PRACTICAL MULTI-OBJECTIVE SCHEDULING THROUGH SOFT COMPUTING APPROACH

Y.Shimizu[*]

Department of Production Systems Engineering
Toyohashi University of Technology
Toyohashi, Aichi 441-8580, Japan
Email: shimizu@tutpse.tut.ac.jp

Y.Tanaka

Department of Production Systems Engineering
Toyohashi University of Technology
Toyohashi, Aichi 441-8580, Japan
Email: tanaka@sc.tutpse.tut.ac.jp

ABSTRACT

Due to diversified customer demands and global competition, scheduling has been increasingly notified as an important problem-solving in manufacturing. Since the scheduling is considered at stage close to the practical operation in production planning, flexibility and agility in decision making should be most important in real world applications. In addition, since the final goal of such scheduling has many attributes, and their relative importance is likely changed depending on the decision environment, it is of great significance to derive a flexible scheduling through plain multi-objective optimization method. To derive a rational scheduling in this paper, we have applied a novel multi-objective optimization named $MOON^{2R}$ ($MOON^2$ of Radial-based function)by incorporating with simulated annealing. Finally, illustrative examples are provided to outline and verify the effectiveness of the proposed method.

INTRODUCTION

Recently, agile and flexible manufacturing has been highly required to deal with diversified customer demands and global competition. Under such circumstances, multi-objective scheduling has been increasingly notified as an important problem-solving in manufacturing. However, since the optimization of scheduling is seriously difficult to solve in itself, its multi-objective optimization has never been studied so much previously (Bogchi, 1999; Saym & Karabau, 1999; Murata, Ishibuchi & Tanaka, 1996; Tamaki, Nishino & Abe, 1999; Sakawa & Kubota, 2000). To work with the problem, in this paper, we will apply a novel approach of multi-objective optimization named $MOON^{2R}$, which is derived from $MOON^2$(Multi-Objective optimization with value function modeled by Neural Network) (Shimizu, 1999; Shimizu & Kawada, 1999). It can not only overcome the stiffness and shortcomings of the conventional methods, but also derive a best-compromise solution readily in the decision environment mentioned already. After giving a general procedure for solving the multi-objective scheduling by $MOON^{2R}$, illustrative examples will be provided to outline the proposed method, and to verify its effectiveness.

SOFT COMPUTING APPROACH FOR MULTI-OBJECTIVE SCHEDULING

Problem Formulation

Generally, we can describe a multi-objective scheduling problem as a multi-objective optimization problem (MOP) described below.

$$(p.1)\ \min\ \boldsymbol{f}(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_N(\boldsymbol{x})\}$$
$$\textit{subject to}\ \boldsymbol{x} \in X$$

where $\boldsymbol{x}$ denotes an n-dimensional decision variable vector, X a feasible region, and $\boldsymbol{f}$ an N-dimensional objective function vector some elements of which conflict with each other, and are incommensurable. It should be noted that the above formulation for scheduling refers to integer and/or mixed-integer programming problems (Bagchi,1999) whose combinatorial nature makes the solution process very complicated and time consuming (NP-hard/complete). Though the recent studies known as meta-heuristic such like multi-objective GA (Schaffer,1985; Fonseca & Fleming, 1993) and multi-objective SA (Czyzak & Jaszkiewicz, 1998) can deal with the problem somewhat, they can generate only the Pareto optimal solution set. In contrast to it, an approach proposed below ($MOON^{2R}$) can derive a unique solution that should be the best compromise of the decision maker (DM). This presents a great advantage for the flexible and agile engineering in real world.

[*]Address all correspondence to this author.

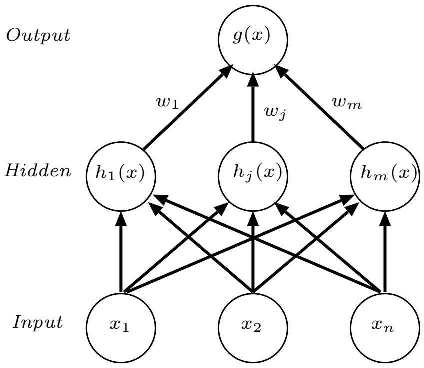

Figure 1. TRADITIONAL STRUCTURE OF RBFN.

