Proceedings of the 2000 Winter Simulation Conference
J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

SIMULATION OPTIMIZATION USING TABU SEARCH

Berna Dengiz
Cigdem Alabas

Department of Industrial Engineering
Gazi University
Celal Bayar Bulv. Maltepe 06570
Ankara/TURKEY

ABSTRACT

Investigation of the performance and operation of complex systems in manufacturing or other environments, analytical models of these systems become very complicated. Because of the complex stochastic characteristic of the systems, simulation is used as a tool to analyze them. The trust of such simulation analysis usually is to determine the optimum combination of factors that effect the considered system performance. The purpose of this study is to use a tabu search algorithm in conjunction with a simulation model of a JIT system to find the optimum number of kanbans.

1 INTRODUCTION

When investigating the performance and operation of complex systems in manufacturing or other environments, analytical models of these systems become very complicated. Because of the complex stochastic characteristic of the systems, simulation is used as a tool to analyze them. However, the major drawback of simulation for practical applications is that is time consuming.

To overcome the limitations of simulation, metamodeling was first proposed by Blanning (1975). Major benefits of metamodeling were summarized by Madu (1990). Yu and Popplewell (1994) presented a general review of metamodels in manufacturing. Barton (1992) reviewed different methods for choosing a functional form for the metamodel relationship such as polynomials, Taguchi models and generalized linear models. Madu and Kuei (1992) employed group screening and Taguchi models in the design of experiment stage of a multi-echelon maintenance float simulation. Regression metamodels for steady-state systems are used, and they can also represent dynamic behavior in response to several unexpected real-time events in manufacturing. Lin and Cochran (1990) analyzed the dynamic performance of a hypothetical multi-station, multi-server assembly line. Lin and Chiu (1993) modeled a manufacturing cell with a fixed flow pattern considering dynamic effects of machine breakdowns and job changes. Ozdemirel et al. (1996) proposed a knowledge-based system, namely design of experiments for simulation, that assist an experienced analyst. A general review on simulation optimization was given in Tekin and Sabuncuoglu (1998).

Recently, many studies using modern heuristic techniques for simulation optimization have been encountered. Bulgak and Sanders (1988) used modified simulated annealing to optimize buffer sizes in automatic assembly systems. Grefensette (1991) considered strategy acquisition with genetic algorithms. Stuckman (1988, 1990) has discussed the use of a particular Bayesian global search algorithm for optimizing a design via simulation. Stuckman (1991) has compared three classes of global search algorithms such as genetic algorithms and Bayesian/sampling algorithms for design optimization. Haddock et al. (1992) used simulated annealing to optimize parameter levels considering the total profit of an automatic production system. Hall and Bowden (1997) presented a comparative study of direct search methods such as tabu search (TS), evolution strategies, and the Nelder Mead Simplex algorithm for simulation optimization. Lutz et al. (1995) have built a simulation model of a manufacturing process and used TS, a heuristic procedure, to optimize buffer location and storage size in this manufacturing system. Dengiz et al. (2000) used a regression metamodel to optimize batch sizes in a Printed Circuit Board assemble line.

Due to the manufacturing companies interest, researchers started investigating the JIT philosophy and also much work has been done to find number of kanbans required in a JIT system. In general, a JIT system, if implemented properly, will result in increased productivity, reduced work-in-process (WIP), and higher product quality depending on the environmental factors of the JIT system. In JIT systems both the level of WIP and order lead-time are important performance parameters. Inventory control in

a JIT system is controlled by the number of kanbans allocated. Kanban, which is a card in Japanese is used to direct materials to workstations and passes information as to what and how much to produce (Wang and Wang 1991). Kimura and Terada (1981) describe the operation of kanban systems and examine the accompanying inventory fluctuations in a JIT environment. Rees et al. (1987) studied empirical approaches for setting kanban levels dynamically. Bitran and Chang (1987) presented a mathematical formulation of the kanban determination problem. The formulation assumes planning periods of known length and finds the minimum feasible number of kanbans. Deleersnyder et al. (1989) places the kanban determination problem into the context of the overall pull system implementation problem. Monden (1981) described the model in equation 1 for setting the number of kanbans for the Toyota Motor Company.

$$k_i = \frac{\tau_i D_i (1+\alpha)}{n_i} \tag{1}$$

where k_i is the number of kanbans for part type i, n_i is the container size, τ_i is the sum of the lead time, waiting time and kanban collecting time, and D_i is the average demand rate for part type i. While the demand rate is known on average, some variability does exist due to the order sequence at final assembly and drifts in demand. Because the safety factor, α, in equation 1 is to handle variability, the problem is the selection of α. Askin et al. (1993) proposed an economic approach for selecting k_i, and accordingly α. Their objective is to minimize the sum of inventory holding and backorder cost. They formulated a continuous time, steady-state Markov model to determine the optimal number of kanbans to use for each part type at each workcenter in a JIT system. The model selects the proper safety factor in each case.

