AN EFFICIENT ALGORITHM TO ALLOCATE PARTS TO CELLS MAXIMIZING CELL EFFICIENCY

R. Tavakkoli-Moghadam, M. Rabbani and A. Godratnama

Department of Industrial Engineering, Faculty of Engineering, University of Tehran
Tehran, Iran, tavakoli@ut.ac.ir - mrabani@ut.ac.ir

N. Safaei

Department of Industrial Engineering, Iran University of Science & Technology
Tehran, Iran, nimasafaei@iust.ac.ir

(Received: March 10, 2003 - Accepted in Revised Form: August 18, 2004)

Abstract In the design of a cellular manufacturing system (CMS), one of the important problems is the cell formation in the form of machine grouping and parts family. This paper investigates an allocation of parts to common and specific cells; in such a way that each common cell is able to process all required parts. Further, this paper presents a mathematical programming model comprising constraints such as available time for common and special cells in each time horizon, and variables such as excess time required by each cell to process parts in each period. The main objective of the model is maximize cell efficiency by minimizing the total tardiness in production of goods and the sum of idle times of machines in each cell as well as by minimizing the maximum tardiness and idle times. To obtain good solutions, a simulated annealing (SA) method has been used. To verify the quality and efficiency of the SA algorithm, a number of test problems with different sizes are solved to show the efficiency of the proposed algorithm. Finally, the results are compared with solutions obtained by Lingo 6 in terms of objective function values and computational time.

Key Words Cellular Manufacturing System, Common and Special Cells, Tardiness, Idle Time, Simulated Annealing

چکیده سیستم تولید سلولی (CMS) یکی از روش‌های نوین تولید می‌باشد، که هدف اصلی آن دسته‌بندی قطعات و ماشین‌ها بر اساس یک معیار مشخص جهت کمینه‌سازی نقل و انتقالات درون کارگاهی مواد می‌باشد. این مقاله، با نگاهی مبتنی بر زمانبندی؛ مساله تشکیل سلول را مورد بررسی قرار می‌دهد. ایده اصلی عبارت است از تقسیم سلول‌ها به دو نوع سلول‌های عمومی و اختصاصی، بنحوی که هر سلول عمومی قادر به پردازش تمامی انواع قطعات بوده و هر سلول اختصاصی تنها برخی از قطعات را می‌تواند پردازش کند. در هر دوره برنامه‌ریزی، هر سلول زمان محدودی را جهت پردازش قطعات در اختیار داشته و در صورت نیاز می‌تواند زمان بیشتری را با رعایت محدودیت کل زمان در دسترس در افق برنامه‌ریزی و همچنین تحمل یک هزینه اضافی استفاده نماید. زمان تحویل هر قطعه مشخص می‌باشد. هدف ایجاد خانواده قطعات بگونه‌ای است که زمان بیکاری و تأخیر در تحویل قطعات کمینه شده و از راندمان سلول‌ها حداکثر استفاده صورت گیرد. مساله فوق با ارایه یک مدل عدد صحیح غیرخطی و حل آن توسط نرم افزار Lingo 6 مورد بررسی قرار می‌گیرد. به دلیل پیچیدگی مدل ریاضی پیشنهادی، حل آن توسط رویکردهای متداول در اندازه‌های بزرگ مستلزم هزینه زمانی بالا بوده و در نتیجه استفاده از روش‌های فراابتکاری اجتناب ناپذیر است. در این مقاله از الگوریتم Simulated Annealing (SA) برای حل مسأله در ابعاد بزرگ استفاده شده است.

1. INTRODUCTION

According to the literature survey, one of the significant criteria applied to machine scheduling is to minimize delay in getting a product to the customer and to maximize machine efficiency. A product is considered to be delayed or behind the schedule, when its production and completion time is greater than its due date or delivery time. Delay time (known as tardy) of jobs or products does not necessarily determine the number of the delayed jobs. This paper specifically deals with the minimization of the tardy jobs as well as maximization of machine efficiency.

