Multiple objectives and system constraints in double-deck elevator dispatching

Janne Sorsa, Mirko Ruokokoski
KONE Corporation
FI-02150 Espoo, Finland
Email: Janne.Sorsa@kone.com, Mirko.Ruokokoski@kone.com

***Abstract*—The elevator group control dispatches elevators to serve passenger calls. The dispatching decisions are determined by solving the Elevator Dispatching Problem (EDP). Passengers enter their destination calls with the destination control already in the lobby. In turn, the group control assigns the serving elevator immediately and announces it to the passenger. A double-deck elevator consists of two elevator cars that are attached on top of each other thus serving two adjacent floors simultaneously. In the existing double-deck destination controls, both the serving elevator and deck are fixed at once after the passenger call and not changed afterwards. In this paper, the effect of these restrictions on the solutions of the double-deck elevator dispatching problem are studied by depicting its objective space with two conflicting objectives, namely, passenger waiting and transit time. This paper aims at characterizing the relationship between passenger waiting time in the lobby and transit time inside the car as optimization objectives.**

I. INTRODUCTION

The main task of an elevator group control is to dispatch the elevators under its control to passenger calls. The group control optimizes dispatching decisions by solving the Elevator Dispatching Problem (EDP). Nowadays all major elevator companies offer an advanced elevator product, *Destination Control* (DC), in which passengers enter their destination calls already in the lobby using a numeric keypad. After registering the call, the group control immediately assigns the elevator to serve the call and announces it on the screen associated with the keypad. All the existing DC products are designed according to this principle.

A double-deck elevator is a special elevator consisting of two elevator cars. The cars are attached one on top of the other and move as a single unit, thus serving adjacent floors at the same time. To allow simultaneous service of the decks on the ground floor, the lobby is arranged on two floors which are interconnected by escalators [1]. In this arrangement, the lower deck serves only the lower lobby level, while the upper deck serves only the upper lobby level. However, both decks can serve all the upper floors, the topmost floor being a possible exception to the rule if the building does not have adequate space above it for the upper deck.

Modern double-deck group controls can minimize passenger waiting and journey times as optimization objectives of the Double-Deck Elevator Dispatching Problem (DD-EDP) [2][3]. However, these objectives result in different solutions even in simple situations, which raises the foreseeability of conflicting objectives. Therefore, the DD-EDP is put here in the context of multi-objective optimization, where passenger waiting and transit times are considered simultaneously. Transit time is considered as the second objective instead of journey time because journey time is about the same as waiting plus transit time [4]. Multi-objective optimization has not been considered earlier for different passenger service objectives, but balancing of passenger service and energy consumption has been realized [5][6].

The existing double-deck DC assigns both the serving elevator and deck immediately after the passenger call is registered [3]. Recently, a new approach to DC has been proposed, which postpones the final elevator assignment to a later time. The final moment for fixing the assignment can be, for example, when the elevator starts to decelerate to the call floor. Such a product would require a different user interface and signalization concept compared to existing products. The delayed assignment does not affect the structure of the DD-EDP but only the complexity of the actual problem instances. In this paper, objective spaces of small and moderately large DD-EDP instances are studied by relaxing the system constraints gradually. The cases under consideration are:

1) the existing DC, which considers all the existing calls as constraints and makes the immediate assignment of both the elevator and the deck only once for a newly registered call;
2) *Semi-Continuous* (SC) DC, which makes the final elevator shaft assignment immediately but allows the change of the serving deck until a defined distance from the call floor;
3) *Fully-Continuous* (FC) DC, which allows the change of both the serving elevator and the deck until a defined distance from the call floor.

In the above terms, “continuous” refers to the gradual change of the DC, which traditionally operates according to the immediate assignment policy, towards the continuous assignment policy, which is typical of the conventional control with up and down call buttons as call-giving devices. Since DC is generally less efficient in mixed lunch-hour traffic than in morning up-peak [7], the delayed elevator assignment is expected to reduce passenger service times during that time. Therefore, the example instances in this paper are taken from lunch-hour traffic simulations conducted with the Building Traffic Simulator (KONE BTS^{TM}) [8]. The instances are solved in an off-line environment with the genetic algorithm [9] adapted to double-deck elevators [2][3].

