Static Elevator Dispatching Problem with Destination Control

Branch-and-Price

Fabian Torres

Michel Bierlaire

STRC conference paper 2023 **April 14, 2023**

STRC | **23rd Swiss Transport Research Conference**
Monte Verità / Ascona, May 10-12, 2023

Static Elevator Dispatching Problem with Destination Control

Fabian Torres
TRANSP-OR
EPFL Lausanne
fabian.torres@epfl.ch

Michel Bierlaire
TRANSP-OR
EPFL Lausanne
michel.bierlaire@epfl.ch

April 14, 2023

Abstract

The elevator dispatching problem with destination control (EDPDC) is a complex scheduling problem that requires efficient allocation of elevators to handle passenger demand in high-rise buildings. Traditionally, passengers enter the elevator and select their destination floor. With destination control, passengers make a call in the lobby specifying their destination floor. Afterwards, the interface assigns an elevator to the passengers. Destination control provides additional information about destinations that can be used to allocate elevators optimally.

In this paper, we consider the EDPDC as static and deterministic, where all the information about passengers, arrival times, origin and destination floors is known beforehand. This is a relaxation of the dynamic and stochastic case where this information is uncertain. Hence, this formulation provides a lower bound on the dynamic problem.

We present a set partitioning formulation and propose a branch-and-price algorithm to solve the EDPDC. To evaluate the performance of the approach, we conducted experiments on small instances of the EDPDC with up to 3 elevators, 8 floors, and a maximum of 33 passengers. The results show that the branch-and-price algorithm was able to solve these instances within 5 hours of running time, which is a promising first step towards solving larger instances of the EDPDC. We suggest future research directions, including exploring different branching rules, finding relaxations of the pricing problem, and incorporating uncertainty in passenger demand.

Keywords

Elevator dispatching, destination control, branch-and-price, scheduling, labeling algorithms, column generation, group control, combinatorial optimization

Contents

List of Tables

1 Introduction

The Burj Khalifa stands at 828 meters tall, with a total of 57 high speed elevators. Large buildings allow the use of vertical space without the need to expand and develop more land. In addition, the world population continues to concentrate on large mega-cities (i.e., cities with more than 10 million inhabitants) where the population density (i.e., the population per square kilometer) continues to rise (Kraas and Coy, 2016). With the increasing number of tall buildings, the need for efficient vertical transportation systems becomes more important. Early elevators were designed to have an elevator operator (i.e., a human being) inside the elevator that would take the passengers to their desired floors, similar to a bus driver that takes passengers to their destinations by driving the bus (Bernard, 2014). Today, most elevators still have the floor controls inside, i.e., you have to enter the elevator first before you can select your floor. However, destination control (DC) is a system that allows passengers to choose their destination floor before they get inside the elevator, within seconds they get assigned an elevator that will take them to their destination. Destination control allows more information to be known by the controller, and thus, a more efficient dispatching of elevators is possible.

In this paper, we consider the Elevator Dispatching Problem with Destination Control (EDPDC). Elevators must be assigned to arriving passengers and must be routed to minimize the average passenger journey time. While the EDPDC is naturally a highly dynamic and stochastic problem, we consider the static-deterministic problem where all the inputs are known in advance, i.e., passenger arrival times, service times and destinations. The static-deterministic variant is useful for evaluating the performance of stochastic multi-stage algorithms.

The static EDPDC has received little attention in the scientific literature. The EDPDC is a variant of the more studied pickup-and-delivery problem with time windows (PDPTW), where passengers must be picked-up at a location and delivered at their destination within a specific time window. The most successful exact solution methods for routing problems with time windows has been branch-and-price algorithms (e.g., Ropke and Cordeau (2009), Torres *et al.* (2022b), Torres *et al.* (2022a)). Although branch-and-price algorithms are usually used to solve the PDPTW, we have not been able to find any branch-and-price algorithm for the EDP with destination control in the literature. In fact, due to the complexity of the problem and the restriction of movement of elevators, the literature has focused mostly in metaheuristics (e.g., Sorsa *et al.* (2018)).

In Ruokokoski *et al.* (2016) an assignment formulation for the EDP with destination control is introduced. The problem considers the static-deterministic off-line problem where there is a set of passengers already in the elevators, another set of passengers that have already been assigned to elevators and a set of passengers that needs to be assigned to elevators. A commercial solver is used to solve the model and only small instances with at most 12 unassigned requests, 4 elevators and 22 floors. Nevertheless, the model fails to find a relation between time and decision variables that allows elevators to bypass some requests due to maximum elevator capacity. To over come this modeling issue, the authors restrict the problem to cases where an elevator route length cannot exceed one round trip. This limits the applicability of the model and it potentially excludes feasible solutions.

