

OPTIMAL CAR DISPATCHING FOR ELEVATOR GROUPS USING GENETIC ALGORITHMS

BERNA BOLAT

*Yildiz Technical University, Faculty of Mechanical Engineering,
Mechanical Engineering Department, Yildiz, TR-34349, Istanbul, Turkey.
Email: balpan@yildiz.edu.tr*

PABLO CORTES

*Escuela Técnica Superior Ingenieros, Ingeniería Organización, Seville University,
Camino de los Descubrimientos s/n, Sevilla 41092, Spain.
Email: pca@esi.us.es*

ERSUN YALÇIN

*Yildiz Technical University, Faculty of Mechanical Engineering,
Mechanical Engineering Department, Yildiz, TR-34349, Istanbul, Turkey.
Email: yalcin@yildiz.edu.tr*

MUSTAFA ALIŞVERİŞÇİ

*Yildiz Technical University, Faculty of Mechanical Engineering,
Mechanical Engineering Department, Yildiz, TR-34349, Istanbul, Turkey.
Email: alis@yildiz.edu.tr*

ABSTRACT— The car dispatching problem in an elevator group consists of assigning cars to the hall calls at the same time that car call are served. The problem needs to coordinate the movements of individual cars with the objective of operating efficiently the whole group. In this paper, we propose an elevator group control system based on a genetic algorithm which makes use of a novel fitness function to evaluate the individuals. The fitness function allows a quick execution of the algorithm. Tests are provided for various types of high-rise buildings to assess the elevator service performance. Comparative simulations show that our genetic algorithm outperforms traditional conventional algorithms developed in the industry. It is important to note that the algorithm is quickly evaluated allowing a real-life implementation.

Key Words: Genetic Algorithms, elevator control systems, vertical transportation system

1. INTRODUCTION

Elevators are widely utilized for vertical transportation of people and goods in buildings. The elevator expectations are to provide good service at every floor of the building, to reduce the travel time from one floor to another, to reduce the

waiting time at each floor, and to provide all these services at the same time that energy consumption is reduced. One of the main problems corresponds to the dispatching of cars when a hall call is registered. The group control system needs to select and allocate an elevator car to the call immediately. So, elevator group control systems respond to the necessity of providing efficient control for a group of automatic elevators that serve a common set of landing calls in an efficient manner [1].

Additionally, the elevator traffic intensity and its pattern depend on the building and can vary during the day [2]. In fact, the variation of the passenger movement results in a traffic pattern to be built up in a specific building. Generally, the elevator system design method makes use of the up-peak period as a basis for calculations, which consists of a traffic main stream from the ground or basement floors to the rest of floors, although this aspect is being revised currently [3]. Sometimes, lunch-peak traffic can perform a more complex traffic pattern than up-peak, and down-peak should not be disregarded at all. So, we take these aspects into consideration for our trials.

Soft computing applications have been studied in elevator control. Fuzzy logic approaches, neural networks and expert systems can be cited among the most relevant researches. So, fuzzy logic rules to dispatch elevator cars has been studied in [4]. A fuzzy logic controller is presented to take non-predetermined control decisions for a fuzzy orbital scheduler in [5]. Traffic control simulation and traffic detectors using fuzzy logic base system rules are also provided in [6]. Artificial neural networks are used in [7] to control the elevator system, and results of simulations using dual/triple traffic control systems get to reduce the passenger waiting time. In other line, algorithms providing the capability of dynamic zoning according to variable traffic conditions in buildings have been investigated in [8]. Also, an approach for lift traffic control based on traffic sensing and a rule-based expert system is presented in [9]. The system links the expert system to the traffic sensing system to calculate the optimum car assignment in a continuous manner.

Genetic algorithms are a powerful technique that can be conveniently used to optimize the elevator dispatching problem. Some studies carried out using genetic algorithms are: [10] where a genetic algorithm is proposed to control lunch-peak situations. The paper presents the results for a specific professional building showing the reduction of the average passenger waiting time. [11] has studied the application of genetic algorithms using a multi-objective function that tries to satisfy a specified average passenger waiting time with the least consumption of energy. [12] gives genetic algorithm based routings for double-deck elevators what are a two elevator cars which are attached one on top of

the other allowing passengers on two consecutive floors to be able to use the elevator simultaneously, increasing the passenger capacity of an elevator shaft.

