without loss of generality we can assume that $a_{ij} = 0$ for all the arcs $(i, j) \in (V, n) \cup (2n, V) \cup (V, 2n) \cup \{(n+k, n+l) \in A_0 | n+k, n+l \in D, n+k \neq 2n\} \cup \{(1, n+1)\}$. Indeed, if this is not true, we can subtract from $ax = a_0$ the equation $x(2n, V)$ weighted by $a_{2n,0}$, the equations $x(V, i) = 1$ for $i \in P \cup D \setminus \{n, 2n\}$ weighted by $a_{2n,i} - a_{2n,0}$, the equation $x(V, 2n) = 1$ weighted by $a_{n,2n}$, the equation $x(V, n) = 1$ weighted by $a_{+0,n}$, the equations $x(i, V) = 1$ for $i \in P \cup D \setminus \{n, 2n\}$ weighted by $a_{i,n} - a_{+0,n}$, the equation $x(H, H) = 3n - 3$ weighted by $(a_{1,n+1} - a_{2n,n+1} + a_{2n,0} - a_{1,n} + a_{+0,n})/(3n - 3)$, and the equations $x_{n+k,n+l} - x_{lk} = 0$ for $n+k, n+l \in D, n+k \neq 2n, (n+k, n+l) \in A_0$ weighted by $a_{n+k,n+l} - a_{2n,n+l} + a_{2n,0}$. In addition, we can use (54) to impose $a_0 = 0$.

Clearly, the resulting 'normalized' equation has the property that it is satisfied by all $x \in P_{EDP}$, and it is linearly dependent on the Eqs. (47)–(49), (52), and (54), if and only if the original equation $ax = a_0$ is.

We will next prove that after the above normalization of $ax = a_0$ we have $a_{ij} = 0$ for all i, j. Let $G'_0 = (V', A'_0)$ be the graph induced by $V' = H$. We call two arcs $(i_1, j_1), (i_2, j_2) \in A'_0 \setminus MK$*-compatible* if they are both used in a feasible m-route R' in G'_0 that contains exactly $|S| - 1$ arcs from M. This ensures that m-routes R_1 and R_2 in G_0 obtained from R' by inserting both n and $2n$ between i_1 and j_1, and between i_2 and j_2, respectively, are feasible and contain exactly $|S| - 1$ arcs from M. Therefore, the characteristic vectors of R_1 and R_2 both are in K, and hence satisfy $ax = a_0$.

There are several cases to distinguish, which we investigate one by one. For each case we provide two K-compatible arcs, of which the other is the arc being under consideration, and a feasible elevator route R' in G'_0 consisting of these two arcs and exactly $|S| - 1$ arcs from M.

1) Arc $(i, n+i), i \in P, i \neq 1, n$. Let $R' = (+0, S \cup \sigma(S), P \cup D \setminus (S \cup \sigma(S) \cup \{n, 2n\}), 0)$. Clearly, any arc $(i, n+i), i \neq 1, n$ is K-compatible with $(1, n+1)$. This implies that $a_{i,n} + a_{n,2n} + a_{2n,n+i} - a_{i,n+i} = a_{1,n} + a_{n,2n} + a_{2n,n+1} - a_{1,n+1}$, and hence $a_{i,n+i} = 0$, due to the normalization of $ax = a_0$. It then follows that $a_{i,n+i} = 0$ for $i \in P$.
2) Arc $(+0, i), i \in P, i \neq n$. If $i \in S$, let $R' = (+0, i, n+i, S \cup \sigma(S) \setminus \{i, n+i\}, P \cup D \setminus (S \cup \sigma(S) \cup \{n, 2n\}), 0)$, otherwise let $R' = (+0, i, n+i, S \cup \sigma(S), P \cup D \setminus (S \cup \sigma(S) \cup \{i, n+i, n, 2n\}), 0)$. Any arc $(+0, i), i \in P$ is clearly K-compatible with $(1, n+1)$. This implies that $a_{+0,n} + a_{n,2n} + a_{2n,i} - a_{+0,i} = a_{1,n} + a_{n,2n} + a_{2n,n+1} - a_{1,n+1}$, and hence $a_{+0,i} = 0$. Thus, $a_{+0,i} = 0$ for all $i \in P$.
3) Arc $(n+i, 0), n+i \in D, n+i \neq 2n$. If $i \in S$, let $R' = (+0, P \cup D \setminus (S \cup \sigma(S) \cup \{n, 2n\}), S \cup \sigma(S) \setminus \{i, n+i\}, i, n+i, 0)$, otherwise let $R' = (+0, S \cup \sigma(S), P \cup D \setminus (S \cup \sigma(S) \cup \{i, n+i, n, 2n\}), i, n+i, 0)$. Again, any arc $(n+i, 0), n+i \in D$ is K-compatible with $(1, n+1)$. Now it holds that $a_{n+i,n} + a_{n,2n} + a_{2n,0} - a_{n+i,0} = a_{1,n} + a_{n,2n} + a_{2n,n+1} - a_{1,n+1}$, and hence $a_{n+i,0} = 0$. Therefore, $a_{n+i,0} = 0$ for all $n+i \in D$.
4) Arc $(n+i, j) \in A_0, n+i \in D, n+i \neq 2n, j \in P, j \neq n$. Now there are four cases to distinguish further. (4.1) $i, j \notin S$. Let $R' = (+0, i, n+i, j, n+j, S \cup$