Fukukawa and Hong (1993) proposed a mixed integer programming approach to examine many factors which play an important role in determining the number of kanbans in a JIT production system. Their objective function was to minimize inventory holding, outage and miscellaneous operating costs. Muckstadt and Tayur (1995) presented a heuristic approach to determine the number of kanbans. Aytug et al. (1996) determined the number of kanbans in a pull production system using a regression metamodel. Hurrion (1997) found the approximate optimum number of kanbans by using neural network metamodel.

The aim of this study is to find optimum number of kanbans in a JIT system using TS. The performance of TS algorithm is compared with the performance of a random search algorithm (RS) applied on the same problem. The example chosen to demonstrate this approach is a stochastic discrete-event system described by Aytug et al. (1996).

2 KANBAN-CONTROLLED SYSTEM

The example concerns the manufacture of two products, which are labeled Part7 and Part8. The production system includes two workcenters which are treated as black boxes (i.e. no specific flow patterns and machines are defined within the cells). The kanban controlled system considered in this study is shown in Figure 1. The products are manufactured in two adjacent workcenters. The first workcenter uses raw materials Part1, Part2, Part3, Part4 and produces two intermediate products, Part5 and Part6. The second workcenter gets the Part5 and Part6 products from workcenter 1, and produces the end products Part7 and Part8. Part5, Part6, Part7 and Part8 are produced in the manner given in Figure 1.

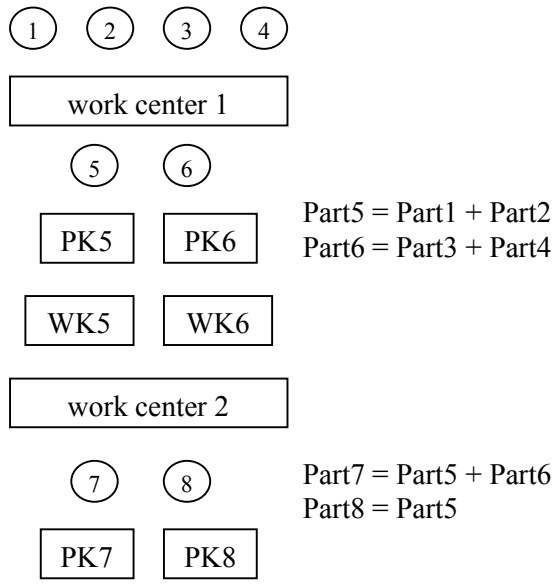

Figure 1: The Kanban-Controlled Manufacturing System

For each type of product at each output buffer, specific production kanbans defined as PK5, PK6, PK7, PK8, and withdrawal kanbans WK5, WK6 are used to signal requests for part transfers between the workcenters. The interarrival rate of the customer orders, order quantity and time delays in the system are given in Table 1.

Table 1: Input Distributions for the Problem

Interarrival rate of the customer orders		N(3, 0.6)
Order quantity		N(1,3)
Time Delays	Part5	N(1, 0.2)
	Part6	N(1, 0.2)
	Part7	N(2, 0.4)
	Part8	N(1, 0.2)

Problem assumptions are:

- An infinite supply of parts 1 - 4 is in the first workcenter
- Transfer time for production kanbans (PKs) is negligible
- Withdrawal kanban (WK) transfer times are also zero (the succeeding workcenter is located very close)
- Container size is one for all the parts
- The kanbans are released as soon as their containers are empty
- Customer orders (demand) are external
- Demands for Part7 40% of the time and for Part8 60% of the time
- Machine capacities are fixed and set at a level so that they will not generate any bottlenecks within the investigated range of the input variables

As shown in Table 1, the normal distribution is used to represent all time delays in the system to create a relatively stable environment, which is the one of the main assumptions of a kanban implementation.