This paper is focused on reducing the passenger average journey time, average waiting time, and average travel time. To do so, we design a genetic algorithm quickly evaluated that makes use of a novel fitness function allowing a real implementation in the industry. The fitness function provides better results compared to conventional control methods. In fact, the reduction of average passenger waiting time, average passenger journey time and average passenger travel time are nearly 25% respectively. Thus, the new controller allows a more efficient use of the system, reducing the waiting and journey times of passengers under different types of traffic. The rest of the paper follows with an analysis of the service performance parameters of elevator groups in section 2. Section 3 includes the description of the genetic algorithm. Section 4 shows and compares the results obtained by the genetic algorithm and conventional algorithms for several cases studies. And finally we summarize the main aspects of the paper and the most relevant contributions in the conclusion section.

2. SERVICE PERFORMANCE PARAMETERS OF ELEVATOR GROUPS

The performance of a group of elevators is assessed analysing the passenger Average Waiting Time (AWT), the passenger Average Travel Time (ATT) and the passenger Average Journey Time (AJT) that is calculated as the sum of the other two (1).

$$AJT = ATT + AWT \quad (1)$$

ATT is the time the responding elevator doors begin to open to the time the doors begin to open again at the passenger's destination.

AWT is defined as the actual time a prospective passenger waits after registering a hall call (or entering the waiting queue if a call has already been registered) until the responding elevator doors begin to open. For car loads less than 50%, it is possible to develop an approximate equation for AWT as (2), [1].

$$AWT = 0.4INT \quad (2)$$

For car loads more than 50%, it is possible to develop an approximate equation for AWT as (3).

$$AWT = \frac{2Ht_v + (S+1)t_s + 2Pt_p}{L} \left[0.4 + \left(\frac{1.8P}{CC} - 0.77^2 \right) \right] \quad (3)$$

being INT the interval (the main floor arrival average time), P the number of passengers, CC the rated car capacity, H the highest reversal floor, S the expected number of stops, t_v the single floor transit time (in seconds), t_s the stopping time (in seconds), t_p the passenger transfer time (in seconds), and L the number of cars within the elevator group.

Finally, ATT is calculated as (4).

$$ATT = t_v \frac{H}{2S} (S + 1) + t_s \frac{S + 1}{2} + t_p \cdot P \quad (4)$$

A full discussion of parameters in formula (2-4) can be found in [1] and [13].

3. A GENETIC ALGORITHM FOR THE ELEVATOR GROUP CONTROLLER

Our proposal consists of a fast and efficient algorithm to minimise the average journey time based on genetic algorithms. Genetic algorithms are inspired in the natural selection principle, whose main idea is that a new powerful offspring forms is expected from old generations. The algorithm makes use of a hall call allocation strategy to define the genome of individuals to perform the elevator group controller.

The initial population is created according to a hall call allocation based on a 5-minute period. The chromosome of the individuals is defined by as many arrays of size $2 \times [\text{Number of floors} - 1]$ as cars are in the group. For each car, the partial chromosome consists of up and down hall call allocations as described in Table 1. The table example corresponds to a 20-floor building. The genotypes are binary encoded. Bit 0 indicates no hall call allocation for the car, and bit 1 indicates that the registered hall call of that floor is allocated to the car.

Table I. Up and down hall calls in a car

Floors	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
Upwards landing calls	1	1	0	0	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0
Downwards landing calls	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1

The definition of the population size is one of the most important aspects in genetic algorithms because large populations lead to a major effectiveness of searching, mapping wide areas of the feasibility region, but this searching process usually takes non-feasible time consumption. On the contrary, small populations do not map wide enough areas, and bad solutions are expected [10]. After testing several instances we selected a 30 individuals' population for buildings up to 20 floors and a 50 individuals' population for higher buildings.

