Designing Traffic-Sensitive Controllers for Multi-Car Elevators through Evolurtionary Multi-Objective Optimization

Kokolo Ikeda[1], Hiromichi Suzuki[2], Sandor Markon[2], and Hajime Kita[1]

[1] Kyoto University, Yoshida-Nihonmatsu, Sakyo, Kyoto 606-8501, JAPAN,
kokolo@media.kyoto-u.ac.jp kita@media.kyoto-u.ac.jp,
WWW page: http://www.ipe.media.kyoto-u.ac.jp/indexE.html
[2] Fujitec Co. Ltd, Big Wing, Hikone, Shiga 522-8588, JAPAN

Abstract. Multi-Car Elevator (MCE) that has several elevator cars in a single shaft attracts attention for improvement of transportation in high-rise buildings. However, because of lack of experience of such novel systems, design of controller for MCE is very difficult engineering problem. One of the promising approaches is application of evolutionary optimization to from-scratch optimization of the controller through discrete event simulation of the MCE system. In the present paper, the authors propose application of evolutionary multi-objective optimization to design of traffic-sensitive MCE controller. The controller for MCE is optimized for different traffic conditions in multi-objective way. By combining the multi-objective optimization with the exemplar-based policy (EBP) representation that has adequate flexibility and generalization ability as a controller, we can successfully design a controller that performs well both in the different traffic conditions and works adequately by generalization in the conditions not used in the optimization process.

1 Introduction

The elevator system is a critical component of high-rise buildings, and its design and control have been studied for many years. The control of cooperating elevator cars for efficient service of passengers is known as "the elevator group control problem". This problem is recognized as a difficult control task, involving stochastic, online scheduling with high combinatorial complexity and real-time response requirements. Since no effective analytical solution has been found to date, current commercial systems are controlled by using a combination of heuristic and artificial intelligence methods [Kim et al., 1998][Beielstein et al., 2003] [Zhou et al., 2005].

Recently, with increasing building heights and more complex usage patterns, multi-car elevators (MCEs) consisting of several cars in a single elevator shaft, usually driven by linear motors, are receiving increasing interest as high-performance transportation systems [Kita et al., 2002] [Sudo et al., 2002]. However, the accumulated knowledge for conventional elevators is not readily applicable to MCEs, which exhibit distinctly different behavior.

The most promising approach for MCE control appears to be simulation-based optimization, in which the policy of controller is represented by a function model and the parameters are optimized through a simulation. Sudo et al. have shown that the approach with genetic algorithms (GAs) is hopeful [Sudo et al., 2002] [Takahashi et al., 2003]. In those researches, MCE control is performed by assigning a hall-call to a certain car, with using the linear-sum weights $[\alpha_i]$. When a new call occurs, for all available cars, several feature values $[w_i^k]$ expressing the state of the car k are calculated. Then the car with the minimum linear-weighted sum $\sum \alpha_i w_i^k$ is assigned.

In [Ikeda et al., 2006], an exemplar-based policy representation (EBP) [Ikeda, 2005] is employed as a non-linear controller for MCE systems. An advantage of EBP to the controller of linear-sum type is the ability to control flexibly according to the current situation, and the result of numerical experiments has shown its superiority in MCE control.

In those simulation-based researches, the policy of control with adjustable parameters has been evaluated and optimized in a single traffic situation. So, there is no guarantee that the optimized policy works adequately in the other situations, such as in the other building or in the other traffic condition which changes largely depending on such as time-of-day. For practical use, considering the cost and difficulty of detecting and switching the policy depending on the situations, it is preferable that one policy works adequately in various situations as much as possible. In large part of conventional control methods, the current situation is detected by the set of rules such as fuzzy rules, and the corresponding control policy tuned separately is performed. Normally the rules are written out by experts and so very expensive.

In this paper, we employ the multi-objective optimization approach [Deb et al., 2000] [Obayashi and Sasaki 2004] in order to obtain the traffic-sensitive controller for MCE. In this approach, the policy with parameters is evaluated in multiple situations, the objective functions are defined respectively, and multi-objective optimization method is applied. The advantages of EBP for this approach are the ability to control flexibly according to the situation and the ability to generalize it.

This paper is organized as follows. In Section 2, a brief overview of the MCE system and its controller are shown. In Section 3, the simulation-based policy optimization of MCE controller, single-objective and multi-objective, are explained. In Section 4, experiments are done and the result is analyzed, and in Section 5, the paper is concluded.

2 MCE System and Controllers

2.1 Multi-Car Elevator Systems

The almost same MCE system described in [Takahashi et al., 2003] is considered in this study. The elements comprising the system are as follows (see Fig. 1).