II. DD-EDP Objective Functions

The DD-EDP can be decomposed into a generalized assignment problem and a set of elevator routing problems following the well-known decomposition of the vehicle routing problem. In the case of the DD-EDP, a deck of an elevator corresponds to a vehicle and the main decision variables are deck to passenger call assignments. The set S consists of all the origin floors of the destination calls requiring pick-up, and the set T contains all the destination floors of the passengers travelling inside the elevators as well as the passengers still waiting for pick-up. The waiting time of a passenger $i \in S$ is simply the sum of the estimated arrival time of an elevator, t_i, and the time elapsed since the registration of the call, γ_i. The transit time of a passenger $i \in T$ is then the difference between the elevator arrival times to the destination and the origin floor.

Each call is also associated with the number of passengers requiring service, ω_i, which is assumed to equal one in this paper. However, in the real DC, passengers can give a group call in which they define the number of travelling passengers. The number of passengers can also be estimated from traffic statistics collected by the group control [10].

The objective functions of the DD-EDP are f_1 to minimize passenger waiting time and f_2 to minimize passenger transit time,

$$f_1(S,T) = \sum_{i \in S} (\gamma_i + t_i)\,\omega_i, \tag{1}$$

$$f_2(S,T) = \sum_{i \in T} (t_i - t_j)\,\omega_i, \tag{2}$$

where $j \in S$ is the known origin floor of the destination call and $i \in T$ is the destination floor of the destination call. For passengers already travelling inside the car, t_j is the actual time of pick-up as recorded by the group control.

III. Analysis of Two DD-EDP Instances

Two instances of the DD-EDP are analyzed to describe its characteristics as a multi-objective optimization problem. To get a realistic initial state and calls for the instances, an elevator group consisting of five double-deck elevators was simulated with the BTS for a building with 19 floors including a two-floor lobby on the ground floor. The initial states of the instances occurred during lunch-hour traffic simulations, in which the passenger arrival rates were 6% and 12% of the building population in five minutes, or 95 and 190 persons in absolute terms. The simulated traffic was split to incoming, outgoing, and inter-floor traffic components with proportions 40%, 40% and 20%, respectively. During the simulations, the locations, loads and other state variables of the elevators as well as the passenger calls were recorded into log-files for all solved DD-EDP instances. A single instance from both simulations was chosen to represent a small instance and a moderately large instance. The small instance included 10 passenger calls and the larger one 30 passenger calls.

A. Off-line Environment to Solve the Instances

A custom-built software was used to read the initial state and calls of an instance from the log-files. It then modified the DD-EDP system constraints and solved the problem off-line by the same genetic algorithm as used in the actual elevator product [9][2][3]. However, in this off-line study, the number of chromosomes were increased to 400 and the same value was used for the maximum number of generations. With these parameters, the genetic algorithm produces many more iterations and unique chromosomes than the actual real-time algorithm which gives a better idea of the whole objective space.

TABLE I. Total number of alternative and evaluated chromosomes

	Instance with 10 passengers		Instance with 30 passengers	
	Alternatives	Evaluated	Alternatives	Evaluated
DC	10	10	10	10
SC	320	297	$1.31 \cdot 10^6$	7802
FC	$6.25 \cdot 10^8$	33072	$2.44 \cdot 10^{26}$	42737

The used genetic algorithm optimizes only a single objective. However, it was originally developed to be capable of evaluating two or more objective functions and then calculating the final fitness as a weighted sum over the objective function values. Thus, the DD-EDP can be formulated as a scalarized multi-objective optimization problem with weight factor w,

$$\min w f_1(S,T) + (1-w)\, f_2(S,T). \tag{3}$$

For this study, the individual objective function values were stored to analyze the objective space afterwards. It turned out that the weighing had a strong influence on the areas where the genetic algorithm solutions converged. Therefore, instances with a large number of possible solutions, namely, the ones with both SC and FC, were solved several times with different weight factors, $w \in \{0, 0.25, 0.5, 0.75, 1\}$.