The selection of individuals is carried out with the roulette wheel selection method. Roulette wheel selection is intuitive, easy to implement and proportional to an individual's fitness. Crossover and mutation are the genetic operators we used. The crossover combines the genes randomly to produce a population of chromosomes. Single-point, two-point and uniform crossover techniques were applied. All of them provided different solutions with different ranges of applicability that are discussed in section 4. Also, mutation was applied

to the offspring after crossover. Mutation in a binary encoded string is carried out by interchanging a '1' by a '0' or vice versa between a pair of cars. The aim of mutation is to enrich the population genetically, ensuring differences of offspring from old generations. Mutation probability varying from 0.9 to 1.0 was considered. Figure 1 shows the flowchart of the genetic algorithm.

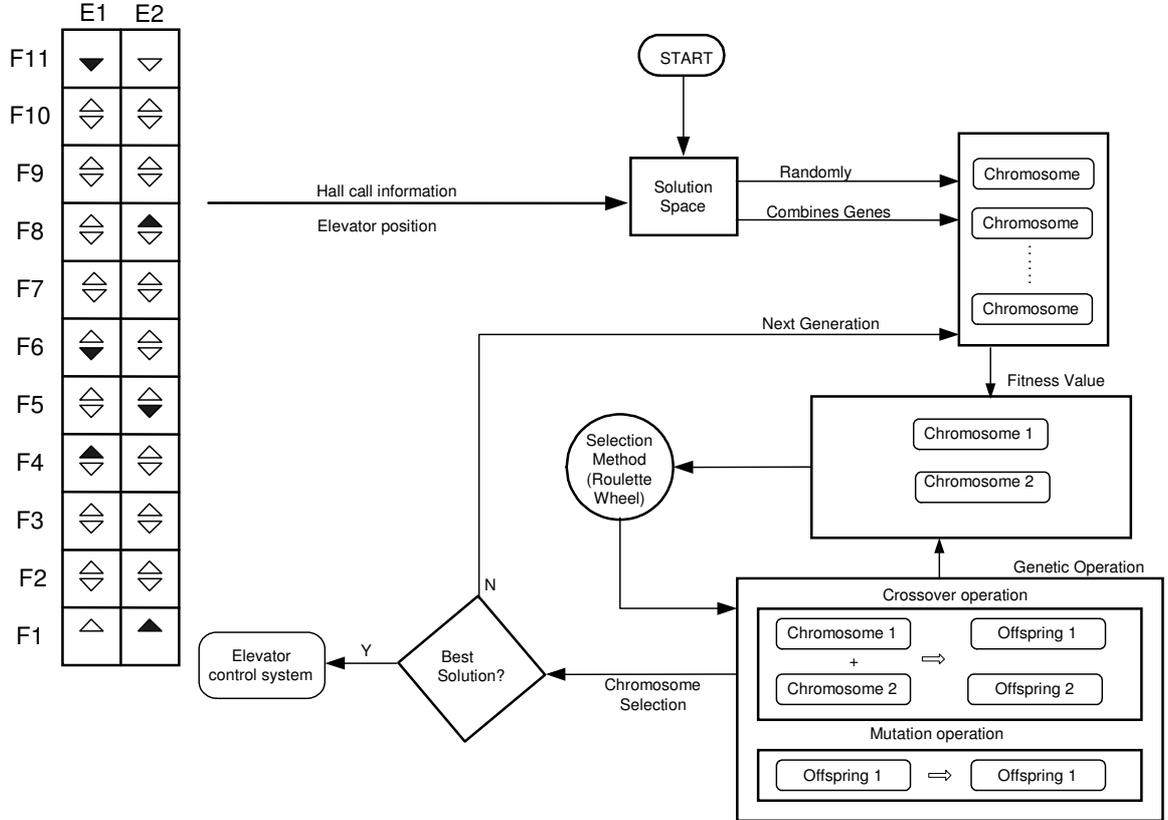


Figure 1. Genetic algorithm flowchart

During the generations of the genetic algorithm, the fitness of each individual is calculated evaluating the fitness function. The success of genetic algorithms generally depends on the efficiency of assessing the individuals' fitness. The fitness function has to be calculated quickly and with accuracy. So, the fitness function we propose can be calculated separately for each car using the corresponding hall call allocations defined in the chromosome (5).

$$F_{Ci} = [(F_1 - 1) + |F_1 - F_2| + (F_2 - F_3)] \cdot t \quad (5)$$

where F_{Ci} is the fitness function for car i , and F_1 represents the highest floor in upward direction, F_2 the highest floor in downward direction, F_3 the lowest floor in downward direction, and t the inter-floor trip time (in seconds). Finally, the

total fitness function for the individual is evaluated summing all the partial chromosomes' fitness of the cars in the group, (6).

$$F_T = F_{C1} + F_{C2} + F_{C3} + F_{C4} + \dots + F_{CN} \quad (N \text{ cars}) \quad (6)$$

Formula (5-6) allows evaluating the fitness of the individuals (car allocations) quickly, so a very fast execution of the genetic algorithm can be obtained.