Floors The lowest level of the building is assumed to be the sole point of entry (and exit) to the building, and thus experiences the highest traffic (10

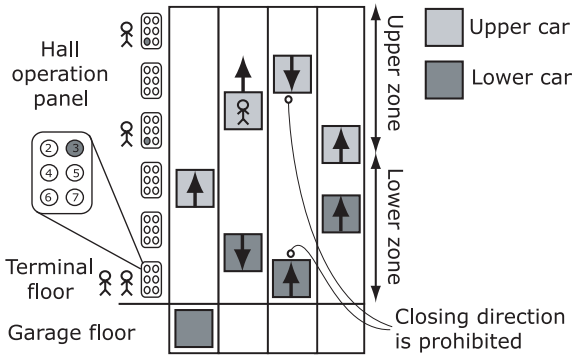

Fig. 1. MCE system

times higher than other floors in this simulation). The lowest floor is called "the terminal floor". The other floors are assumed to be identical in terms of traffic demand, and are called "general floors".

Elevator Shafts Shafts represent the space in which elevator cars (or cages) move. In the present simulation, the building has 4 shafts.

Elevator Cars Each elevator shaft is considered to host two cars, which can only move vertically and cannot pass each other. Furthermore, to avoid collision and dead lock, cars in a single shaft are not allowed to approach each other simultaneously. These constraints make efficient control of MCEs difficult to achieve.

Registration of Destination Floor It is assumed that the passengers register their destination floors not in the car but in the hall, and that passengers are guided to the car serving their need.

Zone Operation For ease of operation, the floors are divided into upper and lower zones. The upper car in each shaft serves only the traffic demands whose origin or destination is in the upper zone. The lower car serves only the lower zone.

Garage Floor To allow the upper car to serve the terminal floor, a garage floor at which the lower car stops is introduced below the terminal floor.

2.2 MCE Controller

In [Sudo et al., 2002], [Takahashi et al., 2003], [Ikeda et al., 2006] and this paper, MCE control is performed by assigning a hall call to a certain car. When a new call occurs, the call is assigned to a car by the following procedure:

1. For each shaft, the car that can serve the call is nominated, according to the definition of the service zone.

2. For all the nominated cars, several feature values expressing the state of the car are calculated. In this paper, $N_{\text{features}} = 4$ features $[w_1^k, w_2^k, w_3^k, w_4^{(k)}]$ are utilized where k is the car index. w_1^k is the estimated waiting time of the new call if assigned, w_2^k is the estimated maximum load of the car if assigned, and w_3^k is the estimated delay time when the car pass through the call and the next car serves it. Finally $w_4^{(k)}$ is the feature expressing the degree of current traffic, which is calculated by $\sum_k w_1^k$ and $\sum_k w_2^k$. The feature $w_4^{(k)}$ is common to all the cars. All the features are normalized so that almost all features are distributed in $[0, 1]$.
3. The preferences of cars are evaluated using the calculated feature values (or feature vectors), and the most preferable car is assigned to the call.

This decision of the car based on the feature values at the Step 3. in the above procedure is the central issue of design problem, and we have proposed the following two methods.

2.3 Linear-Sum Policy Controller

In this controller, given feature vectors are evaluated by a linear-weighted sum function, that has been used in [Sudo et al., 2002] [Takahashi et al., 2003]. Given weights α_i, the car with the minimum weighted sum $k^* = \arg\min_k \sum_{i=1}^4 \alpha_i w_i^k$ is assigned to the given call. For example, if $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 0, 0, 0)$, the policy assigns the earliest car to a call. And if $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0, 1, 0, 0)$, the policy assigns the lightest car.

This approach, referred to as the linear-sum policy (LSP), is very simple and easy to implement. However, it is unable to make decisions flexibly depending on the state common to all the cars, such as whether the traffic is light or heavy. In other words, the weights of each feature value in the LSP approach are fixed, and do not vary according to the situation.

2.4 Exemplar-Based Policy Controller

In [Ikeda et al., 2006], an exemplar-based policy (EBP) representation was employed as a non-linear evaluator of candidate feature vectors.

An EBP consists of a set of exemplars, and an exemplar is defined as the pair of feature vectors $(v_j^1, v_j^2) \in \mathbb{R}^{N_{\text{features}}} \times \mathbb{R}^{N_{\text{features}}}$, meaning "to assign the call to the car with the feature vector v_j^1 is better than to assign it to the car with v_j^2". When a set of candidate feature vectors $C = \{w_c\}_c$ is given, a tournament is created, and $|C|-1$ competitions based on the set of exemplars are conducted, finally the car corresponding to the winning vector is assigned to a given hall call. In this procedure, only the set of nearer exemplars to the feature vectors are referred (For detail, see Appendix A).

3 Simulation-based Policy Optimization

It is assumed that the zone boundary is fixed, and that the calculation of feature vectors is also fixed. Then, optimization of the controller is performed in terms of the parameters of the policy selecting the most preferable vector from the candidates. The parameters are optimized through a simulation of the MCE and a Genetic Algorithm (GA).

