Multi-objective optimization in service systems

Tad Gonsalves, Kei Yamagishi, Kiyoshi Itoh
Department of Information and Communication Sciences
Faculty of Science & Technology
Sophia University, 7-1 Kioicho
Chiyoda-ku, Tokyo, 102-8554 Japan
{t-gonsal, yamagi-k, itohkiyo}@sophia.ac.jp

Journal of Digital Information Management

ABSTRACT: *Muti-objective optimization deals with the simultaneous optimization of two or more conflicting objective functions in real-life systems. This paper deals with the multi-objective optimization in service systems. The goal of service systems is to provide cost-efficient service to customers, while at the same time, reducing the customer waiting time for service. In general, a low cost in system operation leads to longer waiting times, while a higher cost in system operation leads to shorter waiting times. The two objectives – service cost (operational cost) and waiting time (customer satisfaction) are, therefore, conflicting in nature. We use the novel Multi-Objective Particle Swarm Optimization (MOPSO) algorithm to optimize the two conflicting objective functions simultaneously. MOPSO is a fairly recent swarm intelligence meta-heuristic algorithm known for its simplicity in programming and its rapid convergence. The multi-objective optimization procedure is illustrated with the example of a practical service system. MOPSO produces a family of well-spread Pareto fronts for the two objective functions in the practical service system.*

Categories and Subject Descriptors
E. 5 [Files]; Optimization; I.3.5 Computational Geometry and Object Modeling

General Terms: Multi object optimization, Swarm optimization, Evolutionary algorithms

Keywords: Service systems, Service cost, Simulation optimization, Multi-objective Particle swarm optimization

Received: 19 October 2001; Revised: 21 December 2009; Accepted: 30 December 2009

1. Introduction

The goal of practical service systems such as banks, hospitals, restaurants, theme parks, reservation counters, etc. is to provide cost-efficient service to customers taking into consideration customer satisfaction and the cost of providing service. In general, a low cost in system operation leads to a poor customer satisfaction, while a high level of customer satisfaction demands higher cost in system operation. The two objectives are conflicting, making it an ideal situation for the application of multi-objective optimization techniques.

Customer satisfaction, of late, has become a major issue in marketing research and a number of customer satisfaction measurement techniques have been proposed (Hanan, Karp, 1991). Increasing efforts have been made to analyze the causes of customer dissatisfaction and suggest remedies (Taylor, 1994). In service systems, nothing can be as detrimental to customer satisfaction as the experience of waiting for service. Waiting is frustrating, demoralizing, agonizing, aggravating, annoying, time consuming, and incredibly expensive (Maister, 1985). For customers, waiting for service is a negative experience (Scotland, 1991), leading to a poor evaluation of the quality of service (Clemmer, Schneider, 1989; Davis, Heineke, 1998). The level of customer satisfaction can be raised by reducing the waiting time; however, this leads to the increase in cost due to the hiring of greater number of personnel to provide service. We propose an evolutionary multi-objective optimization scenario to find the optimal trade-off between the two objective functions.

Evolutionary algorithms are inspired by biological processes which are at work in nature. The Genetic algorithm (GA) (Davis, 1991; Goldberg, 1989; Koza, 1992; Mitchell, 1996) modeled on the Darwinian evolutionary paradigm, is the oldest and the best known Evolutionary Algorithm. It mimics the natural processes of selection, cross-over and mutation to search for optimal solutions in massive search spaces. Another very recent algorithm belonging to the class of biologically inspired methods is the Particle Swarm Optimization (PSO) (Kennedy, Eberhart, 1995; 2001). PSO imitates the social behavior of insects, birds or fish swarming together to hunt for food. PSO is a population-based approach that maintains a set of candidate solutions, called particles, which move within the search space. During the exploration of the search space, each particle maintains a memory of two pieces of information: the best solution (*pbest*) that it has encountered so far and the best solution (*gbest*) encountered by the swarm as a whole. This information is used to direct the search. PSO has been found to be successful in a wide variety of optimization tasks (Englebrecht, 2005; Kennedy, Eberhart, 2001).