As an example, figure 2 depicts the variation of the fitness value with respect to the generations. The figure shows the values of all the individuals in the population in relation to each generation. It can be viewed the convergence of the method being reduced the fitness of the population as well as the fitness of the best individual. To illustrate the behaviour we have represented 500 generations, but the best solution is not modified after generation 115. In general, reasonably good solutions are obtained for a number of generations around 100 generations. The calculus corresponds to a 20 floor building that is analysed in section 4.2.

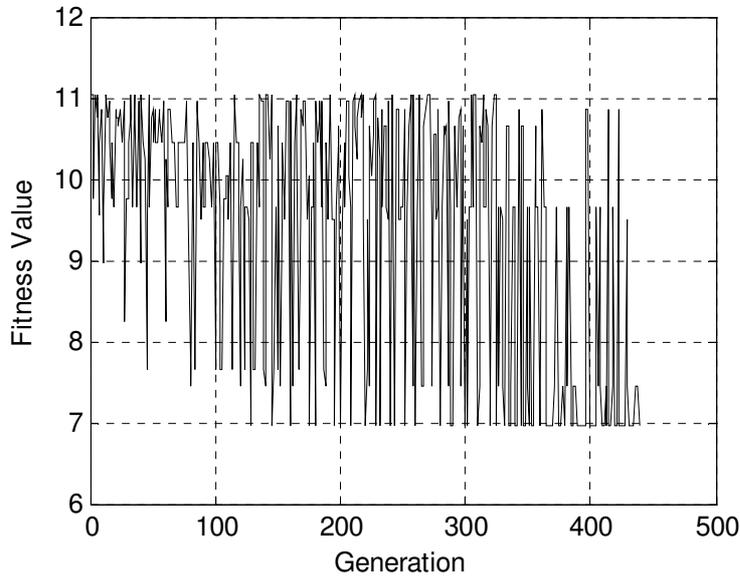


Figure2. Genetic algorithm learning curve

4. NUMERICAL SIMULATIONS AND RESULTS

To test the efficiency of the genetic algorithm we constructed several buildings with different characteristics that are analysed in the next case studies.

We simulated a lunch-peak period which includes characteristics from up-peak and down-peak situations being one of the most critical periods in vertical transportation. The lunch-peak period was simulated with a 40% up-peak, 40%

down-peak flow and 20% inter-floor traffic following the CIBSE guide [13]. Arrival of passengers was simulated using a statistical uniform distribution. A five minute period was analysed, what is a usual rule in vertical transportation systems, [1;13].

The genetic algorithm was tested with respect to conventional algorithms. Conventional control system is based on principles of automatic passenger collecting control system. Figure 3 depicts the architecture of such conventional control algorithms, and [14] provides full description of such algorithms. The car collects passengers wanting to travel in the same direction of the car. The car moves from the initial floor to the last registered hall call or car call. This method known as simplex system is used to control each single car in multi-cars systems. Simplex system aims to prevent cars from going to the backward direction and to prevent cars from moving without hall calls. In multi-car systems, cars are assigned to certain floor calls trying to reduce passenger waiting times. This system is adjusted according to changes in the traffic conditions.