3.1 Evaluation using MCE Simulation

For this simulation, the same simulator used in [Takahashi et al., 2003] is employed. This simulator is based on the discrete event model called the Extended State Machine (ESM), which models the system using finite state machines with timers, among which messages are exchanged for synchronization [Kita et al., 2002][Mimaki et al., 1999]. In the ESM model, each of the elevator cars and the corresponding doors is represented by an ESM.

Table 1. Specifications of building and MCE

Item	Value
No. of Floors	30
Zone boundary : the lowest floor of the upper zone	16
No. of Elevator Shafts	4
No. of Cars/Shaft	2
Floor Height	4.34 m
dv^2/d^2t of Car	2.0 m/s^3
Max. Car Acceleration	1.1 m/s^2
Max. Load (persons/car)	20
Time Needed for	
Opening Doors	1.8 s
Closing Doors	2.4 s
Riding/Leaving	1.2 s/person
Passenges to serve (/hour)	750 to 2250
Traffic distribution (Terminal Floor $\leftrightarrow$ General Floor : General Floor $\leftrightarrow$ General Floor)	(10:1)

For evaluating and selecting in a GA, the fitness of a solution (policy) is defined by the averaged squared waiting time ($ASWT$) over the period of simulation (90 min in this case). To reduce the effect of transient stage of traffic, simulation result for a certain period (30 min) is excluded from evaluation. The specifications of the building considered in the simulation are listed in Table 1. Simulations were performed using a supercomputer Fujitsu HPC2500 of Kyoto University, using 32 CPUs among 128 CPUs in a node in a master-slave architecture for parallel computing.

3.2 Obtaining Traffic Sensitive Controller through Single and Multi Objective Optimization

Now, the purpose of optimization is defined as obtaining the control policy of MCE, which performs adequately in the wide-range conditions, i.e. from the light traffic (1000 persons/hour) to the heavy traffic (2000 persons/hour).

An important ability expected for the policy representation is the condition-sensitive control. Unfortunately, it is unable for LSP to make decisions flexibly depending on the condition, whether the traffic is light or heavy. Another important ability is the generalization. Strictly, this purpose can be formalized as the 1001-objective optimization problem, $\min\{ASWT_i(x)\}_{1000 \leq i \leq 2000}$, where $ASWT_i(x)$ is the $ASWT$ of the policy x when i passengers occur per hour. However, evaluations of 1001-objective functions are very expensive and not necessarily required if generalization ability is expected for the policy representation.

For comparison study of combination of the controller representation and the type of GA, we employ two styles of controllers EBP and LSP, and compared five types of GAs to optimize their policies.

One GA, we call GA_{1000}, is carried out to attain the policy for light traffic situation (1000 persons/hour), only the $ASWT$ for the situation ($ASWT_{1000}$) is used for selection. In the same way, GA_{1500} is a GA in which only the $ASWT_{1500}$ is used for selection, and GA_{2000} is a GA in which only the $ASWT_{2000}$ is used. By applying such single-objective optimization, the optimized policy is expected to perform well at least in the considered situation.

In GA_{1000}, GA_{1500} and GA_{2000}, only one fitness function is considered and the others are ignored. In GA_{moop}, both of $ASWT_{1000}$ and $ASWT_{2000}$ are referred, and multi-objective optimization method is carried out. Generally, the purpose of multi-objective optimization is not to attain "the best solution" but to attain the set of non-dominated (Pareto) solutions. However, if the policy representation has enough condition-sensitive control ability, the optimized policy will perform well at both of light and heavy traffics.

In GA_{comb}, the combined single fitness $ASWT_{\mathrm{comb}} = 2 \times ASWT_{1000} + ASWT_{2000}$ is used for selection. GA_{comb} is the GA to attain one of Pareto solutions using a fixed tradeoff rate. So, this is a single-objective optimization but two situations are considered as GA_{moop}.

The characteristics of GAs we employ are summarized in Table 2.

Table 2. Characteristics of five GAs we employ

Name	use $ASWT_{1000}$	use $ASWT_{1500}$	use $ASWT_{2000}$	optimization method
GA_{1000}	yes	no	no	single objective
GA_{1500}	no	yes	no	single objective
GA_{2000}	no	no	yes	single objective
GA_{comb}	yes	no	yes	single objective (combined)
GA_{moop}	yes	no	yes	multi objective

3.3 GA for Single Objective Optimization

The parameters to be optimized for LSP is the set of weights $\alpha_i \in \mathbb{R}^{N_{\text{features}}}$, and the parameters to be optimized for EBP is the set of exemplars that one of them is $(v_j^1, v_j^2) \in \mathbb{R}^{N_{\text{features}}} \times \mathbb{R}^{N_{\text{features}}}$. The common framework of GA
[Ikeda and Kobayashi, 2002] is used for EBP and LSP single optimization as follows:

1. Parameters such as N_{pop} are fixed (see Table 3). In this research, they were selected by some exploratory experiments.
2. As the population, N_{pop} solutions are initialized. If LSP, each solution, N_{features} weights, $\alpha_i \in \mathbb{R}^{N_{\text{features}}}$ is randomly generated. If EBP, each solution E_i, set of $N_{\text{exemplars}}$ exemplars are randomly generated. An exemplar $e_{i,j} \in E_i = (v_{i,j}^1, v_{i,j}^2)$ is generated such that $v_{i,j}^1 + v_{i,j}^2 \in [0, 2]^{N_{\text{features}}}$ and $v_{i,j}^2 - v_{i,j}^1 \in [-1, 1]^{N_{\text{features}}}$.
3. N_{pop} solutions are randomly ordered, s_1, s_2, ..., $s_{N_{\text{pop}}}$. Then N_{pop} pairs (s_1, s_2), (s_2, s_3), ..., ($s_{N_{\text{pop}}}$, s_1) are passed to the following alternation procedure.
   (a) Parents (p_1, p_2) are given.
   (b) Children are reproduced by applying the crossover operator N_{children} times. For LSP, UNDX [Ono 1997] is used as crossover operator of real value vectors, and mixture of exemplars [Ikeda et al., 2006] is used for EBP (For detail, see Appendix B).
   (c) The evaluation value for each policy of the family (p_1 and children) is calculated by simulation of the MCE. To reduce the random fluctuation of evaluation values, N_{sims} simulation runs are performed independently and the average of the evaluation criterion is used. Such a GA is referred to as a N_{sims}-sample GA.
   (d) The policy p_* having the best evaluation value (lowest $ASWT_{1000}$, $ASWT_{1500}$, $ASWT_{2000}$ or $ASWT_{\text{comb}}$) in the family is selected, and p_1 in the population is replaced by p_*.
4. Step 3. is repeated $N_{\text{generations}}$ times, after which the final result, trained MCE control policy is obtained.

3.4 GA for Multi Objective Optimization

As GA_{moop}, a common framework of multi-objective optimization is used for both of EBP and LSP. Considering the noisy fitness evaluation and then uncertainty of ranking, NSGA-II [Deb et al., 2000] with some minor modification is employed as follows:

1. Parameters are fixed (see Table 3).
2. As the population, N_{pop} solutions are initialized by the same procedure to Section 3.3.
3. N_{children} solutions are reproduced by applying the crossover operator (used in Section 3.3). In this step, parents are randomly selected for each reproduction.

Table 3. Notation and parameter values used in optimization

Symbol	Explanation	Value
N_{pop}	Number of solutions (policies) in a population	30(single-objective), 60(multi-objective)
N_{children}	Number of children produced per reproduction step	6(single-objective), 150(multi-objective)
N_{sims}	Number of simulations for one evaluation	4(EBP), 8(LSP)
$N_{\text{generations}}$	Number of generations	80(EBP), 40(LSP)
$N_{\text{exemplars}}$	Number of exemplars in a EBP	900
k_{NN}	Localization parameter (the smaller, the localized)	30
E_i	The ith EBP, the set of exemplars of the ith policy	-
$e_{i,j}$	The jth exemplar of E_i	-

4. $N_{\text{pop}} + N_{\text{children}}$ solutions are evaluated, *i.e.* $ASWT_{1000}$ and $ASWT_{2000}$ are calculated by N_{sims} simulation runs each.
5. For each solution, the dominance-rank and the crowding-distance are calculated. As the crowding-distance, the Euclid distance to the nearest solutions with even-or-better rank is used.
6. The best N_{pop} solutions are selected to survive. The solution with the lower rank wins, and the solution with the smaller distance wins if draw in their ranks. Further, when draw in both ranks and distances, their distances to the second nearest solutions are compared.
7. Step 3. to Step 6. are repeated $N_{\text{generations}}$ times, after which the final result, trained MCE control policies with varieties are obtained.

Please note, though the different parameters are used, the total evaluation times are of the same numbers for EBP/LSP and for single-objective/multi-objective optimization.

4 Experiments

The specifications of the building considered in the experiments are listed in Table 1, and the parameters used for the GAs are shown in Table 3.

We employed two styles of controllers, EBP and LSP, and five types of GAs, GA_{1000}, GA_{1500}, GA_{2000}, GA_{comb} and GA_{moop}. To assess the performance of the optimization procedure, five independent GA trials with different random seeds were conducted for each series.