B. DD-EDP Complexity

The complexity of the DD-EDP varies a lot depending on the system constraints. Theoretically, the total number of solution alternatives equals $(2E)^C$ where E denotes the number of elevators, C the number of calls subject to optimization, and 2 stands for the two decks of the elevator. However, in the case of DC, usually $C = 1$, but in the case of FC, C equals the total number of non-served passenger calls. The case of SC differs slightly from the others since it contains usually only one new call which can be served by all elevators and $C-1$ earlier calls for which the serving deck can be changed. Therefore, the number of solutions for SC equals $2E \cdot 2^{C-1}$. Table I shows how quickly the number of alternative solutions increases depending on the number of passenger calls and the system constraints.

The maximum number of calls at any given time is restricted by elevator capacity since the DC is capable of planning elevator routes for 1.5 roundtrips ahead of their current location. Let θ denote the maximum number of calls that can be assigned to one deck at a time. Then, group control can assign up to 2θ calls for a single elevator during one trip upwards or downwards. By combining these for the whole group, the maximum number of calls is estimated at $C_{\max} = 6E\theta$. For the elevator group used in these examples $C_{\max}$ equals 420 if 14 passenger calls is the momentary maximum along the elevator routes.

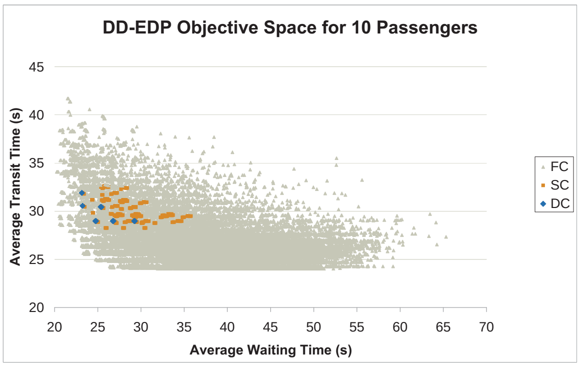

Fig. 1. DD-EDP solutions in the objective space

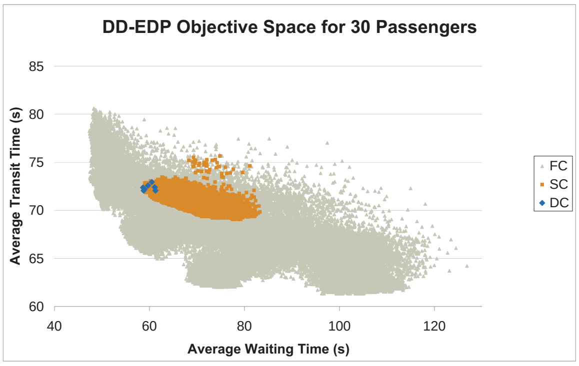

Fig. 2. DD-EDP solutions in the objective space

C. DD-EDP Objective Space

Figures 1 and 2 show the evaluated solutions of the small and the large instance, and for DC, SC and FC, respectively. In the figures, each data-point corresponds to one solution to the problem instance and it depicts average passenger transit time as a function of average passenger waiting time. The solutions for SC and FC clearly exhibit a typical form for conflicting objectives since the shape of the Pareto-optimal front is visible. Since DC has only a few alternative solutions, the relationship is not well visible but seems similar to the others.

Figures 3 and 4 show the respective Pareto-optimal fronts for the small and the large problem. The Pareto front of the large instance seems to be non-continuous at some specific areas of the objective space for FC. The gaps result from the chosen weight factors that make the search to converge on the areas close to the gaps but not reaching them. The DC has only three Pareto-optimal solutions to the small instance and two for the large. This occurs probably because of the many constraints posed by the existing passengers on the service of the new one. The Pareto-optimal solutions of SC are slightly better than the solutions of the DC as there are additional possibilities to change the serving deck. According to the figures, FC promises a clearly better service level compared to DC and SC: the freedom of choice improves the minimum waiting and transit times up to 15%. However, the improvement only reflects the estimation inside the group control and its optimization model. The result can only be interpreted as an indication, and not as an actual improvement in passenger service level.