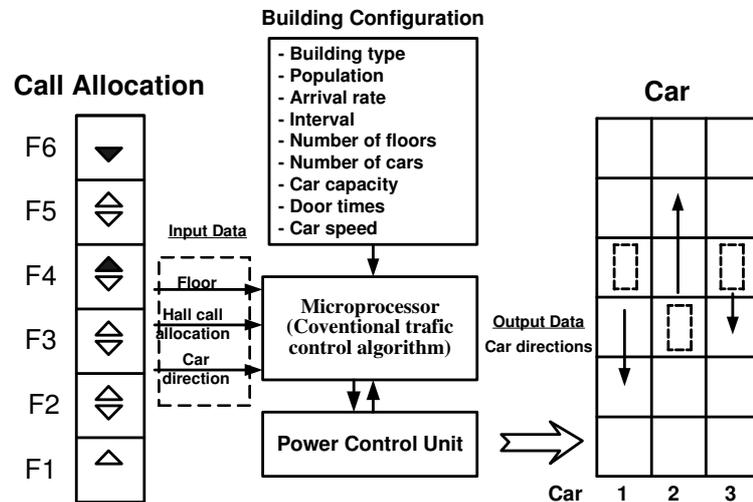


Figure 3. Control architecture with a conventional algorithm

4.1 First case study: 12 floors building and two cars group

We constructed a building with 12 populated floors and we simulated lunch peak traffic which is a very critical situation in vertical traffic because it includes up-peak and down-peak traffic at the same time. The elevator group consisted of two 20-people capacity cars.

AWT was 195.6 seconds for the conventional system. The genetic algorithm reduced the system waiting times significantly providing 147.40 seconds what corresponds to a 24.64% waiting time reduction. Results of genetic algorithm are

shown in Figure 4 which depicts the system waiting times with respect to the generations of the algorithm. It shows the quality of the solution being improved when generations are increased.

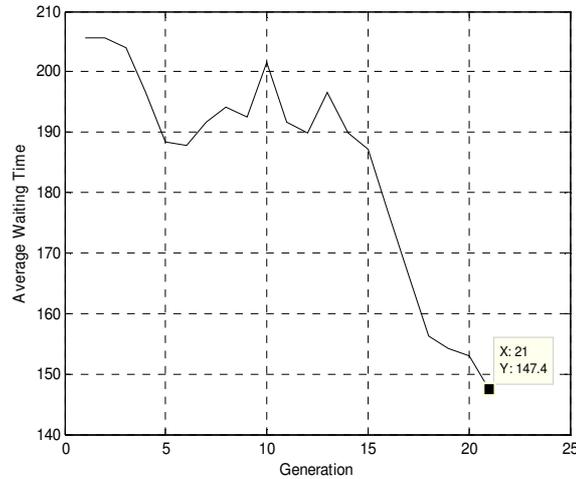


Figure 4. AWT evolution depending on the GA generations

Table II shows the summary of results for genetic and a conventional algorithm implemented by Buga Otis Elevator Company in Turkey (input data for the case study were supplied by the company and are related to a high-rise building). The worse results for conventional algorithms are due to several reasons. In fact, the conventional elevator control method is simple and rough, being this method suitable for low rise buildings only. So, it cannot satisfy the complicated traffic that is requested in high-rise buildings. Moreover, the total service performance of elevator group systems is incompletely and/or narrowly evaluated.

Table II. Comparison of GA and conventional algorithms for case study I

Input data			Output data		
Building	Number of floors	12	AWT	GA algorithm	Conventional algorithm
	Population of floors	30		147.70 s (SPC) 147.40 s (TPC) 148.32 s (UC)	195.60s
	Traffic	20		AJT	372.10 s

	pattern generation			(SPC) 371.80 s (TPC) 375.32 s (UC)	
Cars	Number of cars	2	ATT	224.40 s (SPC) 224.40 s (TPC) 227.00 s (UC)	269.28 s
	Capacity of cars	20	Computational time	5,4 s (TPC) 5,4 s (SPC) 6.1 s (UC)	Note: SPC: <i>Single-Point Crossover Technique</i> TPC: <i>Two-Point Crossover Technique</i> UC: <i>Uniform Crossover Technique</i>

4.2 Second case study: 24 floors building and different configurations of car groups

A 24 floors building (30 people populated per floor) is considered. Results are provided for different car group configurations. The capacity of cars is eight people. Table III summarizes the results for the average waiting time (AWT) and the round trip time (RTT). Results in Table III correspond to extreme values trying to appreciate the algorithm performance and to show how the genetic algorithm reaches a better performance attending to the system waiting time than conventional controllers. The reduction of waiting times is around 25%. Consequently, the RTT was significantly reduced in a similar percentage.