4.1 Evolution process

Measured by a single criterion : Figure 2(left) shows the evolution processes of GA_{1000}, GA_{comb} and GA_{moop} for EBP. The average $ASWT_{1000}$ of a period is calculated for each trial, and their averages and standard deviations are shown. Figure 2(right) shows the evolution processes of GA_{2000}, GA_{comb} and GA_{moop}

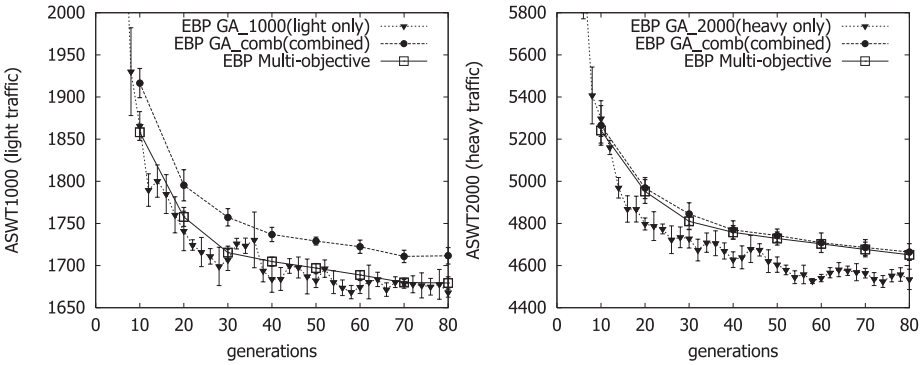

Fig. 2. Evolution process of EBP, $ASWT_{1000}$(left) and $ASWT_{2000}$(right)

for EBP. We can find that the GAs for one criterion, GA_{1000} and GA_{2000} are superior to others in their niches. But the averages of GA_{comb} and GA_{moop} were also soundly decreasing, this means that both $ASWT_{1000}$ and $ASWT_{2000}$ were simultaneously improved.

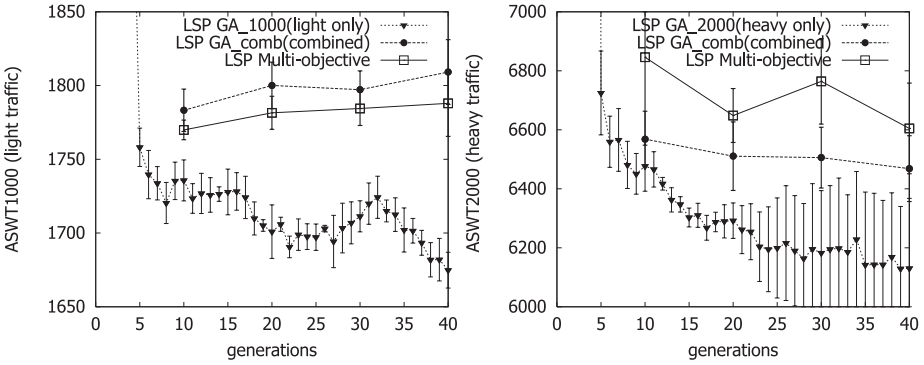

Fig. 3. Evolution processes of LSP, $ASWT_{1000}$(left) and $ASWT_{2000}$(right)

Figure 3 shows the evolution processes of LSP. Like as the case of EBP, that GA for one criterion, GA_{1000} and GA_{2000} are superior to others in their niches, and their $ASWT$ was decreasing over generations. However, especially in $ASWT_{1000}$ (left figure), the averaged performances of policies of GA_{comb} and GA_{moop} were getting worse. This suggests that performance of $ASWT_{1000}$ was sacrificed for the improvement of $ASWT_{2000}$. In other words, both $ASWT_{1000}$ and $ASWT_{2000}$ couldn't be simultaneously improved in LSP framework.

Measured by two criteria : In Figure 4, the sets of solutions in a period of a trial of GA_{moop}, LSP and EBP, are plotted. Triangles shows the sets of performances $(x, y) = (ASWT_{1000}, ASWT_{2000})$ of EBP, generation 10 and generation

80. From the figure, both criterions are simultaneously improved. On the other hand, circles show the set of performances of LSP, generation 10 and generation 40. The improvement is little, and the slight slide to right($ASWT_{1000}$ worse) and down($ASWT_{2000}$ better) can be observed.

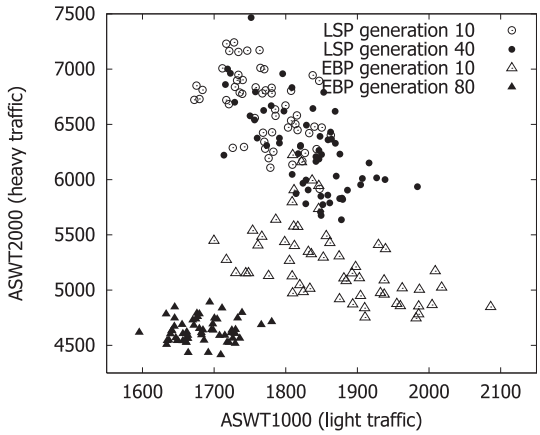

Fig. 4. Performance plot of solutions in a period of GA_{moop}

4.2 Performance Comparison of Policies Obtained

In this section, we focus on the performances of "elite" solutions of the evolved ones, instead of the averaged performances. About GA_{1000}, GA_{1500}, GA_{2000} and GA_{comb}, all individuals are carefully (30 times for each) re-evaluated per 10 generations. Their temporal elites are re-evaluated (180 times for each) and finally the elite of the trial is selected. By this selection, totally 40 solutions are given (LSP/EBP, five trials, four GAs).

