A SIMULATED ANNEALING APPLICATION ON FLOWSHOP SEQUENCING PROBLEM: A COMPARATIVE CASE STUDY

Güzin ÖZDAĞOĞLU(*)

Özet: 90'lardan bu yana geniş bir uygulama alanı bulan modern sezgisel teknikler, bir problem çözümünde, kendi yerel arama sistemleri ile en iyiye en yakın sonuca ulaşmayı amaçlamaktadırlar. Bu çözümler her zaman için tam optimum sonucu bulmayı garanti edemezler, ancak bir çok olay ya da problem için yeterli ve kaydadeğer olurlu çözüm bulabilirler. Bu tekniklerden bir de tavlama benzetimi algoritmasıdır ve çalışmanın uygulama yöntemi olarak seçilmiştir. Bu çalışmada seri çalışma prensibine sahip bir üretim akış hattı alt süreci için sıralama probleminin çözümünde tavlama benzetimi algoritmasının nasıl kullanıldığı sunulmakta ve örnek olay üzerinde uygulama anlatılmaktadır. Tavlama benzetimi algoritmasından elde edilen sonuçlar tanınmış bir çizelgeleme programı olan LEKIN'in sonuçlarıyla karşılaştırılmaktadır. Sonuç olarak, sunulan SA algoritması ile gözetilen amaç doğrultusunda, çizelgeleme uygulamalarında yaygın kullanıma sahip olan LEKIN programına yakın sonuçlar elde edilmektedir.

Anahtar Kelimeler: tavlama benzetimi, sıralama, akış tipi çizelgeleme, LEKIN.

Abstract: Modern heuristic techniques that have wide application area since 1990s, aim at achieving the near optimal with their own specific local search systems. These solutions do not guarantee the optimal solution but represent sufficient and significant results for the cases. One of the modern heuristic techniques that called "simulated annealing" is covered by this study. The study also includes a sample case of a sequencing problem of flow shop system for which a simulated annealing algorithm is presented. In addition, the results obtained from the simulated annealing algorithm are compared with the results of scheduling software LEKIN for the same problem. Finally, a simulated annealing algorithm is obtained which is very close to the results of LEKIN which is broadly used within the scheduling applications according to the objective under consideration.

Keywords: simulated annealing, sequencing, flow shop scheduling, LEKIN.

I. Introduction

Flow shop scheduling is a common problem to solve with the modern heuristic techniques in the literature. In an n-job, m-machine flow shop, each of n jobs is to be processed on a set of m machines. The order of the machines is fixed. The processing time of each job on each machine is assumed to be known. A machine processes one job at a time and a job is processed on one machine at a time with or without preemption. Flow shop problems are solve to find the job sequences by considering the objectives such as minimization of "maximum make span", total mean tardiness, total tardiness, total weighted tardiness, earliness/lateness, and etc. Several heuristic techniques are used to

(*) Arş.Gör. Dokuz Eylül Üniversitesi İşletme Fakültesi İşletme Bölümü

solve such problems because of their NP-hard nature. In this study, a flow shop sequencing problem is represented and solved by simulated annealing algorithm and compared with the solutions of scheduling software called *LEKIN*.

As its name implies, the Simulated Annealing (SA) exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system. The simulated annealing approach is based on a Monte Carlo model used to study the relationship between atomic structure, entropy and temperature during the annealing of a sample of material.

The major purpose of this study is to show the usability of one of the modern heuristic techniques called simulated annealing for flow shop scheduling/sequencing problems. A case study is presented as the problem rose in a Small and Medium Enterprise (SME) Company from textile sector manufacturing various model of shirts. A specific heuristic technique called simulated annealing is introduced and applied for flow shop sequencing problem and compared with another local search algorithm or in other words, heuristic technique.

In literature, several algorithms can be seen such as hill climbing, tabu search, simulated annealing, genetic algorithms, neural networks, etc.. Hill climbing and tabu search algorithms perform a search way which is also used as an initial solution within the other modern heuristic techniques depending on stochastic processes (simulated annealing, genetic algorithms, neural networks, etc. Simulated annealing is one of the simplest techniques developed with stochastic way among the others, and preferred within this decision support system.

The study indicates a decision support system for SMEs who need a systematic system for production planning and scheduling. There is no systematic process or even a production planner defined in this company, thus, this decision support provides a defined process.

Firstly, the theoretical framework is presented including flow shop scheduling/sequencing problem and the simulated annealing algorithm, then the details of the simulated annealing algorithm is introduced using the data gathered from the case study. Finally, the results of the simulated annealing algorithm are then compared with the results of the local search algorithm and other dispatching rules within LEKIN software.

II. Theoretical Framework

A. Flow shop Scheduling/Sequencing

Flow shop scheduling is a common problem to solve with the modern heuristic techniques in the literature. In an n-job, m-machine flow shop, each job is to be processed on a set of m machines. The order of the machines is fixed. The processing time of each job on each machine is assumed to be known. A machine processes one job at a time and a job is processed on one machine at a time with or without preemption. General notation is presented n/m/flow shop (F)/objective and additional constraints in the problem denoting the n jobs and m machines work in the system. Simulated annealing and other similar evolutionary algorithms or modern heuristic techniques are widely used in this area of the problems with NP-hardness as either individually or combination of more than one techniques. NP-hardness arises when problems exhibit exponentially with the decision variables and constraints, and when mathematical techniques such as linear programming is used, the problem becomes impossible to be solved after one point. The need for solution triggers the researchers in developing new techniques called heuristics.

