Exemplar-based Control of Multi-Car Elevators and its Multi-Objective Optimization using Genetic Algorithm

Kokolo Ikeda[1], Hiromichi Suzuki[2], Hajime Kita[1] and Sandor Markon[2]
[1]Kyoto University
Yoshida-Nihonmatsu, Sakyo, Kyoto 606-8501, Japan
[2]Fujitec Co. Ltd,
Big Wing, Hikone, Shiga 522-8588, Japan
E-mail: kokolo@media.kyoto-u.ac.jp

Abstract: Multi-Car Elevator (MCE) is the novel system and attracts attention for improvement of transportation in high-rise buildings, but the design of controller for MCE is very difficult engineering problem. In this paper, the works of the authors are summarized: The control of MCE is divided to four phases and the assignment of a car to a hall-call is focused. The exemplar-based policy is used for the assignment, the policy is evaluated through discrete event simulation, and the parameter is optimized by genetic algorithm. To design traffic-sensitive controller, multi-objective optimization is utilized. Finally a policy visualization method is proposed for analyzing how the policy works.

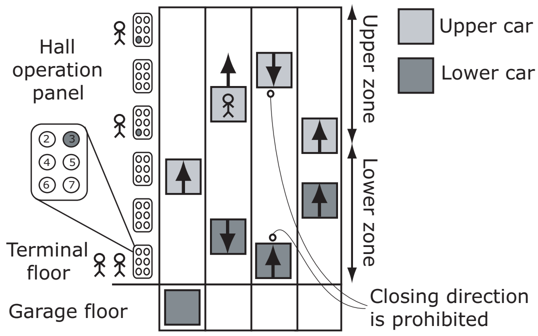

Figure 1. MCE system

1. Introduction

The control of cooperating elevator cars is known as an important task, and has been studied for many years. Such problem, "the elevator group control problem", is recognized as a difficult control task. Current commercial systems are controlled by using a combination of heuristic and artificial intelligence methods.

Multi-car elevators (MCEs), consisting of several cars in a single elevator shaft, are novel high-performance transportation systems [Kita et al., 2002]. Though they recently attract big attention, the problem is that the accumulated knowledge for single-car elevators is not readily applicable to MCEs, because their behaviors are distinctly different.

Simulation-based optimization, in which the policy of controller is represented by a function model and the parameters are optimized through a simulation, are often used for the optimization of MCE controller [Sudo et al., 2002]. We also have studied simulation-based optimization approaches for several years. An exemplar-based control framework and its optimization using Genetic Algorithm [Ikeda et al., 2006], multi-objective optimization for traffic-sensitive controller [Ikeda et al., 2007], and its visualization for analysis [Ikeda et al., 2007] have been proposed. In this paper these works are overviewed.

2. Target Problem

The almost same MCE system described in [Takahashi et al., 2003] is considered in our study (see Fig. 1). The lowest level of the building is assumed to be the sole point of entry/exit to the building, and thus experiences 10 times higher traffic. The other floors are assumed to be identical in terms of traffic demand. There are several elevator shafts in the buildings, and each shaft hosts two cars, which can only move vertically and are not allowed to approach each other simultaneously. The passengers register their destination floors not in the car but in the hall, and are guided to the car serving their need.

The goal of this problem is to develop the "effective" controller of MCE. For the purpose, how to measure the effectiveness, which type of controller is used, how to set its parameters and how to claim its superiority must be presented.

3. Approach

Our approach can be summarized as follows.

1. **Focus:** The whole control is divided to four components as [Takahashi et al., 2003]: "*Candidate selection*" when a new call occurs, the candidate cars for the call are nominated based on a zone operation; "*Feature Computation*" their feature vectors are calculated; "*Call Assignment*" the most preferable car is selected and assigned to the call; "*Transportation*" the cars in a shaft are controlled by a synchronizing rule with simultaneous start from terminals. In our study, call assignment is intensively focused.
2. **Policy representation:** As the procedure of selecting a car from candidates, **Exemplar Based Policy** (EBP) representation [Ikeda, 2005] is employed.
3. **Evaluation and optimization:** A policy is evaluated by a computer simulation, and parameters are optimized using a Genetic Algorithm (GA) [Ikeda et al., 2006]. Further, a policy is evaluated in multiple environments and *multi-objective GA* is employed in order to attain traffic-sensitive policy [Ikeda et al., 2007].
4. **Visualization:** For analysis of EBP, a visualization method **Local Gradient Fan Map** is proposed [Ikeda

et al., 2007].