General Framework for Practical Solution

Since $MOON^{2R}$ belongs to a prior articulation method in multi-objective optimization, we need identify a value function of the decision maker (DM) a priori. To improve such modeling stage, we introduced newly a radial-based function network (RBFN) instead of usual back propagation network (BPN) employed in $MOON^2$. That enables us to model the value function more readily depending on unsteady decision environment popular with scheduling problems. Due to the linear characteristic of RBFN, the computational load is considerably small compared with BPN. The traditional structure of RBFN is shown in Figure 1. There each component of input vector $\boldsymbol{x}$ feeds forward to the basis functions $\boldsymbol{h}$ whose outputs are linearly combined with the weight $\boldsymbol{w}$ to derive the output $g(\boldsymbol{x})$ as follows:

$$g(\boldsymbol{x}) = \sum_{j=1}^{m} w_j h_j(\boldsymbol{x}) \tag{1}$$

Using the training data set such like ($\boldsymbol{x}_i$, y_i) (i=1,c,p), sum squared error with a weight penalty term is minimized with respect to the weights (y_i denotes an observed output for input $\boldsymbol{x}_i$) .

$$C = \sum_{i=1}^{p} (y_i - g_i(\boldsymbol{x}_i))^2 + \sum_{j=1}^{m} \lambda_j w_j^2, \tag{2}$$

where λ_j,$(j = 1, \cdots, m)$ denotes regularization parameters.

To train the above RBFN, data regarding the relative preference of DM among the appropriate trial solutions is gathered through AHP (Analytic Hierarchy Process; Saaty,1980) like pair comparisons. That is, DM is asked to reply which he/she likes, and how much it is between every pair of the trial solutions. Such responses will be taken place by using linguistic statements, and later transformed into the score (Refer to Table 1). After doing such pair comparisons over k trial solutions in turn, we can obtain a pair comparison matrix whose i-j element a_{ij} represents the degree of preference of $\boldsymbol{f}^i$ compared with $\boldsymbol{f}^j$ (Refer to Table 2 appeared later in the example). After all, it provides totally k^2 training data for RBFN. That is, objective values of every pair, say, $\boldsymbol{f}^i$ and $\boldsymbol{f}^j$ becomes $2N$ inputs, and i-j element a_{ij} one output. We are possible to view thus trained RBFN as an implicit function, i.e. V_{RBF}: $(\boldsymbol{f}^i(\boldsymbol{x}), \boldsymbol{f}^j(\boldsymbol{x})) \in R^{2N} \rightarrow a_{ij} \in R^1$. Then noticing the following relation, we can rank the preference of any candidates in objective function space easily by the output of RBFN, a_{*R} calculated by fixing one of the input vector at an appropriate reference, say $\boldsymbol{f}^R$.

Table 1. CONVERSION TABLE.

Linguistic statement	a_{ij}
Equally	1
Moderately	3
Strongly	5
Demonstrably	7
Extremely	9
Intermediate judgements	2,4,6,8

$$\begin{aligned} V_{RBF}(\boldsymbol{f}^i, \boldsymbol{f}^R) &= a_{iR} \geq V_{RBF}(\boldsymbol{f}^j, \boldsymbol{f}^R) = a_{jR} \\ &\Rightarrow \boldsymbol{f}^i \succeq \boldsymbol{f}^j \end{aligned} \tag{3}$$

Now, the foregoing MOP (p.1) is possible to describe as follows.

$$(p.2) \;\; \max V_{RBF}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}^R) \;\; subject\ to\ \boldsymbol{x} \in X$$

Thus describing the multiple objectives into an overall one, we can apply a variety of optimization methods known previously, i.e., nonlinear programs, direct search methods, and even more meta-heuristic methods like GA, SA, TS, etc. Among them, SA is considered favorable due to certain combinatorial natures of scheduling problems. Now its application is straightforward since using V_{RBF}, we can evaluate any candidates under the multi-objectives once $\boldsymbol{x}$ is given.

In the above approach, since the modeling process of the value function is separated from the searching process, DM can carry out his/her tradeoff analyses at his/her own paces without worrying about the hurried/idle responses like the interactive MOP methods. In addition, since the required responses are simple and relative, DM's load in such interaction is very small. Moreover, modeling by RBFN can deal adaptively with the change of the decision environment that makes likely alter the preference of DM. Even in such a case, its retraining is easily taken place against the increase and decrease in the training data and basis from the foregoing one. These are particular advantages aiming at the agile and flexible decision making. After all, we can summarize the proposed solution method as follows.

1. Generate several candidates in objective function space.
2. Ask the preference of DM through pair comparison between every pair of the candidates.
3. Train RBF based on the above result. This provide a value function V_{RBF}.
4. Finally, apply SA to solve the problem (p.2).