$\sigma(S), P\cup D\setminus(S\cup\sigma(S)\cup\{i,n+i,j,n+j,n,2n\}),0)$. (4.2) $i\in S, j\notin S$. Let $R'=(+0, S\cup\sigma(S)\setminus\{i,n+i\}, i, n+i, j, n+j, P\cup D\setminus(S\cup\sigma(S)\cup\{j,n+j,n,2n\}),0)$. (4.3) $i\notin S, j\in S$. Let $R'=(+0, P\cup D\setminus(S\cup\sigma(S)\cup\{i,n+i,n,2n\}), i, n+i, j, n+j, S\cup\sigma(S)\setminus\{j,n+j\},0)$. (4.4) $i,j\in S$. This case is considered in 5). In all cases (4.1)–(4.3) arc $(n+i,j)$ is K-compatible with $(1,n+1)$, which implies that $a_{n+i,n}+a_{n,2n}+a_{2n,j}-a_{n+i,j}=a_{1,n}+a_{n,2n}+a_{2n,n+1}-a_{1,n+1}$, and hence $a_{n+i,j}=0$ for each $(n+i,j)\in A_0\setminus M$.

5) Arc $(n+i,j)\in A_0, n+i\in D, j\in P, i,j\in S$. Let C be a directed cycle with vertex set $S\cup\sigma(S)$ such that none of the arcs of the form $(k,l)\in A_0, k,l\in P$ is used. For each $(n+i,j)\in C$, let $R'=(+0, P\cup D\setminus(S\cup\sigma(S)\cup\{n,2n\}), j, n+j, S\cup\sigma(S)\setminus\{j,n+j,i,n+i\}, i, n+i, 0)$. The characteristic vectors of these elevator routes are in K and thus satisfy $ax=a_0$. Since $a_{ij}=0$ for all $(i,j)\in R'\setminus(\sigma(S),S)$, for each $(n+i,j)\in(\sigma(S),S)$ we have

$$\sum_{(n+k,l)\in A_0|k,l\in P,(n+k,l)\in C\setminus\{(n+i,j)\}} a_{n+k,l}=0$$

In other words, we have a system of $|(\sigma(S),S)|$ linearly independent homogeneous equations in the $|(\sigma(S),S)|$ unknowns $a_{n+k,l}$, $(n+k,l)\in C$. This system has a unique solution that is $a_{n+k,l}=0$ for all $(n+k,l)\in C$. Because C is arbitrary, it holds $a_{n+j,i}=0$ for all $(n+j,i)\in(\sigma(S),S)$.