Evolutionary Algorithms are primarily designed for the optimization of a *single* objective. However, in recent years, their application to multi-objective optimization has given rise to the Evolutionary Multi-Objective Optimization (EMO) research discipline (Zitzler, 1999). A variety of EMO algorithms have been developed. NSGA-II (an extension of GA) (Deb, et al., 2000) and MOPSO (an extension of PSO) are the two well-known algorithms used in EMO. Evolutionary algorithms offer a particularly attractive approach to multi-criteria optimization because they are effective in high-dimensional search spaces (Parsopoulos, Vrahatis, 2002). PSO seems particularly suitable for multi-objective optimization mainly because of the high speed of convergence that the algorithm presents for single-objective optimization (Coelho, et al., 2004, Mostaghim, Teich, 2003). MOPSO algorithms are extensions of the single-optimization PSO for solving multi-objective optimization problems (Alvarez-Benitez, et al., 2005; Coelho, Lechunga, 2002; Coelho, et al., 2004; Fieldsend, Singh, 2002, Hu, Eberhart,

2002; Hui, et al., 2003; Zitzler, 1999). In this study, we show how to apply MOPSO in simultaneously optimizing service cost and customer satisfaction due to waiting. The number of personnel to be assigned to each of the service stations in the service system and the service time per customer are the decision variables. Although there are bounds on the decision variables, the number of combinations even for a modest service system is prohibitively large. The problem, therefore, is a typical combinatorial optimization problem that cannot be solved by an exhaustive search. The fast converging MOPSO is a suitable algorithm for this problem. The multi-objective optimization procedure is illustrated with the example of a practical service system. MOPSO produces a well-spread Pareto front (the curve representing a trade-off relationship between the two objective functions) for the two conflicting objective functions in the practical service system.

2. Modelling of Service Systems

In this section, we define the simulation model of a real-world service system, the two objective functions and the decision variables.

2.1 Static model

We illustrate the simulation optimization technique using MOPSO, by means of a practical service system example shown in Figure. 1. The small clinic service system is made up of seven different "contexts" represented by rectangles in the diagram. Reception, Diagnosis 1, Diagnosis 2, Prescription, Medical tests, Prescription and Accounts are the respective contexts. These contexts are essentially the activities, or the service stations at which service is provided to the customers. The service-providing personnel include doctors, nurses, medical technicians, physiotherapists, pharmacists, etc. These, together with the rest of the resources are labeled above and below the contexts. The modelling details of service systems are found in (Hasegawa, et al., 2001). Discrete event simulation of the system provides with statistics like average waiting time which are used to compute the objective functions.

2.2 Dynamic model

The operational view of the collaborative system is provided by the discrete event simulation (DES). Discrete event simulation (Banks, Carson II, 1984; Fishman, 1978) is event-driven, an event being an occurrence that changes the state of the system. In a discrete event system, the events take place in discrete steps of time. Simulation proceeds as time-keeping of the series of events.

EXTEND SIM is a well-known DES package extensively used in academia and industry (http://www.extendsim.com/). It consists of several libraries containing simulation blocks. The simulation model is built by importing the blocks and connecting them together. EXTEND SIM is also provided with animation which helps to visually verify the operation of the model under consideration. The operation of the system can be simulated for any desired period of time. At the end of the stipulated time, the simulation stops and the software package compiles a report of the simulation. The report gives the operational parameters of the system, such as average queue length, average waiting time in the queue, server utilization, etc. These parameters are used in computing the waiting costs and constraints.

3. Multi-Objective Optimization

Most real-world optimization problems have multiple objectives which are often conflicting. The goal of multi-objective optimization (MOP) is to optimize the conflicting objectives simultaneously. In this section, we define the general form of a MOP and Pareto dominance for identifying optimal solutions. Towards the end of the section, we describe the use of Evolutionary Algorithms to solve MOP problems.