In prior researches [Takahashi et al., 2003], the call assignment was also focused and the other components were almost same. In them, 11 feature values were calculated, linear-weighted policy representation was employed, and single environment was used for evaluation. On the other hand, in our research, 4 feature values are calculated, non-linear policy is employed, and multiple environments are used for evaluation. The features $[w_1^k, w_2^k, w_3^k, w_4^{(k)}]$ are as follows, where k is the car index. All the features are normalized so that almost all features are distributed in $[0,1]$.

- 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.
- w_3^k is the estimated delay time when the car pass through the call and the next car serves it.
- $w_4^{(k)}$ is the feature expressing the degree of current traffic. $w_4^{(k)}$ is common to all the cars.

4. Exemplar-Based Policy Representation

Exemplar-based policy (EBP) is a representation framework for decision making problem [Ikeda, 2005], and is expected to have rich expression ability of both generalization and localization. An advantage of EBP to the linear-weighted policy is the ability to control flexibly according to the current situation.

An EBP consists of a set of exemplars, and an exemplar is defined as the pair of feature vectors $(v_j^1, v_j^2) \in [0,1]^2$, 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 candidate feature vectors are given, the winning vector is decided by a procedure (See Appendix) and assigned to a given hall call.

5. Simulation-based Optimization

The same simulator used in [Takahashi et al., 2003] is employed for evaluation. 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. 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.

5.1 Multi Objective Optimization

When parameters are evaluated and optimized in a single traffic situation, there is no guarantee that the optimized one works adequately in the other situations. For this problem, 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. However, the rules are often written out by experts and so very expensive. So, it is preferable that one policy works adequately in various situations as much as possible.

In our study, multi-objective optimization approach [Deb et al., 2000] is employed to obtain such policy. The policy with parameters is evaluated in multiple situations, in this case from a light traffic to a heavy traffic, the objective functions are defined respectively, and multi-objective GA is applied by following procedure.

1. Parameters are fixed (see Table 1).
2. As the population, N_{pop} solutions are initialized. 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. Children are reproduced by applying the crossover operator N_{children} times. Parents (p_1, p_2) are randomly selected, and mixture of exemplars [Ikeda et al., 2006] is used by the crossover.
4. The evaluation values for each policy, $N_{\text{pop}} + N_{\text{children}}$ solutions, are evaluated using simulator. Both $ASWT_{1000}$ and $ASWT_{2000}$, $ASWT$ in case that the number of passengers are 1000 and 2000 per an hour, are calculated. 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.
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. From Step 3. to Step 6. are repeated N_{gnrs} times, after which the final result, trained MCE control policies with varieties are obtained.

Table 1. 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_{gnrs}	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	-

5.2 Optimization Result

We compare two styles of controllers, EBP and conventional policy LSP, and two styles of optimizer, single-objective and multi-objective GAs. Five independent trials are done for each setting, and elite solutions are selected and re-evaluated.

Fig. 3(top) shows the performances $(x, y) = (ASWT_{1000}, ASWT_{2000})$ of LSP. Pareto curve is very usual as multi-objective problems: If a policy is better in a objective, it is

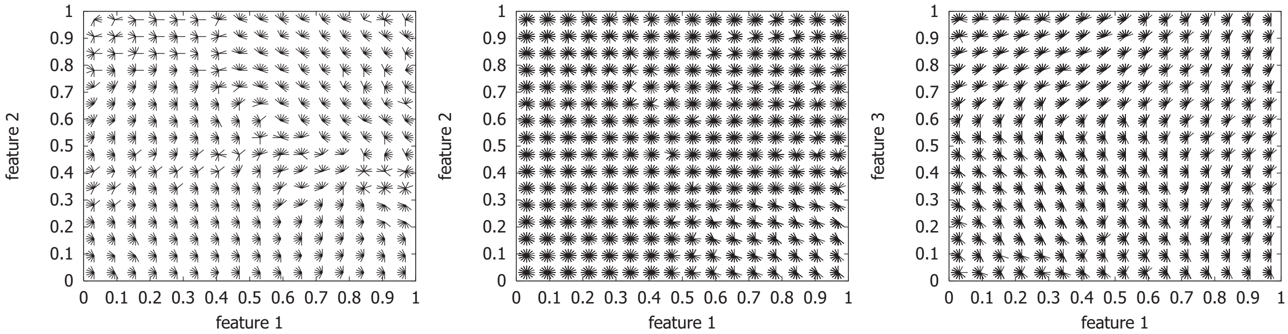

Figure 2. Local gradient fan map of a policy(left), Overlapped LGFMs(center, right)

worse in another objective. It can be concluded that LSP has no condition-sensitive ability.