By increasing worldwide competition, most companies must be able to improve the quality of

Parts \ Cells	1	2	3	4	5
1	1		1		1
2		1	1		1
3	1		1		1
4		1	1		1
5	1	1		1	1

Figure 1. Feasible cells for processing parts.

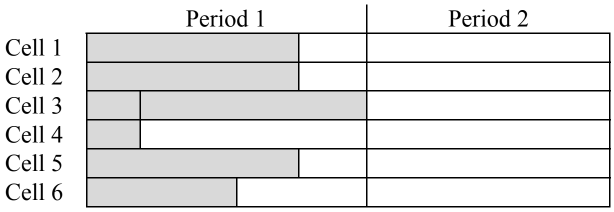

Figure 2. Current loads on special cells.

their products and services significantly. Furthermore, the penalty costs should be reduced and customers should also be kept content by offering products of superior quality and by paying costs for elimination of delay. Thus, minimization of delay times and/or maximization of machine efficiency are vital in an increasing competitive world [1]. In recent decades, the use of cellular manufacturing systems (CMS) has been used significantly. Although, little attention has been paid to group technology concepts including CMS, it has come into focus only in recent decades as a competitive tool for which many success reports have been written.

In this paper, attempt is made to deal with minimization of delay time and maximization of machine efficiency by using the concept of cellular manufacturing system, in which the planning horizon consists of more than one period [2]. Nowadays, companies observing principles of CMS have to pay attention to other aspects of the problem in order to come closer to their goals and ideals. Loading and time scheduling of cells consist of some of these aspects, which are allocation of parts to specific and common cells in order to minimize tardiness and idleness (see Figures 1 and 2).

2. LITERATURE REVIEW

Lawler [3] has indicated that the sum of weighted delays (i.e., $\Sigma\ Wi \times Ti$) have to be minimized belonging to the class of NP-Hard problems. He presented a polynomial algorithm to solve the sum of weighted delays problem. Furthermore, various methods have been presented to investigated for each solution, for both weighted and non-weighted kinds of problems. Emmons [4] has presented various important rules for limiting the search area to find an optimum solution. His rules are used in two algorithms of the branch-and-bound and dynamic programming. Fisher [5], Potts and Van Wassenhove [6,7], and Rinnooy et al. [8] have improved upon Emmons's rules to cover weighted delay problems. Rachomadugu [9] has discovered a situation in which adjacent jobs are placed in an optimum arrangement for the solution of a weighted tardiness problem. Chambers et al. [10] have also presented main rules for flexible decomposition heuristic.

Accurate methods to solve a weighted tardiness problem have been investigated and tested by Abdul-Razaq et al. [11]. They also applied Emmons's main rules to form a preference graph, which is capable of finding upper and lower bounds. They found that the best lower bound for quality solution time is linear. This bound is obtained by Lagrangian elimination of machine capacity limit. Hoogveen and Van Develde [12] have reformulated the problem by the use of extra variables. In this manner, the best lower Lagrangian limit is obtained. Szwarc and Liu [13] have proved the presence of a special arrangement for a problem of the earliest-tardiness (E/T) type in an individual machine with independent penalty of jobs in which the sequence of two adjacent jobs in an optimum succession depends on the starting time. They have also presented a two-stage decomposition method for solution of a sum of the weighted tardiness problem ($\Sigma\ Wi \times Ti$) in which the tardiness penalty is proportional to the processing time.

Akturk and Yildrims [2] have shown the important rule of the relative limit in the weighted tardiness problem in which it is possible to provide sufficient conditions for finding a local solution. Few researches have reported on scheduling problems for an individual machine with two goals of minimizing the number of the delayed jobs and also total delays. Suer et al. [14] have suggested the use of a heuristic method in which the number of the delayed jobs comes as the first goal and total delays as the second one. Lee and Rairaktarakis [15] have suggested a branch-and-bound method to develop an optimum solution for such problems. A similar research by Shanthikumar [16] has taken up the goal of minimizing the maximum tardiness as the second objective. He has also suggested a branch-and-bound method for finding an optimum solution.