The objective function of this study is to minimize the possibility of backorders among workcenters and to keep the customer order cycle time at a reasonably low level. Order cycle time is defined as the difference between the time of the completion of the order and the time of the arrival of the order. Using order cycle time as a measure of performance summarizes the effect of all the internal factors in a kanban system. The order cycle times for each end product constitute the response variable values, which are obtained by a simulation model for Part7 and Part8 separately.

3 THE PROBLEM FORMULATION

A simulation model for the defined example was coded in PASCAL to obtain the average cycle time of Part7 and Part8 for each kanban combination. The problem is given in equation 2.

$$
\begin{aligned}
&\text{Minimize}\\
&f = wc * act * tnop + kc * totkan * makespan\\
&\text{Subject to}\\
&1 \le PK5 \le 6\\
&1 \le PK6 \le 3\\
&1 \le PK7 \le 6\\
&1 \le PK8 \le 6\\
&1 \le WK5 \le 6\\
&1 \le WK6 \le 3\\
&0.4cyc7 + 0.6cyc8 \le maxcyc\\
&totkan = PK5 + PK6 + PK7 + PK8 + WK5 + WK6\\
&PK5, PK6, PK7, PK8, WK5, WK6 \text{ are integers}
\end{aligned}
\tag{2}
$$

where:

f:	total cost under given combination of kanbans,
wc:	waiting cost per minute and per order,
tnop:	total number of order processed,
kc:	kanban cost per minute,
maxcyc:	upper limit on the average order cycle time,
cyc7:	order cycle time of Part7 (obtained from the simulation model),
cyc8:	order cycle time of Part8 (obtained from the simulation model),
act:	average cycle time for both part times = 0.4*cyc7 + 0.6*cyc8
makespan:	completion time of all the orders.

Total cost *f* is a function of the number of kanbans only. Order cycles cyc7 and cyc8 are also functions of the number of kanbans only. Makespan could also be determined as a function of the number of kanbans in the system. The makespan values for different combinations were very similar with only a small standard deviation. Thus, the makespan value is computed as an average value experimentally and considered as a constant for each replication.

All definitions, input values and parameters of the kanban controlled system defined above are considered as the same of the example given in Aytug et al. (1996). Under these conditions, our simulation model was built and validated according to the simulation model results given in Aytug et al. (1996).

Examining kanbans, PK5, PK6, PK7, PK8, WK5 and WK6, for this example means that at least 11,664 kanban combination points may need to be studied if the results of the simulation model have a small standard deviation (meaning that no simulation replication is necessary). If the problem is expanded to a more complex form such as adding one additional workcenter at the end of the line, that is, an additional four kanbans (WK7, WK8, PK9, PK10) in the range of $1 \le WK7 \le 3$, $1 \le WK8 \le 3$, $1 \le PK9 \le 3$, $1 \le PK10 \le 3$, then the problem would require examining over 944784 different kanban combination points.

Due to the combinatorial nature of the number of kanbans optimization problem, the problem can become highly complex as the number of kanbans increase. It is not practical from a computational point of view to search the complete set of all combination points. The approach considered in this study avoids having to evaluate all kanban combinations by employing a TS algorithm. As mentioned previously, the developed TS algorithm employed interacts the simulation model of the JIT system.

4 TABU SEARCH

Tabu Search (TS) is a meta-heuristic that guides a local heuristic search strategy to explore the solution space beyond local optimality. The local procedure is a search that uses an operation called a move to define the neighborhood of any given solution. The neighborhood of the current solution is explored and the best solution is selected as the new current solution. The best solution in the neighborhood is selected, even if it is worse than the current solution. This strategy allows the search to escape from local optima and explore a larger fraction of the search space. *Tabulist* includes recently selected solutions that are forbidden to prevent cycling in the search process. If the move is present in the *tabulist*, it is accepted only if it decreases the objective function value below the minimal level so far achieved according to an *aspiration level.* It was originally proposed by Glover (1989, 1990).

4.1 Implementation of the TS Algorithm

The notations used in the developed algorithm are introduced below:

y_0: initial solution
y: current solution
y': neighbor solution
y'_{best}: best neighbor solution
y_{best}: best solution
$M(y)$: a move that yields solution y
ttim: tabu tenure of increase moves
ttdm: tabu tenure of decrease moves

Solutions of the problem (candidate kanban combinations) are represented by an array with six elements. This six elements include the number of kanban for each kanban type, respectively, PK5, PK6, PK7, PK8, WK5, WK6. Increasing and decreasing feasible moves were used to obtain neighbors of any solution. The possible neighbors of a sample solution [2, 3, 4, 3, 5, 3] are given in Table 2.