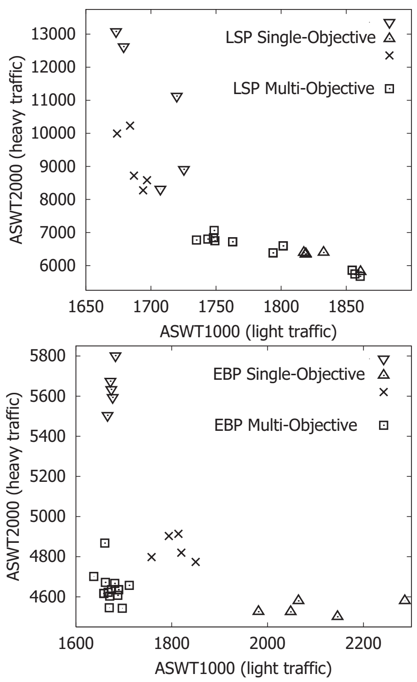

Figure 3. Performance plot of elite solutions, LSP(top) and EBP(bottom)

Fig. 3(bottom) shows the performances ($ASWT_{1000}$, $ASWT_{2000}$) of EBP. In contrast to the case of LSP, the elites of multi-objective GAs are better as the best ones of single-objective GAs, in both traffics. 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 the localizing mechanism. In other words, EBP has the enough condition-sensitive control ability.

6. Visualization of EBP

By the definition of EBP and the experimental results, it is suggested that EBP can detect the current situation and make decision depending on it, by the localizing mechanism. Finally we want to know how EBP works for this ability.

6.1 Local Gradient Fan Map

To assign the best car, EBP is used for evaluating feature vectors. The evaluation is performed between two candidates and the result is given not as the evaluation value but as which candidate wins. The key is whether feature values $[w_i]$ are evaluated positively or negatively at each condition. Local gradient fan map (LGFM) is the 2-dimensional map that shows the local preference about two feature dimensions. The drawing procedure for LGFM is as follows.

1. Two feature dimensions the user want to analyze are selected, and the value ranges are fixed. In this paper, $[0, 1]$ is used as the range.
2. The values of the other dimensions are constantly fixed. Generally, the values can be computed by a function.
3. The sample points are selected. In this paper, 16×16 points are selected like the lattice.
4. The number and the length of fan bones (n, l) are fixed. In our study $(n, l) = (16, 0.025)$ is used.
5. For each sample point (x_1, y_1), center of fan,
   (a) For each $\theta = \{0, 2\pi/n, ..., (n-1)2\pi/n\}$,
   (b) Let $(x_2, y_2) = (x_1 + l \cos\theta, y_1 + l \sin\theta)$.
   (c) Compare (x_1, y_1) and (x_2, y_2) by the policy.
   (d) If (x_2, y_2), top of bone, wins, draw a line between (x_1, y_1) and (x_2, y_2).

Fig. 2(left) shows an example of LGFM, where feature values w_1 and w_2 are selected to analyze, and (w_3, w_4) are fixed to $(0.3, 1.0)$. Around $(w_1, w_2) = (0.65, 0.1)$, the fans turn left, *i.e.* the direction w_1 is smaller. This fact means that the policy prefers the early-arriving car. Around $(w_1, w_2) = (0.65, 0.4)$, the fans turn the direction w_2 is larger, meaning that the policy prefers the heavier car. And around $(w_1, w_2) = (0.65, 0.9)$, the fans turn the direction w_1 and w_2 are larger.

6.2 Overlapping LGFM

Fig. 2(left) shows the complex strategy of an EBP trained by GA_{moop}. However, not all fans are essential in fact. A large part of strategy is not tested and optimized. To know the essential strategy for problems to analyze, we use a method to extract the common strategy among a set of policies. Fig. 2(center) shows the five-overlapped LGFMs, of five EBPs trained by GA_{moop}.