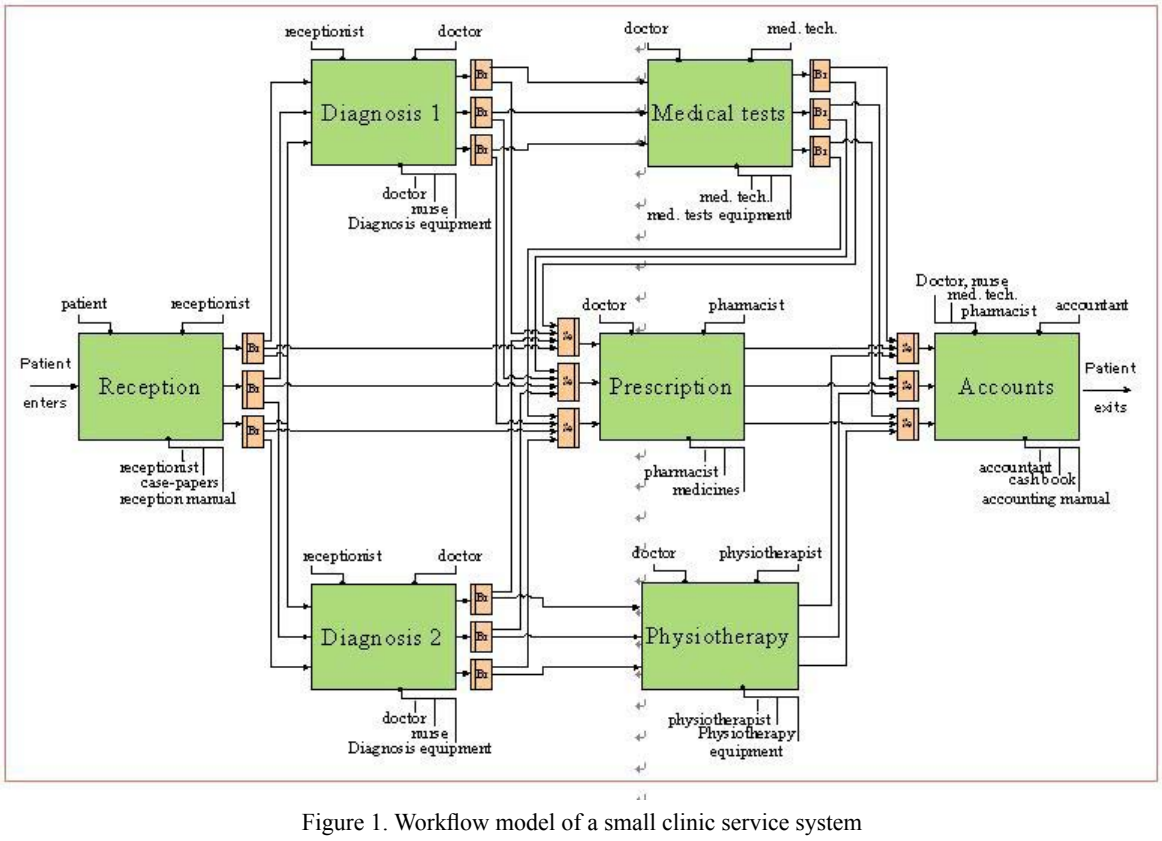

Figure 1. Workflow model of a small clinic service system

3.1 MOP Formulation

In general, a multi-objective minimization problem with M decision variables and N objectives can be stated as:

$$\text{Minimize } f_i(x) = 1, ..., N \quad (1)$$

where $\mathbf{x} = (x_1, ..., x_m) \in X$

$$\text{subject to : } \left.\begin{array}{ll} g_j(x) = 0 & j = 1, ..., M \\ h_k(x) \le 0 & k = 1, ..., k \end{array}\right\} \quad (2)$$

Here, f_i is the i^{th} objective function, $\mathbf{x}$ is the decision vector that represents a solution and X is the variable or parameter space. The functions g_j and h_k represent the equality and the inequality constraints, respectively. The desired solution is in the form of a "trade-off" or compromise among the parameters that would optimize the given objectives. The optimal trade-off solutions among the objectives constitute the Pareto front. MOP deals with generating the Pareto front, which is the set of non-dominated solutions for problems having more than one objective. A solution is said to be non-dominated if it is impossible to improve one component of the solution without worsening the value of at least one other component of the solution. The goal of multi-objective optimization is find the true and well-distributed Pareto front consisting of the non-dominated solutions.

3.2 Pareto Dominance

Most multi-objective optimization algorithms use the concept of domination. In these algorithms two solutions are compared on the basis of whether one solution dominates the other or not. Assume that there are M objective functions to be optimized and the problem is one of minimization. A solution $x^{(1)}$ is said to dominate the other solutions $x^{(2)}$, if conditions (1) and (2) are both true (Deb, 2001).