6) Arc $(i,j)\in A_0, i,j\in P, j\neq n$. Now there are five cases to distinguish further. The last two cases are considered in (7) and (8). (6.1) $i,j\notin S$. Let $R'=(+0,i,j,n+j,n+i,S\cup\sigma(S),P\cup D\setminus(S\cup\sigma(S)\cup\{i,n+i,j,n+j\},0)$. Arcs (i,j) and $(j,n+j)$ are K-compatible, which implies $a_{i,n}+a_{n,2n}+a_{2n,n+i}+a_{n+i,j}+a_{n+j,k}-a_{i,j}-a_{n+j,n+i}-a_{n+i,k}=a_{j,n}+a_{n,2n}+a_{2n,n+j}-a_{j,n+j}$, where k is the first pickup vertex of S visited in R', and hence $a_{i,j}=0$, due to the normalization and cases (1) and (4). (6.2) $i\in S, j\notin S$. Let $R'=(+0,S\cup\sigma(S)\setminus\{i,n+i\},i,j,n+j,n+i,P\cup D\setminus\{S\cup\sigma(S)\cup\{j,n+j\},0)$. Again, arcs (i,j) and $(j,n+j)$ are K-compatible, which implies $a_{i,n}+a_{n,2n}+a_{2n,n+i}+a_{n+i,j}+a_{n+j,k}-a_{i,j}-a_{n+j,n+i}-a_{n+i,k}=a_{j,n}+a_{n,2n}+a_{2n,n+j}-a_{j,n+j}$, where k is now the first pickup vertex of $P\setminus(S\cup\{j,n\})$ visited in R', and hence $a_{i,j}=0$. (6.3) $i\notin S, j=s_1$. Let $R'=(+0,i,s_1,s_2,\ldots,s_r,n+s_r,n+s_{r-1},\ldots,n+s_1,n+i,P\cup D\setminus\{S\cup\sigma(S)\cup\{i,n+i\},0)$. In this case, arcs (i,j) and $(k,n+k)$ are K-compatible where k is the first pickup vertex of $P\setminus(S\cup\{i,n\})$ visited in R', which again implies $a_{i,n}+a_{n,2n}+a_{2n,n+i}+a_{n+i,j}+a_{n+j,k}-a_{i,j}-a_{n+j,n+i}-a_{n+i,k}=a_{j,n}+a_{n,2n}+a_{2n,n+j}-a_{j,n+j}$, and hence $a_{i,j}=0$.

7) Arc $(i,j)\in A_0, i,j\in P, (i,j)\in U$. Let $R'=(+0,i,j,n+j,n+i,S\cup\sigma(S)\setminus\{j,n+j\},P\cup D\setminus\{i,n+i,n,2n\})$. Since (i,j) is the only arc of R' that belongs to U, the characteristic vector of R' satisfies $ax=a_{ij}$, from which we get $a_{i,j}=0$.

8) Arc $(i,j)\in A_0, i,j\in P, (i,j)\in(S,S)$. Let $R'=(+0,i,j,n+j,n+i,S\cup\sigma(S)\setminus\{i,n+i,j,n+j\},P\cup D\setminus\{n,2n\})$. Since (i,j) is the only arc of R' that belongs to (S,S), the characteristic vector of R' satisfies $ax=a_{ij}$, from which we obtain $a_{i,j}=0$.

This completes the proof. We have shown that any equation $ax = a_0$, satisfied by all $x \in K$, is a linear combination of (47)–(49), (52), and (54), and hence (30) defines a facet of P_{EDP}. □

The hypothesis of Theorem 8 that $n \notin S$ is not restrictive. One can always introduce a dummy request $n+1$ and a set of arcs incident to corresponding vertices with appropriate weights such that the new problem is equivalent to the old one.

Theorem 9 *If $i, j \in P$ and $(i, j) \in A_0$, then logical inequalities are not facet defining.*

Proof Now $p = n+j, r = n+i, i < j$, and $U_j^+ = U_p^- = U_i^- = \emptyset$. Consider first set U_p^+. With this set, the logical inequality is $x_{ij} + \sum_{(k,l)\in(n+j,V)\setminus(n+j,n+i)} x_{kl} \leq 1 \Leftrightarrow x(n+j, V) \leq 1$, and therefore is not proper. Similarly, logical inequality is not proper with set U_r^-. Consider next set U_r^+. With this set, logical inequality is $x_{ij} + x_{n+i,j} \leq 1$ which is dominated by the in-degree equation of vertex j. Thus logical inequalities are not facet defining for $i, j \in P$. □

Theorem 10 *If $i \in P, n+j \in D$, and $(i, n+j) \in A_0$, then logical inequalities are not facet defining, except when $i = n-1$ and $j = 2n-1$.*