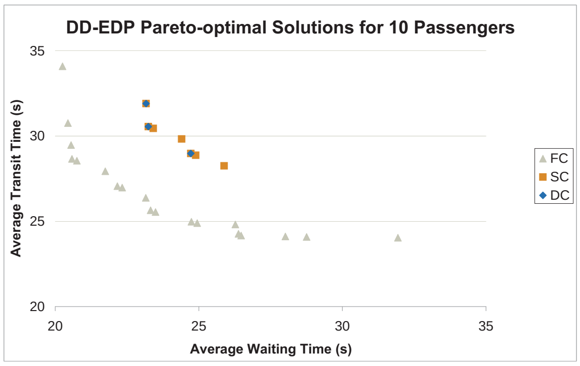

Fig. 3. DD-EDP Pareto-optimal solutions for 10 passengers

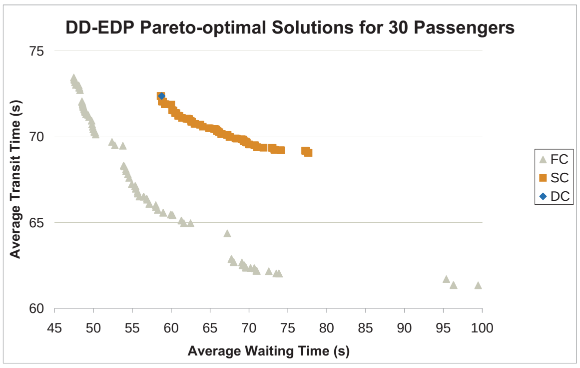

Fig. 4. DD-EDP Pareto-optimal solutions for 30 passengers

TABLE II. PROPORTION OF EVALUATED SOLUTIONS INSIDE THE BOX BOUNDED BY IDEAL AND NADIR VECTORS, 10 PASSENGERS

	Ideal		Nadir		Inside the box
	f_1	f_2	f_1	f_2	
DC	23.16	28.98	24.72	31.91	30.0%
SC	23.16	28.26	25.88	31.91	12.8%
FC	20.25	24.04	31.93	34.09	19.0%

TABLE III. PROPORTION OF EVALUATED SOLUTIONS INSIDE THE BOX BOUNDED BY IDEAL AND NADIR VECTORS, 30 PASSENGERS

	Ideal		Nadir		Inside the box
	f_1	f_2	f_1	f_2	
DC	58.62	72.06	61.28	72.37	50.0%
SC	58.62	69.05	77.68	72.37	94.4%
FC	47.37	61.34	99.47	78.29	79.9%

D. DD-EDP Pareto-Optimal Region

The Pareto-optimal region is a box in the objective space which is bounded by the ideal and the nadir objective vectors [11]. The ideal objective vector defines the lower left corner of the box as the minimum values of both of the objectives. The nadir point lies in the opposite corner and can in practice be defined by the values of the other than the minimum component of the ideal vector. These special objective vectors are shown in Tables II and III for the small and the large instance, respectively. As can be expected, the volumes of the Pareto-optimal regions are small for the DC, a bit wider for the SC, and the largest for the FC.

The tables also show the percentage of the evaluated solutions that lie in the Pareto-optimal region. In the case of DC, the percentage is not statistically significant because there are only 10 solutions to the problem instances. In the case of the small instance, the percentages are rather low due to tight boundaries of the Pareto-optimal region. With the large problem instance, the percentages are high but also the Pareto-optimal regions are quite large, especially for the FC. The genetic algorithm does unnecessary work when evaluating the solutions outside the Pareto-optimal region. In addition, since the volume of the Pareto-optimal region as well as the amount of useless processing varies, the algorithm should be capable of dynamically recognizing the iterations in the areas far from the Pareto front and adapting its search direction accordingly.