1. The solution $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives, or $f_j(x^{(1)}) \tilde{} f_j(x^{(2)})$ for all $j = 1, 2, \ldots M$.
2. The solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective, or $f_j(x^{(1)}) < f_j(x^{(2)})$ for at least one j belonging to $\{1,2,..M\}$.

If either of the above conditions is violated, the solution $x^{(1)}$ does not dominate the solution $x^{(2)}$. The non-dominated solutions give rise to a Pareto front. The points on the front are used to select a particular combination of functions that are in a trade-off balance.

Figure 2 illustrates the concept of Pareto dominance for the minimization of two objective functions. Solution S is dominated by solution P in the f_1 objective, while solution T is dominated by solution R in the f_2 objective. The solutions P, Q, R are Pareto-optimal solutions since none of them is dominated by any other solutions.

4. MOP in service systems

4.1 Service Costs

The servers in the system are often called "Perspectives". Service systems consist of different groups of Perspectives employed to provide service. For instance, the Perspectives in a clinic include groups of doctors, nurses, laboratory technicians, therapists, pharmacists, etc. If the service cost per unit time per personnel in a group assigned to a context is S_C, then the total service cost of the context is:

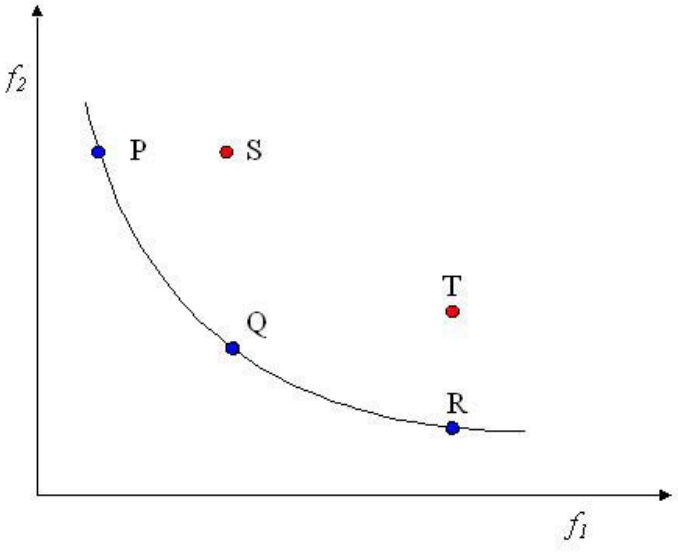

Figure 2. Pareto Dominance (min-min problem)

$$\text{Service cost} = \sum_{j=1}^{m} N_{Pj} S_{Cj} \quad (3)$$

where N_{Pj} is the number of Perspectives in the j^{th} group, and, m is the number of groups of Perspectives providing service at the given context.

Summing over the costs of all the contexts in the system, the total service cost (objective function f_1) is:

$$f_1 = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{Pij} S_{Cij} \quad (4)$$

where n is the total number of contexts in the system.

4.2 Waiting time

The customers (patients) queue in front of the respective contexts till the server becomes available. If Q_{Li} is the average number of customers in queue for Q_{Ti} period of time, then the weighted average of the waiting time (objective function f_2) is:

$$f_2 = \sum_{i=1}^{n} Q_{Li} Q_{Ti} \Big/ \sum_{i=1}^{n} Q_{Li} \quad (5)$$

The two functions are opposed to one another. An increase in one decreases the other and vice-versa. The MOPSO algorithm finds a trade-off between the two.

4.3 Decision variables

The number of professionals working at each context and the average service time of the context are the decision variables directly related to the cost function. In practical service systems, both these variables have lower as well as upper bounds. The capacity N_p of the server represents the number of personnel or the number of pieces of equipment that are assigned to a given context.

$$N_{P<} \leq N_P \leq N_{P>} \quad (6)$$

where $N_p<$ and $N_p>$ are the lower and the upper bounds, respectively.

Further, each context has an appropriate service time that is usually drawn from an exponential distribution. The service time limits can be expressed as:

$$t_< \leq t \leq t_> \quad (7)$$

where $t<$ and $t>$ are the lower and the upper bounds, respectively.