About GA_{moop}, all individuals of the final generations are re-evaluated 30 times, and three elites, the solution with the best $ASWT_{1000}$, $ASWT_{2000}$, $ASWT_{\mathrm{comb}}$ are selected. By this selection, totally 30 solutions are given (LSP/ EBP, five trials, three solutions each) with few duplications.

All elites are again re-evaluated 300 times for the comparison.

Comparison of Four GAs : Figure 5(left) shows the performances $(x, y) = (ASWT_{1000}, ASWT_{2000})$ of LSP. Pareto curve is very usual as multi-objective problems. In more detail, there observed three groups of LSP. One is such as $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 5, 0, 0)$, they prefer to assign the lighter car, and perform well at the heavy traffic (right bottom) . One other is such as $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 0, 1, 0)$, they avoid assigning the near-followed car in order to maintain an adequate distance between cars. The last is such as $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, -1, 0, 0)$, they prefer to assign the heavier car in order to bias the loads to keep an adequate distance, and perform well at the light traffic (left top). We can conclude

that LSP has no condition-sensitive ability, and there is no versatile policy in LSP.

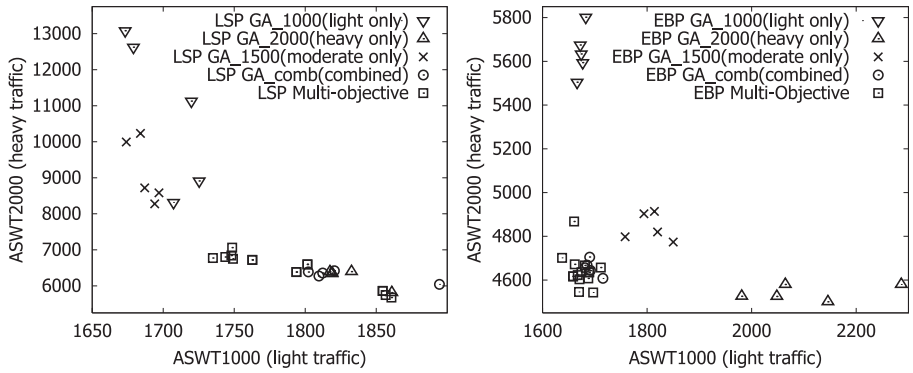

Fig. 5. Performance plot of elite solutions, LSP(left) and EBP(right)

Figure 5(right) shows the performances $(ASWT_{1000}, ASWT_{2000})$ of EBP. In contrast to the case of LSP, the elites of GA_{comb} and GA_{moop} are better as GA_{1000} in $ASWT_{1000}$ and better as GA_{2000} in $ASWT_{2000}$. This fact suggests that such EBP can automatically detect the current situation (for example from the fourth feature) and make decision depending on it, by its localizing mechanism. In other words, EBP has the enough condition-sensitive control ability.

GA_{1500} works not so bad, but its performance is worse than GA_{comb} and GA_{moop}. This fact suggests that the training in the two conditions helps the generalization ability of EBP.

Performance in Wide-Range Traffic : To show the generalization ability of policies, seven delegates are selected from elites, LSP/EBP elites of GA_{1000}, GA_{1500} (EBP only), GA_{2000} and GA_{moop}. They are re-evaluated in several traffic situations, from 750 persons/hour to 2250 persons/hour. Their performances of a traffic are measured by the overrun ratio of $ASWT$ to the best of the seven delegates in the traffic.

Figure 6(left) shows the performances of LSP. In this case, as the prediction from Figure 5(left), the elite from GA_{moop} is not versatile but just intermediate performance.

Figure 6(right) shows the performances of EBP. The elite from GA_{1000} works well at traffic is light, but the performance is increasingly worse when the traffic is heavier. The elite from GA_{2000} has the opposite problem. The elite from GA_{1500} performs not so bad for all conditions, and the elite from GA_{moop} performs better than it in almost all conditions.