There are various cellular loading problems to allocate parts to cells. In contrast to parallel machines scheduling, cells cannot process all available parts. In fact, in many instances, economically, it has neither been feasible nor required to create such flexibility in the design of cellular systems. There is another difference between these two in parallel machines; processing times are used as a base for scheduling problems. Whereas, in cellular loading, the production rate is calculated on the basis of sequences, processing times, and so on before handling cellular loading.

In the literature concerned, very little has been said about cellular loading. Green and Sadowski [17], and Green and Cleary [18] have worked on such subjects as scheduling, benefits, losses, and system's change in cell manufacturing environments. Suer and Saiz [19] have suggested a simple classification design to deal with cellular loading problems. Suer et al. [20] have studied in detail loading rules and connected cells algorithms and also has presented an applied example in a real production environment. A heuristic algorithm based on simulated annealing (SA) has been recognized as a significant stochastic search technique. This algorithm is a calculating process that attempts to solve complicated combinatorial optimization problems, by random ordering and controlling of feasible or infeasible solutions. Krikpatrick et al. [21] has first suggested the solution of the process based on the work by Metropolis et al. [22] in statistical mechanisms. SA method has successfully been applied in such research areas as production control, super-computer design, manpower scheduling, parallel processors, work shop production, installations design and graph theory [23].

3. PROBLEM FORMULATION

A cellular manufacturing system (CMS) model attempts to minimize the delay of product delivery to the costumer and to minimize machine idle time. In contrast to other problems related to cell formation, here it is assumed that the processing capability of the products in cells are known in advance in a zero-one matrix. Thus, by knowing demand for products in different time periods, and by having the given production capability of each cell, the number of cells required to satisfy demands is calculated. In order to have an appropriate scheduling, some cells named common cells have been allotted being capable of processing all types of products. Because of their high cost, a number of these types of cells are limited in case of delivery time delay and a waiting time for costumers.

The use of the cell capacity can be put off to a late time delay, which is only not permissible at the end of the time horizon and the end period. In case, there is not delay in products completion, machines will be in an idle position in cells at any point in time horizon. To introduce this factor in calculations, an attempt has been made of slack variables trying to keep the amount of these variables as low as possible. To keep the amount of delays and idleness of machine in a reasonable range, i.e., to level down the maximum delay and maximum idleness of machines, the objective function has been formulated as shown in Equation 1. In other word, at the end, the goal of the above problem is to load cells as far as possible based on the time available for product processing, without any delay, and within the time limit.

$$\text{Min } \operatorname*{Max}_{i=1}^{n} \{T_i\} \quad \text{and} \quad \text{Min } \operatorname*{Max}_{i=1}^{n} \{S_i\} \tag{1}$$

3.1. Assumptions of the Model

Following assumptions have adopted for the model.

1- Time horizon is known and is comprised of more than one period.
2- Delay in the product completion is permissible and the customer may remain in waiting for some specific time.
3- In case of contingencies and based on the factory's policies, machines and installation in cells may remain idle if the costs accrued are

below a certain accepted level.

4- Time required for the processing of parts in given amount or at a certain production rate is considered as an input into the model.
5- Parts to be produced in each period are also considered as an input to the model.
6- Number of cells, common or specific, is known and is constant at any period.
7- Efficiency of machines and production is 100 % during the production processing.
8- The set up time of machines to produce other products than the present one is zero.

Figure 1 illustrates a typical example problem that five parts have to be processed on five special cells. The element of the matrix is one if a part is processed on the assigned cell. Figure 2 depicts the current loads on five cells within each period.

3.2. Model's Objective and Constraints

Model's objective function minimizes the total delivery delay of the product to customers on one hand and minimizes the maximum of these delays on the other hand. In the second part of this objective function, the time which machines remain idle, either in common and/or specific cells is minimized. In another word the objective is to load and allocate products to cells in as much amounts as possible while not transgressing the available time during the specified period and while the idle times are kept within a reasonable range. Like before, variables of the idle time are leveled out by minimizing the maximum of these times.