5. Evolutionary Algorithms for Multi-objective Optimization

Evolutionary Algorithms (EA) seem to be especially suited to MOP problems, due to their abilities to search simultaneously for multiple Pareto optimal solutions and to perform better global searches of the search space (Michell, 1996). Many evolutionary algorithms have been developed for solving MOP. Examples are: GA (Holland, 1992), NSGA-II (Deb, et al., 2000), a variant of NSGA (Non-dominated Sorting Genetic Algorithm) (Srinivas, Deb, 1994); SPEA2 (Zitzler, et al., 2001) which is an improved version of SPEA (Strength Pareto Evolutionary Algorithm); and PAES (Pareto Archived Evolution Strategy). These EAs are population-based algorithms that possess an in-built mechanism to explore the different parts of the Pareto front simultaneously. The Particle Swarm optimization (PSO), which was originally designed for solving single objective optimization problems, is also extended to solve multi-objective optimization problems. Among those algorithms that extend PSO to solve multi-objective optimization problems are Multi-objective Particle Swarm Optimization (MOPSO) (Alvarez-Benitez, et al., 2005; Coelho, Lechunga, 2002; Coelho, et al., 2004; Fieldsend, Singh, 2002, Hu, Eberhart, 2002; Hui, et al., 2003; Zitzler, 1999), Non-dominated Sorting Particle Swarm Optimization (NSPSO) (Li, 2003) and the aggregating function for PSO (Knowles, Corne, 2000).

5.1 Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) algorithm conducts a search using a population of individuals. The individual in the population is called the particle and the population is called the swarm. The performance of each particle is measured according to a predefined fitness function. Particles are assumed to "fly" over the search space in order to find promising regions of the landscape. In the minimization case, such regions possess lower functional values than other regions visited previously. Each particle is treated as a point in a d-dimensional space which adjusts its own "flying" according to its flying experience as well as the flying experience of the other companion particles. By making adjustments to the flying based on the local best (pbest) and the global best (gbest) found so far, the swarm as a whole converges to the optimum point, or at least to a near-optimal point, in the search space.

The notations used in PSO are as follows: The ith particle of the swarm in iteration t is represented by the d-dimensional vector, $x_i(t) = (x_{i1}, x_{i2},\ldots, x_{id})$. Each particle also has a position change known as velocity, which for the ith particle in iteration t is $v_i(t) = (v_{i1}, v_{i2},\ldots, v_{id})$. The best previous position (the position with the best fitness value) of the ith particle is $p_i(t\text{-}1) = (p_{i1}, p_{i2},\ldots, p_{id})$. The best particle in the swarm, i.e., the particle with the smallest function value found in all the previous iterations, is denoted by the index g. In a given iteration t, the velocity and position of each particle is updated using the following equations:

$$v_i(t) = wv_i(t-1) + c_1r_1(p_i(t-1) - x_i(t-1)) + c_2r_2(p_g(t-1) - x_i(t-1)) \quad (8)$$

and

$$x_i(t) = x_i(t-1) + v_i(t) \quad (9)$$

where, $i = 1, 2,\ldots,N_p$; $t = 1, 2,\ldots,T$. N_p is the size of the swarm, and T is the iteration limit; c_1 and c_2 are positive constants (called "social factors"), and r_1 and r_2 are random numbers between 0 and 1; w is the inertia weight that controls the impact of the previous history.

5.2 Multi-objective Particle Swarm Optimization (MOPSO)

The Multi-Objective Particle Swarm Optimization (MOPSO) is an extension of the single objective Particle Swarm Optimization (PSO). In PSO, gbest and pbest act as guides for the swarm of particles to continue the search as the algorithm proceeds. The main difficulty in MOPSO is to find the best way of selecting the guides for each particle in the swarm. This is because there are no clear concepts of pbest and gbest that can be identified when dealing with a set of multiple objective functions. Our algorithm is similar to the ones described in (Alvarez-Benitez, et al., 2005; Coelho, Lechunga, 2002; Coelho, et al., 2004). It maintains an external archive A, containing the non-dominated solutions found by the algorithm so far.