ILLUSTRATIVE EXAMPLE

To examine the effectiveness of the foregoing approach, we solved multi-objective flow shop scheduling problems under two objective functions i.e. minimization of sum due time delay f_1 and total changeover cost f_2.

As a generic property of MOP (subjective decision problem), it is impossible to derive a preferentially optimal solution just by the mathematically provided conditions. Hence to verify the effectiveness of the method in the numerical experiments, we supposed the virtual DM whose preference on the problem is given as a utility function defined by

$$U(\boldsymbol{f}(\boldsymbol{x})) = \left\{ \sum_{i=1}^{N} w_i (f_i(\boldsymbol{x})/\hat{f}_i)^p \right\}^{1/p}, \ (p = 1, 2, \cdots) \quad (4)$$

where w_i denote a weight factor, $\hat{f}_i$ an appropriate nominal value, and p a parameter to specify the adopted norm respectively.

Moreover, we need characterize the virtual DMs more minutely to simulate their preference i.e., subjective judgment in their pair comparisons. That is, the degree of preference mentioned already is assumed to be given as

$$\begin{cases} a_{ij} = 1 + [\frac{8(U(\boldsymbol{f}^i) - U(\boldsymbol{f}^j))}{U(\boldsymbol{f}^{utop}) - U(\boldsymbol{f}^{nad})} + 0.5] & if\ U(\boldsymbol{f}^i) \geq U(\boldsymbol{f}^j) \\ a_{ij} = 1/a_{ji} & Otherwise \end{cases} \quad (5)$$

where $\boldsymbol{f}^{utop}$ and $\boldsymbol{f}^{nad}$ denote the utopia and nadir respectively.

Then among the trial solutions generated as shown in Figure 2, by equation (5), the pair comparison matrix of

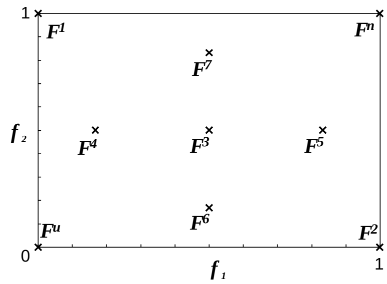

Figure 2. LOCATION OF TRIAL SOLUTIONS.

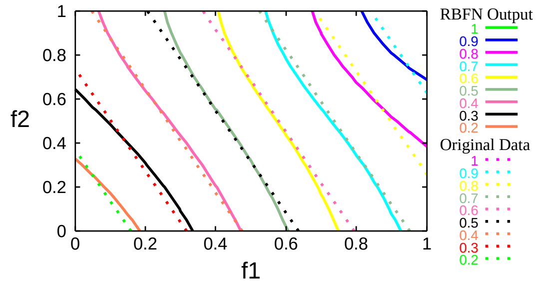

Figure 3. COMPARISON OF CONTOUR OF VALUE FUNCTION.

Table 2. PAIR COMPARISON MATRIX (p=1)

	F^u	F^n	F^1	F^2	F^3	F^4	F^5	F^6	F^7
F^u	1	9	3	7	5	3	7	4	6
F^n	1/9	1	1/7	1/3	1/5	1/7	1/3	1/6	1/4
F^1	1/3	7	1	4	3	1	4	2	3
F^2	1/7	3	1/4	1	1/3	1/4	1	1/3	1
F^3	1/5	5	1/3	3	1	1/3	3	1/2	2
F^4	1/3	7	1	4	3	1	5	2	4
F^5	1/7	3	1/4	1	1/3	1/5	1	1/4	1/2
F^6	1/4	6	1/2	3	2	1/2	4	1	3
F^7	1/6	4	1/3	2	1/2	1/4	2	1/3	1

the virtual DM will be given as Table 2 (Actually only the upper triangular part should be provided noticing the relation $a_{ij} = 1/a_{ji}$). Using the normalized values of those, we trained the RBFN to obtain the value function defined as equation (3).

We compared the contour lines of preference (indifference curves) between the supposed and $V_{RBF}(\boldsymbol{f}, \boldsymbol{f}^R)$ in Figure 3 when p=1. Except for the marginal regions, we confirmed that RBFN could model the supposed utility function fairly correctly.

Under thus identified value function and more, we solved three flow shop scheduling problems i.e.,

Table 3. COMPARISON OF NUMERICAL RESULTS(p=1).