The remainder of this paper is divided as follows: In Section 2, we provide a description of the static-deterministic problem and present a new set partitioning formulation. In Section 3, we model the previously described problem. In Section 3, we propose a branch-and-price algorithm to find optimal solutions. In Section 4, we present some preliminary results on small instances that show promise for the proposed method. Finally, in Section 5 we present final comments and future research directions.

2 Problem Description

Let V be the set of all vertices, let P be the set of all pickup requests and let D be the set of delivery locations, i.e., $P \cup D = V$. For each pickup request $p \in P$ there is a delivery location $p + n \in D$, where the value n is the total number of pickup requests, i.e., $n = |P|$. Each pickup request has an earliest pickup time, i.e., e_p, and a latest pickup time, i.e., l_p, for all $p \in P$. The earliest pickup time is equal to the moment the passenger makes the call in the lobby by selecting the desired destination floor. The latest pickup time represents some threshold time at which it is too long for a passenger to wait for an elevator, e.g., $l_p - e_p = 90$ seconds. The time it takes to pickup a passenger once the door opens is the service time at pickup, i.e., s_p for $p \in P$ and the total time to deliver once the doors open is s_d for $d \in D$.

Let F be the set of floors in the building and let A be the set of all arcs connecting each floor, i.e., $(i, j) \in A$ and $i, j \in F$. Let E be the set of homogeneous elevators and Q the capacity of each.

In addition, to improve customer experience while ridding the elevator, the following rules have to be respected when there is at least one passenger in the elevator:

- The elevator cannot change direction;
- The elevator cannot stop at a floor where no one enters or exits;
- A passenger cannot enter an elevator with passengers that are going in the opposite direction to her destination floor.

2.1 Formulation

Let Ω be a set with all possible elevator pickup-and-delivery routes and let λ_r for all $r \in \Omega$ be a binary variable that equals to one if route $r \in \Omega$ is part of the optimal solution. The cost of a route, i.e., c_r for $r \in \Omega$, is equal to a weighted sum of the average waiting time of all passengers in the route and the total energy consumption of the elevator. Let $a_{i,r}$ be a binary parameter that indicates if passenger $i \in P$ is picked up in route $r \in \Omega$. The set partitioning formulation is as follows:

$$
\begin{aligned}
\min \; & \sum_{r \in \Omega} c_r \lambda_r & & \qquad (1) \\
\text{s.t.} \; & \sum_{r \in \Omega} a_i^r \lambda^r = 1 & \forall i \in P & \qquad (2) \\
& \sum_{r \in \Omega'} \lambda_r \leq |E| & & \qquad (3) \\
& \lambda_r \in \{0, 1\} & \forall r \in \Omega &
\end{aligned}
$$

The objective (1) is to minimize a weighted sum of the average journey time of all passengers. Constraints (2) ensures that all passengers are picked up by exactly one elevator. Finally, constraint (3) bounds above the total number of routes to the available elevators.

3 Branch-and-Price

The exponential number of possible routes makes model(1)-(3), i.e., the master problem (MP), rapidly intractable for even some small instances of the problem. Hence, Branch-and-Price (BP) is a solution method that allows us to solve the model (1)-(3) by considering a series of pricing sub-problems that generate routes. We consider a restricted version of the MP called the restricted master problem (RMP) that only has a few number of feasible routes. Unlike the master problem, the RMP is rapidly solved by any commercial solver(e.g., CPLEX). With a solution of the RMP we can find routes in the pricing problem by using the dual variables fromt the linear relaxation of the RMP.

3.1 Pricing

The pricing problems are called Elementary Shortest Path Problem with Resource Constraints (ESPPRC). Labeling algorithms that use dynamic programming are considered the state-of-the-art to solve these types of problems. Let $\mathcal{L} = (f, l, t, c, d, \bar{D}, \bar{P}, \bar{R})$ be a label that represents a partial path (or state) of an elevator. Each label has a current floor, i.e., f, a total load, i.e., l, a time, i.e., t, a total cost, i.e., c, a direction, up or down, i.e., d, a set of destination floors, i.e., $\bar{D}$, a set of passengers that have been picked up, i.e., $\bar{P}$, and a set of passengers that are currently inside the elevator, i.e., $\bar{R}$. The cost is a weighted sum of the average journey time of passengers. The following equation defines the reduced cost:

$$c = \sum_{i \in \bar{P}} t_i / |P| - \mu \tag{4}$$

The time t_i is the time passenger $i \in \bar{P}$ has been waiting and μ is the value of the dual variables in the current solution of the RMP.