Table III. RTT & AWT results for GA and conventional (conv) algorithms in a 5-minute period

Floor	RTT-conv	RTT-GA	AWT							
			24-floors/2 cars		24-floors/3 cars		24-floors/4 cars		24-floors/6 cars	
			conv(s)	GA(s)	conv(s)	GA(s)	conv(s)	GA(s)	conv(s)	GA(s)
10	91	68.43 (SPC)	42.67(SPC)	28.45(SPC)	21.33(SPC)	14.28(SPC)				
		61.92 (TPC)	38.61(TPC)	25.74(TPC)	19.31(TPC)	12.87(TPC)				
		68.43 (UC)	42.67(UC)	28.45(UC)	21.33(UC)	14.22(UC)				
11	93.6	75.44 (SPC)	47.04(SPC)	31.36(SPC)	23.52(SPC)	15.68(SPC)				
		75.44 (TPC)	47.04(TPC)	31.36(TPC)	23.52(TPC)	15.68(TPC)				
		78.71 (UC)	49.08(UC)	32.72(UC)	24.54(UC)	16.36(UC)				
12	96	78.26 (SPC)	48.80(SPC)	32.53(SPC)	24.40(SPC)	16.27(SPC)				
		78.37 (TPC)	48.87(TPC)	32.58(TPC)	24.43(TPC)	16.29(TPC)				
		92.01 (UC)	57.37(UC)	38.25(UC)	28.69(UC)	19.12(UC)				
13	98.6	79.61 (SPC)	49.94(SPC)	33.09(SPC)	24.82(SPC)	16.55(SPC)				
		96.21 (TPC)	59.99(TPC)	39.99(TPC)	30.00(TPC)	20.00(TPC)				
		76.29 (UC)	47.57(UC)	31.71(UC)	23.79(UC)	15.86(UC)				
14	101.2	80.97 (SPC)	50.49(SPC)	33.66(SPC)	25.24(SPC)	16.83(SPC)				
		80.97 (TPC)	50.49(TPC)	33.66(TPC)	25.24(TPC)	16.83(TPC)				
		80.97 (UC)	50.49(UC)	33.66(UC)	25.24(UC)	16.83(UC)				
15	103.6	83.6 (SPC)	52.13(SPC)	34.75(SPC)	26.06(SPC)	17.38(SPC)				
		95.9 (TPC)	59.80(TPC)	39.87(TPC)	29.9(TPC)	19.93(TPC)				
		83.6 (UC)	52.13(UC)	34.75(UC)	26.06(UC)	17.38(UC)				
16	105.4	86.57 (SPC)	53.98(SPC)	35.99(SPC)	26.99(SPC)	17.99(SPC)				
		86.57 (TPC)	53.98(TPC)	35.99(TPC)	26.99(TPC)	17.99(TPC)				
		86.57 (UC)	53.98(UC)	35.99(UC)	26.99(UC)	17.99(UC)				
17	108	86.79 (SPC)	54.12(SPC)	36.08(SPC)	27.06(SPC)	18.04(SPC)				
		86.79 (TPC)	54.12(TPC)	36.08(TPC)	27.06(TPC)	18.04(TPC)				
		86.79 (UC)	54.12(UC)	36.08(UC)	27.06(UC)	18.04(UC)				
18	109.6	90.74 (SPC)	56.58(SPC)	37.72(SPC)	28.29(SPC)	18.86(SPC)				
		92.66 (TPC)	57.78(TPC)	38.52(TPC)	28.89(TPC)	19.26(TPC)				
		92.66 (UC)	57.78(UC)	38.52(UC)	28.89(UC)	19.26(UC)				
19	112.2	91.19 (SPC)	56.86(SPC)	37.91(SPC)	28.43(SPC)	18.95(SPC)				
		91.19 (TPC)	56.86(TPC)	37.91(TPC)	28.43(TPC)	18.95(TPC)				
		91.19 (UC)	56.86(UC)	37.91(UC)	28.43(UC)	18.95(UC)				
20	114	90.42 (SPC)	56.38(SPC)	37.59(SPC)	28.19(SPC)	18.79(SPC)				
		90.42 (TPC)	56.38(TPC)	37.59(TPC)	28.19(TPC)	18.79(TPC)				
		97.12 (UC)	60.56(UC)	40.37(UC)	30.28(UC)	20.19(UC)				
21	117.6	94.33 (SPC)	58.82(SPC)	39.21(SPC)	29.41(SPC)	19.61(SPC)				
		94.33 (TPC)	58.82(TPC)	39.21(TPC)	29.41(TPC)	19.61(TPC)				
		102.32 (UC)	63.80(UC)	42.53(UC)	31.90(UC)	21.27(UC)				
22	118.2	105.41 (SPC)	65.73(SPC)	43.82(SPC)	32.86(SPC)	21.91(SPC)				
		97.43 (TPC)	60.75(TPC)	40.50(TPC)	30.38(TPC)	20.25(TPC)				
		94.23 (UC)	58.76(UC)	39.17(UC)	29.38(UC)	19.59(UC)				
23	120	113.94 (SPC)	71.05(SPC)	47.36(SPC)	35.52(SPC)	23.68(SPC)				
		99.51 (TPC)	62.05(TPC)	41.37(TPC)	31.02(TPC)	20.68(TPC)				
		102.72 (UC)	64.05(UC)	42.70(UC)	32.03(UC)	21.35(UC)				
24	121.6	101.52 (SPC)	63.30(SPC)	42.20(SPC)	31.65(SPC)	21.10(SPC)				
		98.29 (TPC)	61.29(TPC)	40.86(TPC)	30.64(TPC)	20.43(TPC)				
		96.69 (UC)	60.29(UC)	40.19(UC)	30.15(UC)	20.10(UC)				