Table 2: The Neighbors of Solution [2, 3, 4, 3, 5, 3]

[1, 3, 4, 3, 5, 3]	[2, 3, 3, 3, 5, 3]	[2, 3, 4, 3, 4, 3]
[3, 3, 4, 3, 5, 3]	[2, 3, 5, 3, 5, 3]	[2, 3, 4, 3, 6, 3]
[2, 2, 4, 3, 5, 3]	[2, 3, 4, 2, 5, 3]	[2, 3, 4, 3, 5, 2]
[2, 4, 4, 3, 5, 3]	[2, 3, 4, 4, 5, 3]	[2, 3, 4, 3, 5, 4]

The initial solution in the TS algorithm was randomly selected and starting from the initial solution, all possible neighbors of the current solution are examined at each iteration, because the number of neighbors is not too large. The TS algorithm calls the simulation model to compute the total cost correspond to considered neighbor solution. Then the best neighbor is selected as the new current solution, if the neighbor is not obtained via a tabu move.

The tabu list drives the search to different regions of the search space. After any increasing or decreasing move operation, the activated kanbans are recorded on the tabu list. Two different tabu lists called *tabu_increase_start* and *tabu_decrease_start* were built to record the tabu conditions associated with the moves a selected increase or decrease move at iteration i, *tabu_decrease_start (increase_move)* = i, or *tabu_increase_start (decrease_move)* = i. The tabu tenure of increasing moves (*ttim*) and tabu tenure of decreasing moves (*ttdm*) are the number of iterations that forbid a kanban number to be increased and decreased, respectively. Any tested increase or decrease move is tabu at current iteration j ($j > i$), if *tabu_increase_start(test_increase_move)* + *ttim* $\geq j$ or *tabu_decrease_start(test_decrease_move)* + *ttdm* $\geq j$. This structure is also called short term memory, which is recency based (Glover 1989). An *aspiration criterion* was used to decide when the tabu rule can be overridden. The aspiration criterion used in this study removes the tabu condition when any tested move yields a better solution than best solution obtained so far.

The parameters values of TS algorithm were determined experimentally to be *ttim* = 3 iterations and *ttdm* = 4 iterations. The developed TS algorithm is terminated when a chosen maximum iteration number is reached. The steps of the TS algorithm are follows:

Algorithm:

Step 1. Choose the initial solution y_0. Current solution $y = y_0$ and best solution $y_{best} = y_0$. i = 0 and start with empty tabu lists.

Step 2. *Repeat*

Step 2.1. Generate the neighbors, y', for current solution y and call the simulation model to calculate the cost function $f(y')$.

Step 2.2. Select the best neighbor y'_{best}. If $f(y'_{best}) < f(y_{best})$ then $y_{best} = y'_{best}$ and go to step 2.4.

Step 2.3. If ($M(y'_{best})$ is tabu) and ($f(y'_{best}) > f(y_{best})$) then $f(y'_{best}) = \infty$ and go to step 2.2. else current solution $y = y'_{best}$.

Step 2.4. $i = i + 1$. Keep $M(y'_{best})$ on associated tabu list for associated tabu tenure.

Step 3. *Until* $i \geq i_{max}$.

5 RESULTS AND ANALYSIS

In this study, the performance of TS algorithm is compared with the performance of RS algorithm. RS is the simplest

heuristic search method that just samples solution space randomly. Thus, randomly generated kanban combination is used as input value for the simulation model of kanban - controlled system to obtain objective function value. The objection function value is compaired with the minumum objective function value yet encountered. If it is smaller than the the best one, it is stored as the new best solution. This process is repeated until a predetermined number of solutions have been generated.

The results show that the TS algorithm and the RS algorithm find the same kanban combination [1, 1, 3, 3, 1, 1] as the best result, while searching 29 and 794 solutions, respectively. The comparison among the heuristics is presented in Table 3 on the basis of three values: coefficient of variation of cost that is found by the relevant heuristic in each replication, number of solutions searched by the relevant heuristic until finding the best solution (best kanban combination that gives minimum cost) and fraction searched %. In order to ensure a comparable computational effort devoted to each heuristic, the stopping criterion has been defined as a number of solutions visited. This number has been set to 900 in this study. Figure 2 represents the convergence of the TS and the RS algorithms. This figure and also Table 3 showed that the TS algorithm performed better than the RS with the same number of solution visited, TS converged very quickly visiting only 0.249% of the search space. On the other hand, the nonlinear regression model used by Aytug et al. (1996) to find optimum kanban numbers for this problem fitted on the 64 simulation results ($2^6 = 64$) and they also used 6 additional simulation runs to represent validation of their regression metamodel. TS used in this study, only searched 29 points (means that 29 simulation runs) to find optimum or very close optimum solution.