In almost all area, fans don't turn any one direction. This fact means that the five policies have no common strategy in such area. On the other hand, around $(w_1, w_2) = (0.9, 0.2)$, fans turn the direction w_1 and w_2 are smaller, though it is a bit wider. This fact means that all the five policies prefer the early-arriving and lighter car in this area, and the strategy is common, perhaps because it is essential.

6.3 Analysis using Overlapping LGFM

An example of analysis for gained EBPs is presented here. Fig. 2(right) shows the five-overlapped LGFM, where feature values w_1 and w_3 are selected to analyze, and (w_2, w_4) are fixed to $(0.1, 0.4)$, comparatively lighter situation.

Though this map is five-overlapped, fans turn clear directions. The tendency divides into two sections :

- In $(w_1, w_3) \in [0.0, 0.6] \times [0.7, 1.0]$, the car which w_3 is larger is preferred. Consider two cars $(w_1, w_3) = (0.0, 0.1)(A)$ and $(0.1, 1.5)(B)$. This situation means that A is the earliest arriving car, B is soon arriving car, and the next (third earliest) car is far. In this case, B is assigned to avoid bunching.
- In other area, the car which w_1 is smaller is preferred.

This strategy is reasonable, and such analysis can be made by checking other dimensions and other parameters.

7. Conclusion

In this paper we presented the summary of our studies in these years for multi-car elevators (MCE). We focused on the car assignment problem of MCE, then proposed exemplar-based policy (EBP) representation and its optimization. For obtaining traffic-sensitive controller, a multi-objective optimization method was applied. We had shown the superiority of EBP to conventional methods, further we proposed a visualization method for EBP, to show the reasonability of it.

References

[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 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*, 2006

[Ikeda et al., 2007] K. Ikeda, H. Suzuki, S. Markon and H. Kita, "Designing Traffic-Sensitive Controllers for Multi-Car Elevators through Evolutionary Multi-Objective Optmization," *Evolutionary Multi-Objective Optimization, Springer LNCS*, pp. 673-686, 2007

[Ikeda et al., 2007] K. Ikeda, H. Suzuki, S. Markon and H. Kita, "Traffic-Sensitive Controllers for Multi-Car Elevators; Design, Multi-Objective Optimization and Analysis," *SICE Annual Conference*, 2007

[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).

[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.

Appendix
Selection of the Best Feature Vector

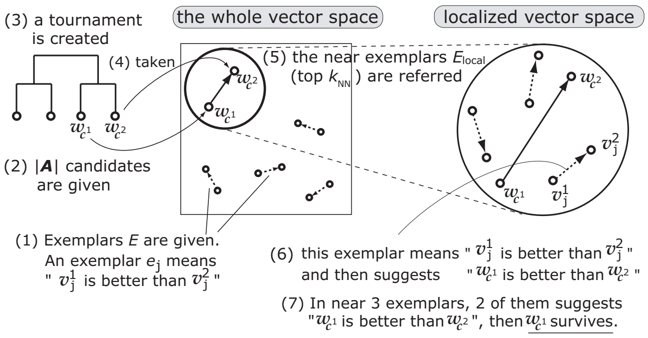

Figure 4. Selection of the most preferable vector

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. 4).

1. Exemplars $E = \{(v_j^1, v_j^2)\}_j$ are given, where $v_j^1, v_j^2 \in \mathbb{R}^{\mathbb{N}_{\text{features}}}$.
2. Feature vectors corresponding to possible cars, candidates, $C = \{w_c\}_c$ are given to be evaluated, where $w_c \in \mathbb{R}^{\mathbb{N}_{\text{features}}}$.
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. $E_{\text{local}} \in E$, the top k_{NN} exemplars nearest within $dist_j$ are selected (k_{NN} is the localization parameter).
7. For each exemplar $(v_j^1, v_j^2) \in E_{\text{local}}$, 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.* $|\{(v_j^1, v_j^2) \in E_{\text{local}}, IP_j > 0\}|$ is counted. When the number is larger than $|E_{\text{local}}|/2$, w_{c^1} survives the competition (otherwise the opposite judgment is obtained).
9. After $|C|$–1 competitions have been completed, the winner is selected.