Proof The only cases where $(i, n+j) \in A_0, i \in P, n+j \in D$ are when $j = i$. This means that all sets $U_j^+, U_i^-, U_{n+j}^+, U_{n+i}^+, U_{n+i}^-$, and U_j^- are empty. Therefore, now logical inequalities correspond to the upper bound constraints of the routing variables, and the only case when they are facet defining is when $i = n-1$ and $j = 2n-1$. □

Theorem 11 *If $n+i, n+j \in D$ and $(n+i, n+j) \in A_0$, then logical inequalities are not facet defining.*

Proof Due to the similar reasoning as in the proof of Theorem 9, logical inequalities are not facet defining for $n+i, n+j \in D$. □

Theorem 12 *If $j \in P, n+i \in D$, and $(n+i, j) \in A_0$, then logical inequalities are not facet defining.*

Proof Observe that $K_{n+i,j} = \emptyset$. Therefore, now logical inequalities are equal to the upper bound constraints of the routing variables, which are not facet defining for $j \in P, n+i \in D$. □

5 Concluding remarks and further research

Controlling an elevator group is a dynamic problem since the transportation requests are gradually revealed throughout time. Thus, the common practice is that the static EDP with up-to-date information is solved repeatedly, either after a certain amount of time or after the registration of a new request. There are many papers studying approaches to solve the EDP. None of them, however, has introduced an explicit formulation of the problem and therefore the polyhedral properties of the EDP have not been studied.

In this paper, we gave a mixed-integer linear programming formulation of the EDP in which all the relevant constraints were given mathematically. The formulation was initially defined on an ”almost complete” graph. We then introduced a set of rules, which can be used to eliminate redundant arcs. These rules were tested on a small but realistic example (two elevators and five requests) and they seemed to be very effective, in the the example 52 arcs out of 82 were eliminated. We also investigated a couple of other techniques that may strengthen the formulation. We showed that load variables can be eliminated by rewriting all constraints that include them in routing variables, which can be considered as main decision variables. We proved that the disaggregation of time variables results in formulation that is at least as strong as the original formulation. Our initial tests, which will be shown in a forthcoming paper, indicate that the disaggregation of time variables has a significant impact on the linear programming relaxation value. The disadvantage of this approach is that it increases the size of the problem since now for each arc there is associated a time variable. We introduced the lifted subtour elimination constraint and the strengthened precedence constraints. Furthermore, we gave two new families of valid inequalities for the EDP that are based on logical implications.

In addition, we analyzed the polyhedral structure of the EDP in a generic case arising in down-peak. More specifically, we provided the expression for the number of feasible solutions, determined the size of the EDP polytope, identified which equalities form a minimal equation system for it, and showed which of the derived valid inequalities are facet defining. We also studied for which arcs the simple upper and lower bounds constraints of the routing variables are facet inducing.

Future work will concentrate on the polyhedral analysis for other traffic patterns and the development of real-time algorithms for the EDP utilizing the findings presented in this paper.

Acknowledgments This work was supported by the Research Foundation of Helsinki University of Technology. Some results were obtained by the help of Kimmo Berg. These supports are gratefully acknowledged.

References

Archetti C, Savelsbergh MWP, Speranza MG (2008) To split or not to split: that is the question. Transp Res Part E 44:114–123

Ascheuer N, Fischetti M, Grötschel M (2000) A polyhedral study of the asymmetric traveling salesman problem with time windows. Networks 36(2):69–79

Balas E, Fischetti M, Pulleyblank WR (1995) The precedence-constrained asymmetric traveling salesman polytope. Math Program 68:241–265

Carrabs F, Cerulli R, Cordeau JF (2007) An additive branch-and-bound algorithm for the pickup and delivery traveling salesman problem with lifo or fifo loading. Inf Syst Oper Res 45:223–238

Dantzig GB, Fulkerson DR, Johnson SM (1954) Solution of a large scale traveling salesman problem. Oper Res 2:393–410

Desrochers M, Desrosiers J, Solomon M (1992) A new optimization algorithm for the vehicle routing problem with time windows. Oper Res 40:342–354

Hiller B (2009) Online optimization: probabilistic analysis and algorithm engineering. PhD thesis, TU Berlin