In general, the GA provided better results for the two-point crossover operator (TPC), especially when we attend to the AWT (for RTT similar results can be appreciated for TPC and UC). The average computational time for the genetic algorithm (TPC case) was 3.7 seconds for the 2 cars case, 3.26 seconds for the 3 cars case, 2.9 seconds for the 4 cars case, and 2.5 seconds for the 6 cars case. It is important to note how computational times for all studied cases are

totally feasible for real industry applications and were only slightly upper than the times provided by conventional controllers.

Figure 5 depicts the evolution of the AJT depending on the number of cars, the number of floor where the passengers make a hall call, and the GA crossover technique. As can be viewed, the time is significantly reduced when a new car is considered. As can be expected, the waiting time is lower for low-rise buildings, but in high-rise buildings the differences are strongly reduced with respect to conventional algorithms. Finally, attending to the crossover technique, the two-point crossover technique showed the better performance in the set of experiments.

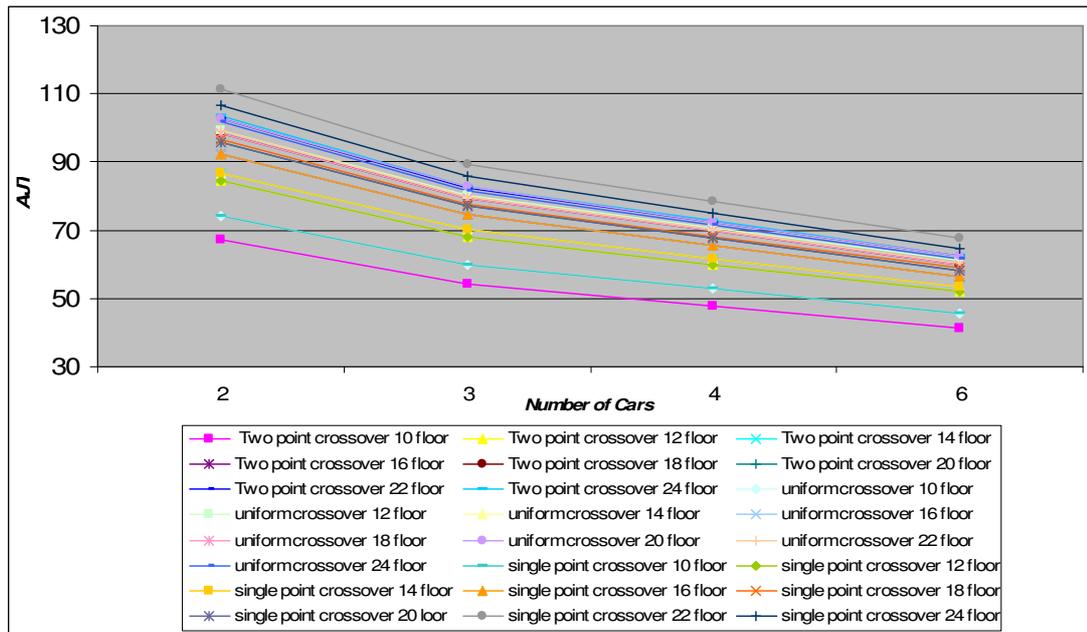


Figure 5. AJT evolution depending on the number of cars, floor and crossover

technique

5. CONCLUSIONS

This paper presents a genetic algorithm for optimizing the car dispatching in an elevator group. Tests have been carried out for several high-rise buildings. Results indicate that the waiting time and journey time of passengers (AWT and AJT) have been significantly reduced. Several crossover operators were tested and different buildings were considered varying the number of floors. Genetic algorithms increased the performance of the elevator control system according

to a decrease around 20-25% in both AWT and AJT. So, the main performance times were decreased when genetic algorithms were used, outperforming clearly other conventional system results and improving the quality of service.

On the other hand, genetic algorithms are iterative and therefore they can take very much computational time when a long population and a great number of iterations are used. The selection of these parameters has to be carried out attending not so much to the algorithm accuracy but to the available time of trip of the elevator between different events (calls). However, the genetic algorithm we are presenting here can be executed in a very short time due to its novel fitness estimation. The computational time of the algorithm outperforms other genetic implementations in the scientific literature. This important characteristic allows the real implementation of the genetic algorithm in the industry. In summary, our genetic algorithm responded to the hall calls quickly (practically in the order of conventional algorithms), and the hall call assignment to cars minimised the passenger waiting times in an efficient manner.