	Overall objective func.	
$Kind of problem^*$	Reference	V_{RBF}
(1,1, 7)	168	168
(2,1,10)	1135	1135
(2,2,10)	119	119

*Numbers of (process:S, machine:M, job:J)

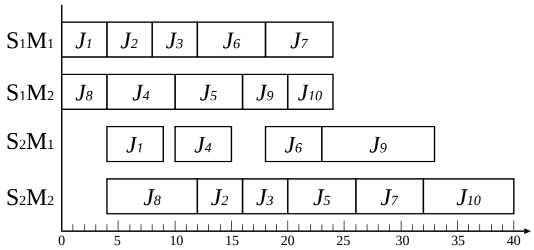

Figure 4. GANTT CHART OF (2,2,10) PROBLEM(p=1).

1. one process, one machine and 7 jobs
2. two processes, one machine and 10 jobs
3. two processes, two machines and 10 jobs.

Each objective function was specified by generating randomly the scheduling data within certain extents i.e., [1,10] for f_1 and [4,40,4] for f_2 respectively. As an optimization method, we applied the simulated annealing (SA) viewed as a randomized neighborhood search algorithm. It uses an analogy with the physical process of annealing, in which a pure lattice structure of a solid is made by heating up the solid in a heat bath until it melts, then cooling it down slowly until it solidifies into a low-energy stage. Presently, we adopted the insertion neighborhood method and, gave the tuning parameters as follow: initial temperature= 0.05; reduction rate of temperature=0.95; number of iteration=400.

In Table 3, we summarized some numerical results in comparison with the reference solution that is derived from the optimization under equation (4) directly, and viewed as a reference in the present consideration. Same results in every case ascertain that the proposed method can solve the problem correctly through accurate identification of the DM's preference by V_{RBF}.

CONCLUSION

To deal with the multi-objective scheduling, in this paper, we have proposed a practical approach characterized mainly by the modeling process of the value function by RBF and the application of SA. As a result, we can expect to deal with the unsteady decision environment popular with the practical scheduling problems. Illustrative examples are provided to outline the proposed method, and verify its effectiveness.

ACKNOWLEDGMENT

The authors thanks Dr. T. Kitajima for allowing us to use the program of SA.

REFERENCES

Bagchi, T. P., *Multiobjective Scheduling by Genetic Algorithms*, Kluwer Academic Publishers, USA 1999.

Czyzak, P. and A.Jaszkiewicz, *Pareto Simulated Annealing- A Metaheuristic Technique for Multiple-objective Combinatorial Optimization*, J. Multi-criteria Decision Analysis, Vol.7, pp.34-47 1998.

Fonseca, C. M. and P. J. Fleming, *Genetic Algorithm for Multi-Objective Optimization: Formulation, Discussion and Generalization*, Proc. 5th Int. Conf. on Genetic Algorithm and Their Applications, Chicago, pp.416-423 1993.

Murata, T., H. Ishibuchi and H. Tanaka, *Multi-Objective Genetic Algorithm and Its Applications to Flowshop Scheduling*, Computers Ind. Engng, Vol.30, No.4, pp.57-968 1996.

Saaty, T. L., *The Analytic Hierarchy Process*, McGraw-Hill, New York 1980.

Sakawa, M. and R.Kubota, *Fuzzy Programming for Multiobjective Job Shop Scheduling with Fuzzy Processing Time and Fuzzy Duedate through Genetic Algorithms*, Europ. J. Oper. Res., Vol.120, pp.393-407 2000.

Saym. S. and S.Karabau, *Bicriteria Approach to the Two-Machine Flow Shop Scheduling Probelm*, Europ. J. Oper. Res., Vol.113, pp.393-407 2000.

Schaffer, J. D., *Multiple Objective Optimization with Vector Evaluated Genetic Algorithm*, Proc. 1st Int. Conf. on Genetic Algorithm and Their Applications, Pittsburgh, pp.93-100 1985.

Shimizu, Y., *Multiobjective Optimization for Site Location Problems through Hybrid Genetic Algorithm with Neural Networks*, J. Chem. Engng, Japan, Vol.32, pp.51-58 1999.

Shimizu, Y. and A. Kawada, *Multiobjective Optimization Method with Value Function Modeled by Neural Networks*, Proc. Kansai Branch Symposium of SICE, T201, pp.120-123 1999 (in Japanese).

Tamaki, H., E. Nishino and S. Abe, *Modeling and Genetic Solution for Scheduling Problems with Regular and Non-Regular Objective Functions*, Trans. SICE, Vol.35, No.5, pp.662-667 1999 (in Japanese).