The algorithm begins with the initialization of an empty external archive, A (line 1 in Figure 3). The positions (x_i) and velocities (v_i) of a swarm of N particles are initialized randomly (line 2). The number of Perspectives assigned to each context and the service time of the contexts are randomly generated within the given bounds imposed by the management. The operation of the service system is simulated by means of the discrete event simulator and the values of f1 (service cost) and f2 (average waiting time) for each particle are computed using equations 2 and 3, respectively. The initial position of each particle is considered to be its personal best ($P_i = x_i$) as well as the global best ($G_i = x_i$).

At each iteration t the velocities of the particles are updated. The updated velocities are forced into the feasible bounds, if they have crossed the lower or the upper bounds. The particle positions are then updated using the updated velocities (lines 5-8). This is followed by the evaluation of the objective functions f1 and f2 for each of the particles (line 9). Any solutions which are not weakly dominated by any member of the archive are added to A (line 12) and any elements of A which are not dominated by xi are deleted from A.

The crucial parts of the MOPSO algorithm are selecting the personal and the global guides. If the current position of x_i weakly dominates P_i or if x_i and P_i are mutually non-dominating, then P_i is set to the current position (lines 15-17). Members of A are mutually non-dominating and no member of the archive is dominated by any x_i. All the members of the archive are, therefore, candidates for the global guide.

6. Simulation optimization results

Initially, the number of Perspectives in each of the Perspective groups, and the service time per context is randomly generated. These values are within the bounds set by the

```
1:  A := □⌀
2:  { x_i, v_i, G_i, P_i}  i = 1,…,N
3:  for t := 1 : G
4:     for i := 1 : N
5:        for k := 1 : K
6:           v_ik := wv_ik + r_1(P_ik − x_ik) + r_2(G_ik − x_ik)
7:           x_ik := x_ik + v_ik
8:        end
9:        y_i := f (x_i)
10:       if x_i  ⋦ u ∀u ∈ A
11:          A := {u ∈ A | u ≮ x_i }
12:          A := A U x_i
13:       end
14:    end
15:       if x_i  ≤  P_i
16:          P_i :=  x_i
17:       end
18:       G_i := select Guide(x_i , A)
19:  End
```

Figure 3. MOPSO Algorithm

management (hard constraints). The operation of the service system is simulated by means of the discrete event simulator. The simulation output produces various queueing statistics. The average number of customers in queues is used as the weights to compute the weighted average of the waiting time.

Several sensitivity analysis experimental runs reveal that $c_1 = 1$, $c_2 = 1$, $w(\text{initial}) = 0.9$ and $w(\text{final}) = 0.4$ yield highest performance. We experimented with a population size of 100 particles because it is a standard in multi-objective evolutionary algorithms that use an external archive. Figure 4 shows the Pareto front generated for the average inter-arrival time of 2 minutes. Since the arrival is rapid, the weighted average of the waiting time for service is relatively high. A similar Pareto front for average inter-arrival rate of 7 minutes is shown in Figure 5. The weighted average of the waiting time is relatively smaller. This is because the arrival is slower. The Pareto fronts obtained after 100 iterations are well-spread. The points on the front represent the optimal trade-off solutions.

The search space of our application problem can be estimated as follows. Assume, on an average, that there are 3 personnel in each of the 3 types of groups. Further, assume that the average service time of the contexts in the clinic system ranges from 10 to 30 minutes. Since there are 7 contexts in all, the search space = 37 x 37 x 37 x 207 = 1.34 x 1019. Even with a modest number of contexts in the service systems with not so long service time ranges, the search space explodes in size.

7. Conclusions and future work

In this paper, we introduced the Multi-Objective Particle Swarm Optimization (MOPSO) Evolutionary Algorithm to optimize two conflicting objectives- service cost and waiting time in service systems. The number of personnel assigned to each service station, their cost and the service time per customer are the decision variables. Although there are bounds on the decision variables, the number of combinations even for a modest service system is prohibitively large. The problem, therefore, is a typical combinatorial optimization problem that cannot be solved by an exhaustive search. Evolutionary algorithms are well suited for MOP problems because of their population based nature. Our experimental results with a practical service system show a near-perfect Pareto-front that represents a trade-off between the service cost and the waiting time which is directly related to