REFERENCES

1. G.C. Barney, *Elevator traffic handbook theory and practice*, Spon Press, London, 2003.
2. Mulvaney. D and M. Hamdi, "Real-time dynamic scheduling and its application to lift systems", *Proceedings of the 12th International Congress on Vertical Transportation Technologies*, Milan, 2002, p.p.207-216.
3. P. Cortes, J. Munuzuri, and L. Onieva, "Design and analysis of a tool for planning and simulating dynamic vertical transport", *Simulation*, 2006, p.p.255-274.
4. B.A. Powell and D. J. Sirag, "Fuzzy Logic: A new way of thinking about the complexities of dispatching elevators", *Elevator World*, 1993, p.p.78-84.
5. M. Ho. and B. Robertson, "Elevator group supervisory control using fuzzy logic", *Canadian Conference on Elevator and Computer Engineering*, 1994.
6. J.H. Lee and H. Lee-Kwang. H, "Distributed and cooperative fuzzy controllers for traffic intersections group" *IEEE Trans. on Systems Man & Cybernetics Part C-Appl. & Reviews*, 1999, p.p. 263-271.
7. C.E. İmrak and G.C. Barney, "The Application of neural networks to lift traffic control", *Elevator World*, 49 (5), 2001, 82.
8. A.T.P. So and W.L. Chan, "Dynamic zoning for intelligent supervisory control", *International Journal of Elevator Engineering*, 1996, p.p. 47-59.
9. R.W. Prowse, T. Thomson, and D. Howells, "Design and control of lift systems using expert systems and traffic sensing", *Elevator Technology 4*, IAEE Publ., 1992, p.p. 219-226.
10. P. Cortes, J. Larrañeta, and L. Onieva, "Genetic algorithm for controllers in elevator groups: analysis and simulation during luncpeak traffic", *Applied Soft Computing*, 2004, p.p.159-174.
11. T. Tyni. and J. Ylinen, "Evolutionary bi-objective optimization in the elevator car routing problem", *European Journal of operational Research*, 2004, p.p.1-17.

12. J. Sorsa, M.L. Siikonen, and H. Ehtamo, "Optimal control of double-deck elevator group using genetic algorithm", *International Transactions in Operational Research*, 2003, p.p.103-114.
13. Cibse Guide D, "*Transportation systems in buildings*", Cibse Pub. London, 2005.
14. C.E. İmrak, Gerdemeli, İ., *Asansörler ve Yürüyen Merdivenler*, Birsen Yayınevi, İstanbul , 2000,p.p. 49-64.