Input to the model is presented in a matrix, i.e. feasibility of different products in each cell is expressed by numbers 0 and 1. Number 1 signifies the possibility of the process of a certain product in a cell, while number 0 implies otherwise. In additional matrices, the time dimension of products processing is given in batches based the production capacity of each cell. By knowing demand for products and the production rate, the number of cells required to be allocated for a certain production is obtained. Further, by considering number of periods and the length of the time horizon, the time constraint is worked out. This time constraint is included by slack variables and also by accepting some degrees of freedom to depart from the available time either in a negative or positive way (- and/or +).

3.3 The Mathematical Model

To build the model, the following symbols are used.

3.3.1. Indices

p: different parts ($p = 1,2,\ldots,P$)
h: different periods ($h = 1,2,\ldots,H$)
c: production cells (specific) ($c = 1,2,\ldots,C$)
k: production cells (common) ($k = 1,2,\ldots,K$)

3.3.2. Input Symbols

t_{pc}: production duration of part p based on the production capacity of the specific cell c.
t_{pk}: production duration of part p based on the production capacity of the common cell k.
A_{pc}: equal to 1 if product p is processed in common cell c, otherwise 0.
AT: time available in each period

3.3.3. Decision Variables

T_{ch}: Additional time a specific cell c in period h requires completing its product (delay time).
T_{kh}: Additional time a common cell k in period h requires completing its product (delay time).
S_{ch}: Time slack variable in period h for specific cell c.
S_{kh}: Time slack variable in period h for common cell k.
x_{pch}: Equal to 1, if part p is assigned to specific cell c in period h, otherwise 0.
y_{pkh}: Equal to 1, if part p is assigned to common cell k in period h, otherwise 0.

$$\begin{aligned}\text{Min } Z = & \sum_{c=1}^{C}\sum_{h=1}^{H} T_{ch} + \sum_{k=1}^{K}\sum_{h=1}^{H} T_{kh} + \sum_{c=1}^{C}\sum_{h=1}^{H} S_{ch} \\ & + \sum_{k=1}^{K}\sum_{h=1}^{H} S_{kh} + \sum_{c=1}^{C}\max_{h=1}^{H}\{T_{ch}\} + \sum_{k=1}^{K}\max\{T_{kh}\} \\ & + \sum_{c=1}^{C}\max_{h=1}^{H}\{S_{ch}\} + \sum_{k=1}^{K}\max_{h=1}^{H}\{S_{kh}\} \end{aligned} \quad (2)$$

ST:

$$x_{pch} \le A_{pc} \qquad \forall p,c,h \qquad (3)$$

$$\sum_{p=1}^{P} t_{pc} \times x_{pc1} = AT + T_{c1} - S_{c1} \quad \forall c \qquad (4a)$$

$$\sum_{p=1}^{P} t_{pc} \times x_{pch} = AT + T_{ch} - S_{ch} - T_{c,h-1} + S_{c,h-1}$$
$$\forall c,h \ ; \ 2 \le h < H \qquad (4b)$$

$$\sum_{p=1}^{P} t_{pc} \times x_{pc1} = AT - S_{cH} - T_{c,H-1} + S_{c,H-1} \quad \forall c \tag{4c}$$

$$\sum_{p=1}^{P} t_{pc} \times x_{pc1} = AT + T_{c1} - S_{c1} \quad \forall c \tag{5a}$$

$$\sum_{p=1}^{P} t_{pc} \times x_{pch} = AT + T_{ch} - S_{ch} - T_{c,h-1} + S_{c,h-1}$$

$$\forall c, h \; ; \; 2 \le h < H \tag{5b}$$

$$\sum_{p=1}^{P} t_{pc} \times x_{pc1} = AT - S_{cH} - T_{c,H-1} + S_{c,H-1} \quad \forall c \tag{5c}$$

$$\sum_{c=1}^{C} x_{pch} + \sum_{k=1}^{K} y_{pch} = 1 \quad \forall c, h \tag{6}$$