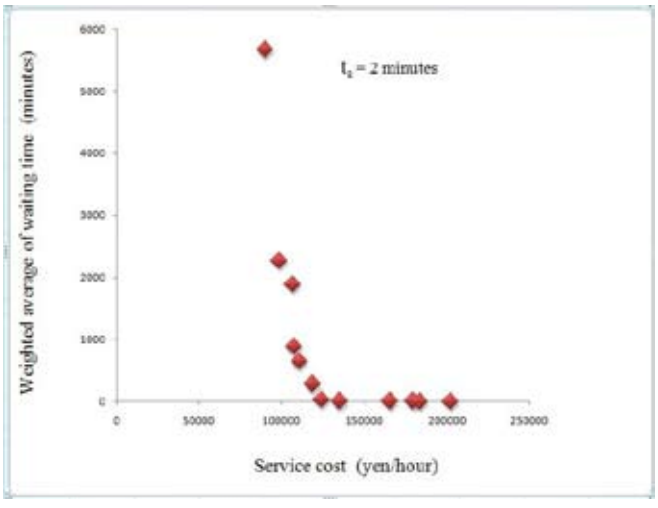

Figure 4. Pareto-front for service cost & waiting time (t_a = 2 minutes)

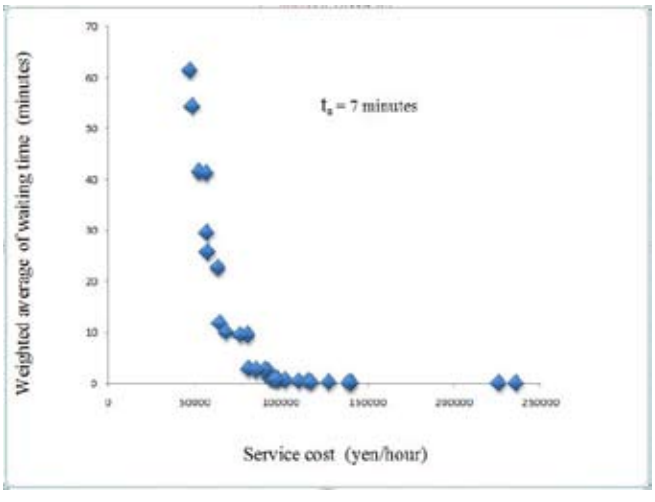

Figure 5. Pareto-front for service cost & waiting time (t_a = 7 minutes)

customer satisfaction. As an extension of this study, we would like to compare the performance of MOPSO to those of other EMO algorithms.

8. Acknowledgements

This research has been supported by the Open Research Center Project funds from "MEXT" of the Japanese Government (2007-20011).

References

[1] Alvarez-Benitez, J.E, Everson, R.M. Fieldsend, J.E (2005). A MOPSO algorithm based exclusively on Pareto dominance concepts. Lecture Notes in Computer Science, Evolutionary Multi-Criterion Optimization, Springer, Berlin / Heidelberg, 410: 459-473

[2] Banks, J., Carson II, J.S (1984). Discrete-Event System Simulation. Prentice-Hall, New Jersey

[3] Clemmer, E.C., Schneider, B (1989).Toward understanding and controlling customer dissatisfaction with waiting. Working paper 89-115, Marketing Science Institute: Cambridge, MA

[4] Coelho, C., Lechunga, M (2002). MOPSO: A proposal

for multiple objective particle swarm optimization. Proc. 2002 Congress on Evolutionary Computation, IEE Press 1051-1056

[5] Coelho, C., Pulido, G., Salazar, M (2004). Handling multi-objectives with particle swarm optimization. IEEE Trans. Evolutionary Computation, 8: 256-279

[6] Davis, L., Ed. (1991). The Genetic Algorithms Handbook, Van Nostrand Reinhold, New York

[7] Davis, M. M., Heineke, J (1998). How disconfirmmation, perception and actual waiting times impact customer satisfaction. International Journal of Service Industry Management 9 (1) 64-73

[8] Deb K., Agrawal S., Pratab A., Meyarivan, T (2000). A fast elitist nondominated sorting genetic algorithm for multiobjective optimization: NSGA-II. Proc. Parallel Problem Solving from Nature VI Conference 849–858