Through the experiments, the localization ability of EBP for condition-sensitive control, and the generalization ability for unknown conditions has been shown. By the multi-objective optimization with $ASWT_{1000}$ and $ASWT_{2000}$,

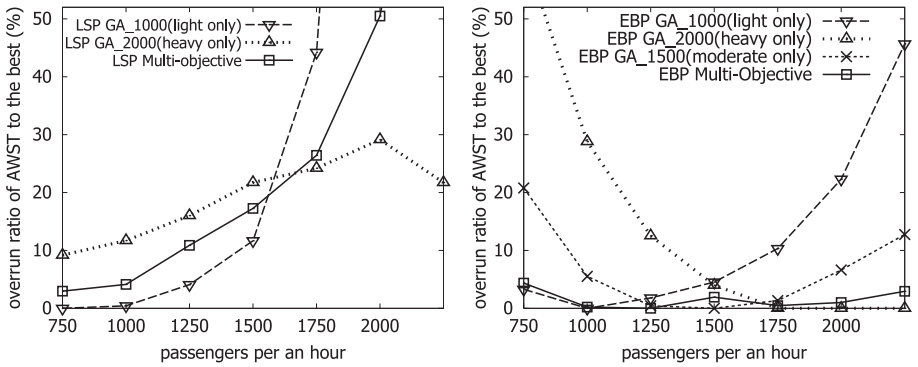

Fig. 6. Comparison of elites from GA_{1000}, GA_{1500}, GA_{2000} and GA_{moop}, LSP(left) and EBP(right), in wide-range traffics

the EBP is optimized as well as both of GA_{1000} and GA_{2000}, further, the EBP performs well also in the conditions that has not been experienced.

5 Conclusion

We presented a multi-objective optimization approach for learning condition-sensitive policy, and showed its effectiveness on the difficult problem of controlling multi-car elevators. The policy with parameters was evaluated in two traffic situations, and the objective functions were defined respectively, and a multi-objective optimization method was applied. We compared conventional linear-sum policy expression and exemplar-based policy (EBP) expression, and compared the multi-objective optimization approach and single-objective approach only for single situation. As the result, it was found that the EBP obtained by the multi-objective optimization worked adequately for wide-range situations. This fact suggests that EBP has the localization ability for condition-sensitive control, and the generalization ability for unknown conditions.

For practical use, the policy should be applicable to much wider situations, such as weekday and holiday, beginning of office hours, lunch hour and clock-out hours. For this demand, two subjects for future work exist. One is to improve the localization ability of EBP to detect the situation and the generalization ability to perform well in intermediate situations which are not tested. Another is to modify the multi-objective optimization method for such problem that has many objectives and the evaluation value is noisy.

Acknowledgments

The simulations were carried out on Fujitsu HPC2500, supercomputer systems in Kyoto University, and this work was supported in a part by a Grant-in-Aid for Young Scientists, Computing Service Group, ACCMS and IIMC, Kyoto University.

References

[Beielstein et al., 2003] T. Beielstein, C. P. Ewald, and S. Markon, "Optimal elevator group control by evolution strategies," *Genetic and Evolutionary Computation*, vol. 2724 of *LNCS*, 1963–1974, 2003.

[Deb et al., 2000] K. Deb, S. Arawal, A. Pratap and T. Meyarivan, "Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization : NSGA-II," *Parallel Problem Solving from Nature 6*, 849–858, 2000

[Ikeda, 2005] K. Ikeda, "Exemplar-Based Direct Policy Search with Evolutionary Optimization," *Congress on Evolutionary Computation*, pp. 2357–2364, 2005.

[Ikeda and Kobayashi, 2002] K. Ikeda and S. Kobayashi, "Deterministic Multi-Step Crossover Fusion: A Handy Crossover Composition for GAs," *Parallel Problem Solving from Nature*, 162–171, 2002

[Ikeda et al., 2006] K. Ikeda, H. Suzuki, S. Markon and H. Kita, "Evolutionary Optimization of a Controller for Multi-Car Elevators," *International Conference on Industrial Technology*, accepted, 2006

[Kim et al., 1998] C. B. Kim, K. A. Seong, H. Lee-Kwang and J. O. Kim, "Design and implementation of a fuzzy elevator group control system," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 28(3): 277–287, 1998.

[Kita et al., 2002] H. Kita, S. Markon, T. Sudo and H. Suzuki, "A Study on Control of Multi-Car Elevators," *SICE Symposium on Autonomous and Decentralized System*, 63–66, 2002 (in Japanese).

[Mimaki et al., 1999] K. Mimaki, S. Markon, H. Kita, Y. Komoriya and Y. Nishikawa, "Modeling and Analysis of Complex Traffic in Buildings," *Proc. IEEE SMC'99*, IV, 589–594, 1999.

[Obayashi and Sasaki 2004] S. Obayashi and D. Sasaki, "Multi-Objective Optimization for Aerodynamic Designs by Using ARMOGAs," *7th Intr. Conf. on High Performance Computing and Grid in Asia Pacific Region*, 2004

[Ono 1997] I. Ono, "A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover," *Proc. of 7th Intr. Conf. on Genetic Algorithms*, 246–253, 1997.

[Sudo et al., 2002] T. Sudo, H. Suzuki, S. Markon and H. Kita, "Effectiveness and control strategies of multi-car elevators for high-rise buildings," *TRANSLOG02*, 2002 (in Japanese).