$$S_{ch} \times T_{ch} = 0 \qquad \forall c, h \; ; \; h < H \tag{7a}$$

$$S_{kh} \times T_{kh} = 0 \qquad \forall c, h \; ; \; h < H \tag{7b}$$

$$T_{ch} \le AT \qquad \forall c, h \; ; \; h < H \tag{8a}$$

$$T_{kh} \le AT \qquad \forall c, h \; ; \; h < H \tag{8b}$$

$$x_{pch}, y_{pch} \in \{0,1\}, S_{ch}, T_{ch}, S_{kh}, T_{kh} \ge 0 \tag{9}$$

The objective Function 2 is to minimize sum of the delays in completion of the product, to minimize idleness of cells in all time periods, and to minimize the maximum of the above-mentioned variables. Equation 3 indicates the feasibility of processing product p in cell c. Equations 4a, 4b and 4c satisfy time available for processing products and also degrees of freedom to deviate from time (slack variables and delay variables) for specific cells in each time periods. Equations 5a, 5b and 5c are similar to previous equations, except that a specific cell is replaced by a common cell. Equation 6 explains the number of cells required for the process of part p in period h. Equations 7a and 7b make sure that at least one of the variables, idleness time, or delay time for all time periods in each cell is zero. Equations 8a and 8b express the upper limit of delay times to deliver products to the customers for all time periods and all cells.

4. THE PROPOSED ALGORITHM

The developed model falls in a class of NP-Hard problems. In another word, to find a final optimum solution at these conditions and constraints becomes difficult or even impossible in action or in a reasonable amount of computational time. Thus, the use of various algorithms is suggested where the structure and performance of which is similar to physical and natural phenomena. For instance, one can refer to neural network (NNs), genetic algorithms (GAs), tabu search (TS), and simulated annealing (SA) methods. In this paper, SA algorithm has been used and designed to solve the proposed model.

4.1. Simulated Annealing Algorithm

Kirkpatrick et al. [21] have introduced the concept of the simulated annealing algorithm. This algorithm is a procedure for solving large combinatorial optimization problems that are similar to the physical annealing process of solids. Solutions in a combinatorial problem are equivalent to states of a physical system, and the cost of a solution is equivalent to the energy of a state. In the searching process, the simulated annealing algorithm accepts not only better but also worse neighboring with a certain probability. This means that the simulated annealing algorithm has the ability to escape from the local minima. Therefore, it can find high quality solutions that do not strongly depend upon the choice of the initial solution compared to local search algorithms. In other words, the algorithm is effective and robust. Furthermore, it has been proven that the computation time of this algorithm has a polynomial upper bound. The simulated annealing procedure includes four basic components [24, 25].

(i) Configurations: all of the possible solutions for the combinatorial problem, i.e. the states;
(ii) Move set: a set of allowable transitions. These transitions must be capable of reaching all of the configurations;
(iii) Cost function: a measure of how good any given configuration is;
(iv) Cooling schedule: the annealing of the problem from a random to a good, frozen solution. Note that the cooling schedule determines the initial temperature, the rule of decreasing the value of temperature, the number of iterations for searching better

configurations at each temperature, and the time at which annealing should be stopped.

Generally, one can use the annealing procedure as shown in steps 1 to 4 to obtain the best solution:

1. Generate an initial configuration *S*.
2. Get an initial temperature $T > 0$.
3. While not yet satisfies the stop criterion.
   3.1. Perform the following loop *L* times:
      3.1.1. Pick a random neighbor *S'* of *S*.
      3.1.2. Let Δ = cost (*S'*)-cost (*S*).
      3.1.3. If $\Delta \leq 0$ then set $S = S'$.
      3.1.4. If $\Delta > 0$ then set $S = S'$ with probability $e^{(-\Delta/T)}$.
   3.2. Set $T = r \times T$, where r is a control parameter small but close to 1.
4. Return *S*.

Step 3.1.1 is known as the generation mechanism of the neighboring configurations. In this paper, generation mechanism focuses on transferring and allocating products to other cells or on substituting these products, generally two products. This procedure continues until the stopping condition is satisfied.