Table 3: Results of the Computational Experiments

	TS	RS
Coefficient of variation	0	11.8
Solution searched	29	798
Fraction searched %	0.249	6.842

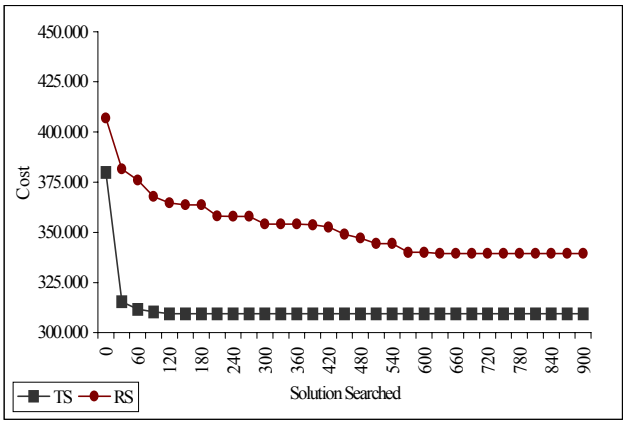

Figure 2: Convergence of the TS and RS Algorithms

6 CONCLUSIONS

In this study, a simulation searched heuristic procedure based on TS was developed and compared with a RS algorithm applied on the same problem considering both solution quality and computational efficiency for determining the optimum number of kanbans to meet production demands in a JIT system. Results indicate that TS algorithm outperforms the RS algortihm searching only 0.249% of the solution space of this problem. The result encourages us to use TS method for simulation optimization.

REFERENCES

Askin, R.G., M.G. Mitwasi, and J.B. Goldberg. 1993. Determining the number of kanbans in multi-item just-in-time system. *IIE Transactions*, 25(1): 89-98.

Aytug,H., C.A. Dogan, and G. Bezmez. 1996. Determining the number of kanbans: a simulation metamodeling approach. *Simulation*, 67(1): 23-32.

Barton, R.R. 1992. Metamodels for simulation input-output relations. in *Proceedings of the 1992 Winter Simulation Conference*, ed., J.J. Swain, D. Goldsman, R.C. Crain, and J.R. Wilson, 289-299.

Bitran, G.R., and L. Chang. 1987. A mathematical programming approach to a deterministic kanban system. *Management Science*, 33(4): 427-441.

Blanning, R.W. 1975. The construction and implementation of metamodels. *Simulation*, 177-184.

Bulgak, A.A., and J.L. Sanders. 1988. Integrated a modified simulated annealing algorithm with the simulation of a manufacturing system to optimize buffer sizes in automatic assembly systems. In *Proceedings of 1988 Winter Simulation Conference*, ed., M. Abrams, P. Haigh, and J. Comfort, 684-690

Deleersnyder, J.L., T.J. Hodgson, H. Muller, and P.J. O'Grady. 1989. Kanban controlled pull systems: an analytical approach. *Management Science*, 35(9): 1079-1091.

Dengiz, B., and K.S. Akbay. 2000. Computer simulation of a PCB production line: metamodeling approach. *International Journal of Production Economics*, 63: 195-205.

Fukukawa, T., and S-C. Hong. 1993. The determination of the optimum number of kanbans in a just-in-time production system. *Computers and Industrial Engineering*, 24(4): 551-559.

Glover, F. 1989. Tabu search Part I. *ORSA Journal on Computing*, : 190-206.

Glover, F. 1990. Tabu search Part II. *ORSAJournal on Computing*, 2: 4-32.

Grefensette, J.J. 1991. Strategy acquisition with genetic Algorithms. In *Handbook of Genetic Algorithms*, ed., L. Davis, 186-201.

Haddock, J., and J. Mittenthal. 1992. Simulation optimization using simulated annealing. *Computers and Industrial Engineering*, 22(4): 387-395.