[Takahashi et al., 2003] S. Takahashi, H. Kita, H. Suzuki, T. Sudo and S. Markon, "Simulation-based Optimization of a Controller for Multi-Car Elevators Using a Genetic Algorithm for Noisy Fitness Function," *Congress on Evolutionary Computation*, 1582–1587, 2003.

[Zhou et al., 2005] J. Zhou, T. Eguchi, K. Hirasawa, J. Hu, and S. Markon, "Elevator group supervisory control system using genetic network programming with reinforcement learning," *IEEE Congress on Evolutionary Computation*, vol 1, 336–342, 2005

A. Selection of the Best Feature Vector

When the feature vectors are given, an EBP selects the best one from them using exemplars as following procedure [Ikeda et al., 2006] (see Fig. 7).

1. Exemplars $E = \{(v_j^1, v_j^2)\}_j$ are given, where $v_j^1, v_j^2 \in \mathbb{R}^{N_{\text{features}}}$.
2. Feature vectors corresponding to possible cars, candidates, $C = \{w_c\}_c$ are given to be evaluated, where $w_c \in \mathbb{R}^{N_{\text{features}}}$.

3. An unbiased tournament for C is randomly created (the transitive law may not necessarily hold in this competition procedure).
4. A pair of competitors $w_{c^1} \in C$ and $w_{c^2} \in C$ are taken by following the tournament.
5. For each exemplar $(v_j^1, v_j^2) \in E$, the distance to the competitors $dist_j = |\frac{w_{c^1}+w_{c^2}}{2} - \frac{v_j^1+v_j^2}{2}|$ is calculated.
6. $E_{\mathrm{local}} \in E$, the top k_{NN} exemplars nearest within $dist_j$ are selected (k_{NN} is the localization parameter).
7. For each exemplar $(v_j^1, v_j^2) \in E_{\mathrm{local}}$, the direction $\overrightarrow{v_j^2 - v_j^1}$ and the inner product $IP_j = \overrightarrow{v_j^2 - v_j^1} \cdot \overrightarrow{w_{c^2} - w_{c^1}}$ are calculated. When $IP_j > 0$, the exemplar suggests that "w_{c^1} is better than w_{c^2}".
8. The number of exemplars in E_{local} for which $IP_j > 0$, *i.e.* $|\{(v_j^1, v_j^2) \in E_{\mathrm{local}}, IP_j > 0\}|$ is counted. When the number is larger than $|E_{\mathrm{local}}|/2$, w_{c^1} survives the competition (otherwise the opposite judgment is obtained).
9. After $|C|-1$ competitions have been completed, the winner is selected.

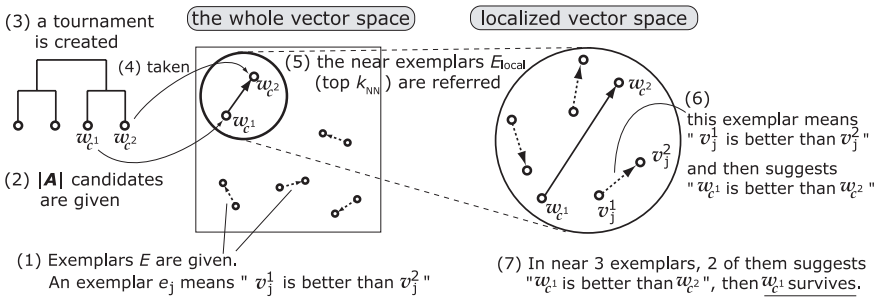

Fig. 7. Selection of the most preferable vector from candidates

B. Crossover operator for EBP-GA

The crossover operator produces a new set of exemplars using parents. In this paper, $N_{\mathrm{fusion}} = 36$ exemplars are newly created by a procedure called "fusion", and the rest exemplars are copied from parents. The crossover operation is performed as follows.

1. The parents E_{p1} and E_{p2} are given, and E_c is initialized as an empty set.
2. An exemplar $e_{p1,j} \in E_{p1}$ is selected randomly, and the exemplar $e_{p2,j*} \in E_{p2}$ nearest to $e_{p1,j}$ in E_{p2} is selected.
3. The rates $0 < \alpha < 1$ and $\beta = 1 - \alpha$ are fixed. For the exemplars $e_{p1,j} = (v_{p1,j}^1, v_{p1,j}^2)$ and $e_{p2,j*} = (v_{p2,j*}^1, v_{p2,j*}^2)$, an exemplar $e = (\alpha v_{p1,j}^1 + \beta v_{p2,j*}^1, \alpha v_{p1,j}^2 + \beta v_{p2,j*}^2)$ is newly created and added to E_c.
4. Steps 2. and 3., fusion procedure, are repeated N_{fusion} times.
5. An exemplar $e \in E_{p1} \cup E_{p2}$ is selected randomly. If $e \notin E_c$, e is added to E_c.
6. Step 5. is repeated until $|E_c| = N_{\mathrm{exemplars}}$.