4.2 Cooling Schedule

4.2.1 Initial Temperature In physical analogy, the initial temperature should be large enough to heat up the solid until all particles are randomly arranged in the liquid phase. This means that in the beginning, the temperature of the annealing process must be high enough to make sure that the system can be shifted to all possible states. By this property, the algorithm can find a solution that does not strongly depend upon the initial configuration. Since the probability to accept worse solutions is p_0, the initial temperature T_0 can be determined by means of the cost-increasing transitions, which would be accepted in the beginning of the annealing process with a probability *p*. Pilot runs are performed, and the mean cost increasing $\dddot{\Delta}$ of the cost-increasing transitions is then computed. In the calculation, T_0 is calculated as follows:

$$T_0 \approx \frac{\dddot{\Delta}}{\ln(p_0^{-1})}$$

4.2.2. Number of Iterations The annealing process transfers from one configuration to one of its neighbors with a certain probability, this is equivalent to a Markov chain. Thus, we have to determine the upper bound of the Markov chain length or the number of iterations at each temperature. The upper bound can be a proportion of the size of the neighborhood. According to the generation mechanism, the size of the neighborhood is the order of *p*.

4.2.3. Rule of Decreasing the Temperature For a certain value of temperature, the temperature is reduced when the numbers of transitions reach the upper bound of the Markov chain length. The control parameter, i.e. the reduction ratio of temperature, usually is chosen for small temperature changes. The Markov chain more easily leads to an equilibrium state if the temperature change is small. Hence, we use the decrement rule as follows:

$$T_k = r \times T_{k-1} \qquad k = 1, 2, \ldots$$

The control parameter *r* is small but close to 1.

4.2.4. Stopping Condition The annealing process is terminated when the system is frozen, i.e. the value of the cost function of the solution does not improve after a certain number of consecutive Markov chains. In this paper, the annealing process is terminated if the current best configuration remains unchanged for $\ln|\Theta|$ number of temperature reduction steps. Aarts and Korst [26] have proven that the upper bound of the total number of temperature reduction steps (i.e. the number of Markov chains) is proportional to $\ln|\Theta|$; Θ is the solution space that denotes the finite set of all possible solutions. In this paper, Θ is equivalent to the factorial of *p*. However, most of the elements in Θ are infeasible solution because there are too many zoning constraints, so we use $\ln|\Theta|$ as the upper bound of the number of markov chains.

5. COMPUTATIONAL RESULTS

To obtain the best parameters of the suggested algorithm, a test example was worked out. Then,

TABLE 1. Computational Results Obtained From Optimal Solution and Simulated Annealing.

No.	(P × C × H)	AT	Optimum	SA	Gap (%)
1	5 × 2 × 2	10	14	14	0
2	5 × 2 × 3	9	18	18	0
3	5 × 2 × 3	10	16	16	0
4	6 × 3 × 2	12	42	45	7.1
5	7 × 3 × 2	10	12	12	0
6	7 × 3 × 4	11	19	21	10.5
7	8 × 3 × 2	11	16	20	25
8	8 × 3 × 4	10	36*	44	22.2
9	8 × 3 × 4	14	119	126	5.8
10	9 × 4 × 2	10	37	40	8.1
11	9 × 4 × 3	10	64	64	0
12	9 × 4 × 4	11	148	149	0.6

* The best integer programming (BIP) solution obtained by branch-and-bound method found in Lingo's documents.

through repetition the SA parameters were given as follows:

a) the rate of cooling: 0.05
b) equilibrium control: 0.05
c) freezing control: 0.05
d) maximum number of accepted solutions for each temperature: 100
e) starting temperature: 5
f) minimum number of accepted solutions for each temperature: 50

All computations are programmed in the Visual Basic language to be run in windows 98, Pentium 4 system.

To confirm the solution quality of the simulated annealing algorithm, twelve problems with different size are tested and evaluated. The results are compared with solution obtained by using Lingo. Table 1 shows the comparison between optimum and SA solution and computation time. Results reveal that by increasing the number of period, SA requires longer CPU time, but the solution quality increases. It is worth to mention that solution obtained by SA algorithm is close to optimum solution. The mean different between optimum and SA solution is 6.6 percent obtained from Table 1. In this table, while the problem size increases, the solution time to run SA algorithm becomes smaller. Moreover, the CPU time for optimum solutions is severely sensitive to available time (*AT*) value. By decreasing *AT*, the CPU time for optimum solutions increases progressively. CPU time takes more than 30 minutes to find the optimum solution, whereas it takes about 9 minutes to find the best solution.