Hiller B, Tuchscherer A (2008) Real-time destination-call elevator group control on embedded microcontrollers. In: Kalcsics J, Nickel S (eds) Operations research proceedings, pp 357–362

Hirasawa K, Eguchi T, Zhou J, Yu L, Hu J, Markon S (2008) A double-deck elevator group supervisory control system using genetic network programming. IEEE Trans Syst Man Cybern C Appl Rev 38(4):535–550

Ho M, Robertson B (1994) Elevator group supervisory control using fuzzy logic. In: Conference proceedings on electrical and computer engineering, pp 825–828

Ikeda K, Suzuki H, Markon S, Kita H (2006) Evolutionary optimization of a controller for multi-car elevators. In: IEEE international conference on industrial technology. ICIT 2006, pp 2474–2479

Inamoto T, Tamaki H, Murao H, Kitamura S (2003) Deterministic optimization model of elevetor operation problems and an application of branch-and-bound method. IEEJ Trans Electron Inf Syst 123:1334–1340

Koehler J, Ottiger D (2002) An AI-based approach to destination control in elevators. AI Mag 23:59–78

Levy D, Yadin M, Alexandrovits A (1977) Optimal control of elevators. Int J Syst Sci 8(3):310–320

Liu J, Liu Y (2007) Ant colony algorithm and fuzzy neural network-based intelligent dispatching algorithm of an elevator group control system. In: IEEE international conference on control and automation. ICCA 2007, pp 2306–2310

Luh PB, Xiong B, Chang SC (2008) Group elevator scheduling with advance information for normal and emergency modes. IEEE Trans Autom Sci Eng 5(2):245–258

Markon S, Kise H, Kita H, Bartz-Beielstein T (2006) Elevator group control by neural networks and stochastic approximation. In: Control of traffic systems in buildings. Springer, London, pp 163–186

Naddef D, Rinaldi G (2002) Branch-and-cut algorithms for the capacitated VRP. In: Toth P, Vigo D (eds) The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp 53–81

Pepyne D, Cassandras C (1997) Optimal dispatching control for elevator systems during uppeak traffic. IEEE Trans Control Syst Technol 5(6):629–643

Ropke S, Cordeau JF (2009) Branch and cut and price for the pickup and delivery problem with time windows. Transp Sci 43:267–286

Roschier NR, Kaakinen M (1979) New formulae for elevator round trip calculation. Supplement to Elevator World of ACIST Members, pp 189–197

Ruland KS, Rodin EY (1997) The pickup and delivery problem: faces and branch-and-cut algorithm. Comput Math Appl 33(12):1–13

Schröder J (1990) Advanced dispatching. Elevator World 40

Siikonen ML (1997) Elevator group control with artificial intelligence. Technical Report Systems Analysis Laboratory A67. Helsinki University of Technology

Sorsa J, Siikonen ML, Ehtamo H (2003) Optimal control of double-deck elevator group using genetic algorithm. Int Trans Oper Res 10:103–114

Tanaka S, Uraguchi Y, Araki M (2005a) Dynamic optimization of the operation of single-car elevator systems with destination hall call registration: part I formulation and simulations. Eur J Oper Res 167:550–573

Tanaka S, Uraguchi Y, Araki M (2005b) Dynamic optimization of the operation of single-car elevator systems with destination hall call registration: part II the solution algorithm. Eur J Oper Res 167:574–587

Tyni T, Ylinen J (2001) Genetic algorithms in elevator car routing problem. In: Spector L et al (eds) Proceedings of the genetic and evolutionary conference (GECCO-2001). Morgan Kaufman Publishers, San Francisco, pp 1413–1422

van Eijl C (1995) A polyhedral approach to the delivery man problem. Technical Report Memorandum COSOR. Eindhoven University of Technology, The Netherlands, pp 95–19

Yu L, Mabu S, Zhang T, Eto S, Hirasawa K (2009) Multi-car elevator group supervisory control system using genetic network programming. In: IEEE congress on evolutionary computation. CEC ’09, pp 2188–2193

Yu L, Mabu S, Hirasawa K (2010) Multi-car elevator system using genetic network programming for high-rise building. In: IEEE international conference on systems man and cybernetics (SMC), pp 1216–1222