Hall, J.D., and R.O. Bowden. 1997. Simulation optimization by direct search: a comparative study. In *Institute of Industrial Engineers 6th Industrial Engineering Research Conference Proceedings*, 298-303.

Hurrion, R.D. 1997. An example of simulation optimization using a neural network metamodel: finding the optimum number of kanbans in a manufacturing system. *Journal of Operational Research Society*, 48: 1105-1112.

Kimura, O., and H. Tereda. 1981. Design and analysis of pull system, a method of multi-stage production control. *International Journal of Production Research*, 19(3): 241-253.

Lin, L., and J.K. Cochran. 1990. Estimating simulation metamodel parameters for unexpected shop floor real time events. *Computers and Industrial Engineering*, 19(1-4): 662-666.

Lin, L., and F. Chiu. 1993. Manufacturing cell operating Characteristics. *European Journal of Operational Research*, 69: 424-437.

Lutz, C.M. 1995. Determination of buffer location and size in production lines using tabu search. *European Journal of Operational Research*, 106: 301-316.

Madu, C.N. 1990. Simulation in manufacturing: a regression metamodel approach. *Computers and Industrial Engineering*, 18(3): 381-389.

Madu, C.N., and C. Kuei. 1992. Group screening and Taguchi design in the optimization of multi-echolen maintenance float simulation metamodels. *Computers and Operation Research*, 19(3): 95-105.

Monden, Y. 1981. *Toyota production system.* Norcross, GA: IE&MPress

Muckstadt, J.A., and S.R. Tayur. 1995. A comparison of alternative kanban control mechanism: I. Background and structural results. *IIE Transactions*, 27(2): 140-150.

Ozdemirel, N.E., G.Y. Yurttas, and G. Koksal. 1996. Computer aided planing and design of manufacturing simulation experiments. *Simulation*, 67(3): 171-191.

Rees, l.P., P.R. Philipoom, B.W. Taylor, and P.Y. Huang. 1987. Dynamically adjusting the number of kanbans in a just-in-time production system using estimated values of lead-time. *IIE Transactions*, 19(2): 199-207.

Stuckman, B.E. 1988. Design of general systems by methods of global search: a computer-aided engineering approach. In *Proceedings of the 1988 AMSE International Istanbul Conference on Modeling and Simulation*, 1B: 81-90.

Stuckman, B.E. 1990. Design optimization using simulation and stochastic global search: a computer-aided engineering approach. *Advances in Modeling and Simulation*, 18(4): 13-33.

Stuckman, B.E. 1991. Comparison of global search methods for design optimization using simulation. In *Proceedings of the 1991 Winter Simulation Conference*, eds., B. Nelson, W.d., Kelton, and G.M. Clark, 937-943.

Tekin, E., and I. Sabuncuoglu. 1998. Simulation optimization: a comprehensive review on theory and applications. Technical Report, Bilkent University, Ankara, Turkey.

Wang, H., and H. Wang. 1991. Optimum number of kanbans between two adjacent workstations in a JIT system. *International Journal of Production Economics,* 22(3): 179-188.

Yu, B., and K. Popplewell. 1994. Metamodels in manufacturing: a review. *Industrial Journal of Production Research*, 32(4): 787-796.

AUTHOR BIOGRAPHIES

DENGİZ BERNA is a Professor of Industrial Engineering and Vice Dean of The Faculty of Engineering at the Gazi University. Her field of study is the modeling and optimization of complex systems by simulation and/or metaheuristics techniques such as genetic algorithms, tabu search, and simulated annealing. Her research has been funded The Scientific and Technical Research Council of Turkey (TUBİTAK) , National Planning Center of Turkey (DPT) and National Science Foundation (NSF). Dr. Dengiz is a member of SCS-Society for Computer Simulations, IEEE –Reliability Chapter, Operations Research Society of Turkey, Informatics Society of Turkey, and Statistic Society of Turkey. Her e-mail address is <berna@mikasa.mmf.gazi.edu.tr>.

ALABAS CİGDEM is a Research Assistant of Industrial Engineering. She received as a B.Sc. (1995), M.Sc. (1999) in Industrial Engineering from Gazi University. She is currently enrolled Ph.D. program in Industrial Engineering department at Gazi University. Her research concerns modeling of complex systems using stochastic optimization techniques. She is a member of the Operations Research Society of Turkey. Her e-mail address is <alabas@mikasa.mmf.gazi.edu.tr>.