IV. Conclusion

In this paper, the Double-Deck Elevator Dispatching Problem (DD-EDP) was described as a multi-objective optimization problem. Generally, passenger waiting and transit times in the elevator system are conflicting objectives. Between these two objectives, a rather typical Pareto-optimal front was found by studying the solutions of two DD-EDP instances in the objective space. It seems that the more solution alternatives there are, the wider they spread in the objective space. In addition, sometimes a large proportion of evaluated solutions is inferior to the Pareto-optimal region. An on-line multi-objective optimization algorithm could use additional heuristics to avoid such bad solutions since time is always a critical resource in a real-time control system.

The system constraints of the DD-EDP arise from the Destination Control (DC), which traditionally makes the final decision of both the serving elevator and deck immediately after the passenger call is registered. The semi-continuous double-deck DC allows the group control to change the serving deck after the initial assignment but the fully continuous DC also allows the serving elevator to be changed. The fully continuous DC also requires a new user interface concept. Based on the Pareto-optimal solutions to the studied problem instances, it seems that the FC could improve passenger waiting and/or transit times by about 10-15%. However, this estimation needs to be verified by proper simulations to get a more reliable estimate.

Naturally, the question arises about what the group control should optimize. The Pareto-optimal solutions showed a rather wide range of values in both the average waiting and transit times. The addition of energy consumption as the third objective would complicate the matter even further. Traditionally, the preference order of the objectives has been the decision maker's dilemma in multi-criteria decision making. In an elevator group control, the decision making must be automatic and adaptive since it is executed continuously. However, building managers and even individual users could have a possibility to state their preferences, which the group control could take into account when making the dispatching decisions.

Acknowledgment

The authors would like to thank Professor Kalyanmoy Deb and Rick Barker, Principal of Barker Mohandas, USA, for inspiring discussions about our common interests.

References

[1] J. Fortune, "Modern double-deck elevator applications and theory," *Elevator World*, vol. 44, no. 8, pp. 63–68, 1996.

[2] J. Sorsa, M.-L. Siikonen, and H. Ehtamo, "Optimal control of double-deck elevator group using genetic algorithm," *International Transactions in Operational Research*, vol. 10, no. 2, pp. 103–114, 2003.

[3] J. Sorsa and M.-L. Siikonen, "Double-deck destination control system," *Elevatori*, vol. 37, no. 5, pp. 42–57, 2008.

[4] G. Barney, "Towards agreed traffic design definitions," *Elevator World*, vol. 53, no. 2, p. 108, 2005.

[5] T. Tyni and J. Ylinen, "Evolutionary bi-objective optimisation in the elevator car routing problem," *European Journal of Operational Research*, vol. 169, no. 3, pp. 960–977, 2006.

[6] S. Kobori, M. Iwata, N. Suzuki, and S. Yamashita, "Energy-saving techniques of elevator group control system," in *Elevator Technology 18*, A. Lustig, Ed. IAEE, 2010, pp. 167–176.

[7] J. Sorsa, H. Hakonen, and M.-L. Siikonen, "Elevator selection with destination control system," *Elevator World*, vol. 54, no. 1, pp. 148–155, 2006.

[8] M.-L. Siikonen, T. Susi, and H. Hakonen, "Passenger traffic flow simulation in tall buildings," *Elevator World*, vol. 49, no. 8, pp. 117–123, 2001.

[9] T. Tyni and J. Ylinen, "Genetic algorithms in elevator car routing problem," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001).* San Francisco, CA, USA: Morgan Kaufman Publishers, 2001, pp. 1413–1422.

[10] M.-L. Siikonen, "Planning and control models for elevators in high-rise buildings," Ph.D. dissertation, Helsinki University of Technology, Systems Analysis Laboratory, 1997.

[11] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, England: John Wiley & Sons, 2001.