We start with an empty label and extend to all possible pickup requests. The resources are calculated with Resource Extension Functions (REFs), e.g., if extending from pickup $i \in P$ to pickup $j \in P$, we can calculate the cost by adding the time passengers travel in the elevator to go from the origin of i to the origin of j. Similarly, all resources are tracked and extended from one node to the next. At each extension we must check for feasibility to be sure that unfeasible solutions are not created. For example, the elevator cannot change directions while the load is more than zero, thus, an extension to pickup a passenger that is going in the opposite direction is not feasible. Feasibility checks at each extension can guarantee that all the constraints are not violated.

3.2 Dominance rules

The labels created in the labeling algorithm grow exponentially with the number of possible combination of pick up and delivery requests. To reduce the number of labels, dominance rules are used to eliminate labels that are not going to lead to the optimal solution. Label $\mathcal{L}_1$ dominates label $\mathcal{L}_2$ if the following conditions are met:

1. $f_1 = f_2$
2. $d_1 = d_2$
3. $t_1 = t_2$
4. $l_1 \leq l_2$
5. $c_1 \leq c_2$
6. $D_1 \subseteq D_2$
7. $P_1 \subseteq P_2$
8. $R_1 \subseteq R_2$

Rules (6)-(8) lead to the exponential growth of labels. If these rules are eliminated, a relaxation of the problem is obtained where the labels are not increasing exponentially, however, passenger can potentially be visited several times, producing cycles in the solution. Cycles can then be eliminated gradually through branching.

3.3 Branching

In a Branch-and-Price algorithm the branching has to be done carefully. If we branch on the fractional variables of the RMP, the pricing problem will become intractable rapidly as setting a variable to zero will lead to a pricing problem with forbidden paths.

In this initial version of the algorithm we branch on the total flow between two nodes in the graph, i.e., between pickup nodes or delivery nodes. This branching strategy does not increase the complexity of the pricing problem, and it can be easily applied. However, it is usually not the best method to use and it is generally used as a last resort (e.g., Torres *et al.* (2022b)).

4 Preliminary results

To test the proposed framework, we run computational experiments in an instance with mixed traffic flow of passengers. In a mixed traffic flow instance, some passengers are entering the building, thus their origin floor is 0, while other passengers are exiting the building, thus, their destination floor is 0, the remaining passengers are traveling inside the building, i.e., neither the origin nor the destination floors are floor 0.

The building has 8 total floors and 3 elevators with a capacity of 13 passengers each.

Table 1: Results on small instances

Passengers	CPU	Ave. Destination time
9	0.3	32.77
13	0.9	35.23
15	0.8	35.7
17	1.3	36.98
21	48.1	38.76
25	755.2	38.96
30	3630.3	39.94
33	17290.0	40.66

5 Conclusions and Future Research

We formulated the static Elevator Dispatching Problem with a set partitioning formulation and developed a branch-and-price algorithm to solve the problem. The preliminary results show that using the proposed method out preforms the assignment formulation by solving larger instances.

Future research directions can be to explore different branching strategies to improve the performance of the algorithm. Branching has to be done carefully to avoid increasing the complexity of the pricing problem.

More efficient dominance rules could be developed. Dominance rules are the main actor in preventing the exponential growth of labels. Better dominance rules would improve the algorithm making the solution of larger problems more likely.

Lastly, extending this research to the dynamic case, where the arrival of passengers is uncertain, can be an interesting avenue for research. Adapting the proposed branch-and-price algorithm to dynamic instances could be challenging.

6 References

Bernard, A. (2014) *Lifted: A cultural history of the elevator*, NYU Press.

Kraas, F. and M. Coy (2016) *Megacities*, Springer.

Ropke, S. and J.-F. Cordeau (2009) Branch and cut and price for the pickup and delivery problem with time windows, *Transportation Science*, **43** (3) 267–286.

Ruokokoski, M., J. Sorsa, M.-L. Siikonen and H. Ehtamo (2016) Assignment formulation for the elevator dispatching problem with destination control and its performance analysis, *European Journal of Operational Research*, **252** (2) 397–406.

Sorsa, J., H. Ehtamo, J.-M. Kuusinen, M. Ruokokoski and M.-L. Siikonen (2018) Modeling uncertain passenger arrivals in the elevator dispatching problem with destination control, *Optimization Letters*, **12**, 171–185.

Torres, F., M. Gendreau and W. Rei (2022a) Crowdshipping: An open vrp variant with stochastic destinations, *Transportation Research Part C: Emerging Technologies*, **140**, 103677.

Torres, F., M. Gendreau and W. Rei (2022b) Vehicle routing with stochastic supply of crowd vehicles and time windows, *Transportation Science*, **56** (3) 631–653.