Figure 3 shows the SA convergence process for a high dimension problem of 50 parts, 10 special cells, 2 common cells, 3 periods and *AT* = 100. Most of decrease accomplished in the first 263 seconds. The best solution obtained by SA is 6954. However, the best solution obtained by branch and bound is 6912.

6. CONCLUSION

In this paper, a mathematical model is developed and proposed to work out a cellular manufacturing

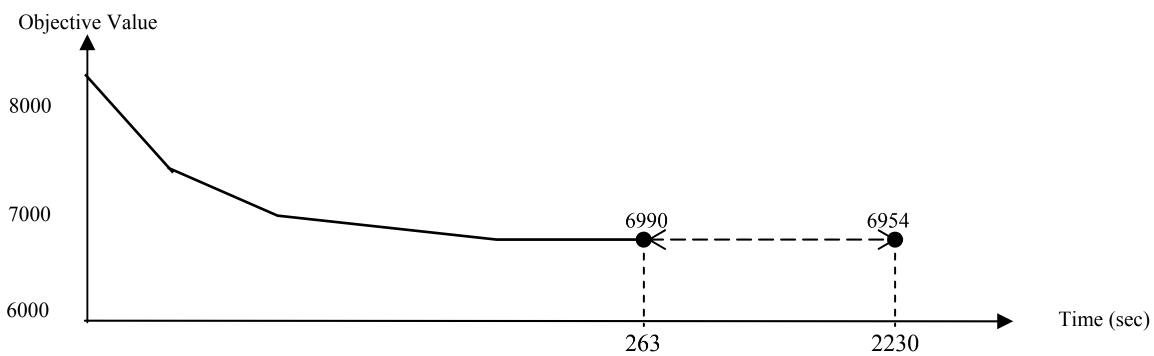

Figure 3. SA convergence process for 50 × 10 × 3 with two common cells and *AT*=100.

system (CMS) problem. The objective function is to minimize delivery time to the customers and also to minimize the idle time of machine in common and special cells. To make the model workable for real time problems, a few constraints are included in the model. Since the model is a complicated one, conventional and traditional optimization methods cannot be utilized in reasonable time. Hence, a meta-heuristic efficient algorithm known as simulated annealing (SA) is used and designed to solve the mathematical model. At the end, some problems of different sizes of number of parts, number of common and specific cells, number of periods and available time in periods are tested and solved. This is done to show the ability and efficiency of the proposed algorithm.