[9] Deb, K (2001). Multi-objective optimization using evolutionary algorithms, John Wiley & Sons, London

[10] Englebrecht, A.P (2005). Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, London

[11] Fieldsend, J.E., Singh, S (2002). A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence Proc. 2002 U.K. Workshop on Computational Intelligence, Birmingham 37–44

[12] Fishman, G.S (1978). Principles of Discrete Event Simulation, John Wiley & Sons, New York

[13] Fonseca, C.M., Fleming, P.J (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization. Proc. Fifth International Conference on Genetic Algorithms, San Mateo, CA

[14] Goldberg, D (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, MA

[15] Hanan, M, Karp, P (1991). Customer Satisfaction: How to Maximize, Measure, and Market Your Company's Ultimate Product. AMACOM/American Management Association, New York

[16] Hasegawa, A., Kumagai, S., Itoh, K (2000). Collaboration Task Analysis by Identifying Multi-Context and Colla-borative Linkage. CERA, 8 (1) 61-71

[17] Holland, J (1992). Adaptation in Natural and Artificial Systems, (2nd ed.), University of Michigan Press, Ann Arbor, MI

[18] Hu, X., Eberhart, R (2002). Multiobjective optimization using dynamic neighborhood particle swarm optimization. Proc. Congr. Evolutionary Computation (CEC'2002), 2: 1677–1681

[19] Hui, X., Eberhart, R.C., Shi, Y (2003). Particle swarm with extended memory for multiobjective optimization. Proc. 2003 IEEE Swarm Intelligence Symp., Indianapolis 193–197

[20] Kennedy, J., Eberhart, R.C (1995). Particle swarm optimization. Proc. IEEE Int. Conf. on Neural Networks, Piscataway, NJ 1942–1948

[21] Kennedy J., Eberhart, R.C (2001). Swarm Intelligence, Morgan Kaufmann

[22] Knowles, J., Corne, D (2000). Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computing 8: 149 -172

[23] Koza, J.R (1992). Genetic Programming, MIT Press, Cambridge, MA

[24] Lee, M.A., Esbensen, H (1996). Evolutionary algorithms based multiobjective optimization techniques for intelligent systems design. Biennial Conference of the North American Fuzzy Information Processing Society, CA 360-364

[25] Li, X (2003). A nondominated sorting particle swarm optimizer for multiobjective optimization. In: Lecture Notes in Computer Science, 2723, Proc. Genetic and Evolutionary Computation—GECCO 2003—Part I, E. Cantú-Paz et al., Eds.. Berlin, Germany, July 2003, 37–48

[26] Maister, D.H (1985). The psychology of waiting lines. In: Czepiel, J, Solomon, M. R. and Surprenant, C. F. (eds). The Service Encounter, Lexington Books: Lexington, MA 113-23

[27] Mitchell, M (1996). An Introduction to Genetic Algorithms. Cambridge, MA, MIT Press

[28] Mostaghim, S., Teich, J (2003). Strategies for finding good local guides in Multi-Objective Particle Swarm Optimization (MOPSO). Proc. 2003 IEEE Swarm Intelligence Symp., Indianapolis 26–33

[29] Parsopoulos, K.E., Vrahatis, M.N (2002). Particle swarm optimization method in multiobjective problems. Proc. 2002 ACM Symp. Applied Computing, Madrid 603–607

[30] Ray, T., Liew, K.M (2002). A swarm metaphor for multiobjective design optimization. Eng. Opt. 34 (2) 141–153

[31] Scotland, R (1991). Customer service: a waiting game. Marketing 1-3

[32] Srinivas, N., Deb, K (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. 2 (3) 221–248

[33] Taylor, S (1994). Waiting for service: the relationship between delays and evaluation of service. Journal of Marketing 58: 56-69

[34] Zitzler, E (1999). Evolutionary algorithms for multiobjective optimization: Methods and applications. Ph.D. dissertation, Swiss Fed. Inst. Technol. (ETH), Zurich

[35] Zitzler, E., Thiele, L (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. 3: 257–271

[36] Zitzler, E., Deb, K., Thiele, L (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput., 8 (2) 173–195

[37] Zitzler, E., Laumanns, M., Thiele, L (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. Proc. EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, T. Fogarty, Eds., Athens, Greece