6. REFERENCES

1. Gursel, A. S., "Minimizing the Number of Tardy Jobs in Multi-Period Cell Loading Problems", ***Computers and Industrial Engineering***, Vol. 33, No. 3-4, (1997), 721-724.
2. Akturk, S. M. and Yildrims, B. M., "A New Lower Bounding Scheme for the Total Weighted Tardiness Problem", ***Computers and Operation Research***, Vol. 25, No. 4, (1998), 265-278.
3. Lawler, E. L. A, "Pseudo Polynomial Algorithms for Sequencing Jobs to Minimize Total Tardiness", ***Annals of Discrete Mathematics***, Vol. 1, (1997), 331-342.
4. Emmons, H., "One Machine Sequencing to Minimize Certain Functions of Job Tardiness", ***Operations Research***, Vol. 17, (1969), 701-715.
5. Fisher, M. L. A., "Algorithm for One Machine Scheduling Problem", ***Mathematical Programming***, Vol. 11, (1976), 229-251.
6. Potts, C. N. and Van Wassenhove, L. N. A., "Branch and Bound Algorithms for Total Weighted Tardiness Problem", ***Operations Research***, Vol. 33, (1985), 363-377.
7. Potts, C. N. and Van Wassenhove, L. N. A., "Dynamic Programming and Decomposition Approaches for the Single Machine Total Tardiness Problem", ***European Journal of Operational Research***, Vol. 32, (1987), 405-414.
8. Rinnooy Kan, A. H. G.; Lageweg, B. J. and Lenstra, J. K., "Minimizing Total Costs in One-Machine Scheduling", ***Operations Research***, Vol. 23, (1975), 908-927.
9. Rachomadugu, R. M. V. A., "Note: One Weighted Tardiness Problem", ***Operations Research***, Vol. 35, (1987), 450-452.
10. Chambers, R. J., Carraway, R. L., Cowe, T. J. and Morin, T. L., "Dominance and Decomposition Heuristics f or Single Machine Scheduling", ***Operations Research***, Vol. 39, (1991), 639-647.
11. Abdul-Razaq, T. S., Potts, C. N. and Van Wassenhove, L. N. A., "A Survey of Algorithms for Single-Machine Total Weighted Tardiness Scheduling Problem", ***Discrete Applied Mathematics***, Vol. 26, (1990), 235-253.
12. Hoogveen, J. A. and Van Develde, S. L., "Stronger Lagrangian Bounds by Use of Slack Variables Application to Machine Scheduling Problems", ***Mathematical Programming***, Vol. 70, (1995), 173-190.
13. Szwarc, W. and Liu, J. J., "Weighted Tardiness Single Machine Scheduling With Proportional Weights", ***Management Science***, Vol. 39, (1993), 626-632.
14. Süer, G. A., Czajkiewicz, Z. and Bacz, E., "Minimizing the Number of Tardy Jobs in Identical Machine", ***Proc. of the 15th Conf. on Computers & Industrial Engineering***, Cocoa Beach, Florida, (March 8-10, 1993).
15. Lee, C. and Rairaktarakis, G., "Complexity of Single Machine Dual Criteria Scheduling with Minimum Number of Tardy Jobs", Working Paper, Industrial and System Engineering Department, University of Florida, (1992).
16. Shanthikumar, J. G., "Scheduling n Jobs on One Machine to Minimize the Maximum Tardiness with Minimum Number of Tardy", ***Computers and Operations Research***, Vol. 10, No. 3, (1983), 255-266.
17. Green, T. J. and Sadowski, R. P., "Cellular Manufacturing Control", ***Journal of Manufacturing Control***, (1985), 181-190.
18. Green, T. J. and Cleary, C. M., "Is Cellular Manufacturing Right for You?", ***Annual International Industrial Engineering Conf. Proc.***, (1985), 181-190.
19. Suer, G. A. and Saiz, M., "Cell Loading in Cellular Manufacturing Systems", ***Proc. of the 15th Conf. on Computers and Industrial Engineering***, Cocoa Beach, Florida, (March 8-10, 1993).
20. Suer, G. A., Saiz, M., Dagli, C. and Gonzalez, W., "Manufacturing Cell Loading Rules and Algorithms for Connected Cell", In Planning, Design and Analysis of Cellular Manufacturing System, Edited by A. Kamrani; H. Parsaei and D. Lies, Elsvier, (1995).
21. Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., "Optimization by Simulated Annealing", ***Science***, Vol. 220, (1983), 671-680.
22. Metropolis, N., Rosenbluth, A., Rosenbluth, M.,Teller, A. and Teller, E., "Equations of State Calculations by Fast Computing Machines", ***Journal of Chemical Physics***, Vol. 21, (1953), 1087-1092.
23. Chen, C. L., Cotruvo, N. A. and Baek, W., "A Simulated Annealing Solution to the Cell Formation Problem", ***Int. Journal of Production Research***, Vol. 33, No. 9, (1995), 2601-2614.
24. Wang, T. Y., Wu, K. B. and Liu, Y. W., "A Simulated Annealing Algorithm for Facility Layout Problems Under Variable Demand In Cellular Manufacturing Systems", ***Computers in Industry***, Vol. 46, (2001), 181-188.
25. Rutenbar, R. A., "Simulated Annealing Algorithms: An Overview", ***IEEE Circuits and Devices Magazine***, (1989), 19-26.
26. Aarts, E. and Korst, J., "Simulated Annealing and Boltzman Machines", John Wiley and Sons, N.Y., (1989).