These kind of models published since 1990s indicate that meta /modern heuristic (also called combinatorial optimization) techniques such as genetic algorithm(Holland, 1975; Goldberg, 1989), tabu search (Glover, 1986), simulated annealing (Kirkpatrick et al.,1983), Lagrange relaxation (Held & Karp, 1970), neural networks (Hopfield,1982), ant colony algorithm (Dorigo & Gambardella, 1997), cellular automata (Wolfram, 1984) are the most successful solution techniques for many decision problems, whereas they have risk to drop a local optimum point instead of global one. This risk can be avoided by developing a clear model and validating the results. In each of the modern heuristic techniques for combinatorial optimization problem, there exist a mechanism that rescues the algorithm fall into the local optimum points and stay in there.

In the recent two decades, solution algorithms for many decision problems including the modern heuristic techniques arise exponentially in the literature. Ant Colony Optimization approach is proposed to solve two machine flow shop problem using the features of simulated annealing (T'kindt et al., 2002). A hybrid simulated annealing and a hybrid genetic heuristics are proposed which can be used for the single criterion of makes span or maximum lateness, or the bicriteria flow shop problem (Allahverdi & Aldowaisan, 2004). In this research, a number of new evolutionary algorithms are proposed and a number of modifications are made to several constructive algorithms to cope with non-unique jobs or jobs with multiple demands of flow shop scheduling problem (Burdett & Kozan, 2000). Another study about the problem of permutation flow shop scheduling with the objectives of minimizing the make span and total flow time of jobs is solved by a Multi-Objective Simulated-annealing Algorithm (Varadharajan & Rajendran, 2005). One of the papers addresses the m-machine no-wait flow shop scheduling problem to minimize make span by using hybrid algorithm including Simulated Annealing and Genetic Algorithm (Aldowaisan & Ali Allahverdi, 2003). Flow shop scheduling is a popular field of study and many more studies can be found solving the NP-hard problems with heuristic techniques.

In this study, simulated annealing has been selected to generate a solution for sequencing, after that dispatching rules are also tested using popular scheduling software called LEKIN, and results have been compared in the last sections.

B.Simulated Annealing

The physical process of annealing aims at reducing the temperature of a material to its minimum energy state, called *thermal equilibrium*. The annealing process begins with a material in a melted state and then gradually lowers its temperature. At each temperature the solid is allowed to reach thermal equilibrium. The temperature must not be lowered too rapidly, particularly in the early stages, otherwise certain defects can be frozen in the material and the minimum energy state will not be reached. The lowering of the temperature is analogous to decreasing the objective value (for a minimization problem) by a series of improving moves. To allow a temperature to move slowly through a particular region corresponds to permitting non-improving moves to be selected with a certain probability, a probability that diminishes as the objective value decreases (Singh&Rajamani, 1996).

The algorithm is based upon that of (Metropolis et al.,1958), which was originally proposed as a means of finding the equilibrium configuration of a collection of atoms at a given temperature. The connection between this algorithm and mathematical minimization was first noted by (Pincus,1970), but it was (Kirkpatrick et al.,1983) who proposed that it form the basis of an optimization technique for combinatorial (and other) problems. Following sections contain detailed information about general procedure of the algorithm, characteristics, and area of applications.

The current literature indicates that the use of simulated annealing algorithms broadens the solution space and results in a higher probability, though not guaranteed, of determination a global optimum rather than a local optimum. The traditional search methods rely on an iterative descent approach that performs well if the objective function has a convex continuous shaped function. In practice Simulated Annealing algorithms yield a polynomial time solution to an exponential time problem.

Simulated annealing has an ability to avoid becoming trapped at local minima. The algorithm employs a random search, which not only accepts changes that decrease objective function, but also some changes that increase it. The latter are accepted with a probability. The implementation of the SA algorithm depends on this annealing process structure and the process requires the following elements:

- a representation of possible solutions,
- a generator of random changes in solutions,
- a means of evaluating the problem functions, and

- an *annealing schedule* - an initial temperature and rules for lowering it as the search progresses.

Another significant component of an SA code is the random number generator, which is used both for generating random changes in the control variables and for the temperature dependent increase acceptance test. SA algorithm always accepts a better solution based on the objective but it also reduces the likelihood of the solution being trapped; by accepting a worse solution if an acceptance criterion value is greater than a selected random number.

General steps in a basic simulated annealing method can be described in clearly (Brandimarte, 1995) and also all artworks in this area. For this study, steps of the simulated annealing algorithm are presented during the case study application. There exist many developments about simulated annealing algorithm and its components, those developments are examined in details in the application section to avoid repetition of formula and steps.

Simulated annealing algorithm uses cost functions defined by users according to the problem under consideration. This function may really be a cost function for a production or financial problem, or a penalty function, such as tardiness, setup, etc. The objective function should be clearly defined, and neighborhood structure should also be considered for this determination. If constraints and objective function is modeled successfully, this may reduce the neighborhood size, and then the performance of the algorithm would be higher (http://csep1.phy.ornl.gov/CSEP, 25.12.2003).

The annealing schedule determines the degree of uphill movement permitted during the search and is, thus, this part is critical to the performance of the algorithm. The principle underlying the choice of a suitable annealing schedule is easily stated; the initial temperature should be high enough to *melt* the system completely and should be reduced towards its *freezing point* as the search progresses. In fact, choosing an annealing schedule for practical purposes is still something not solved completely (Bounds,1987). The standard implementation of the SA algorithm is one in which homogeneous Markov chains of finite length are generated at decreasing temperatures. The following parameters should therefore be specified for annealing schedule:

1. Initial temperature (T_0);
2. Final temperature (T_f) as a terminating criterion or threshold value;
3. Length for the Markov chains; and
4. Rule for decrementing the temperature.

The traditional simulated annealing algorithm works with respect to these critical parameters above, but different parameters can be added if necessary for the decision problem.The explanations below presents the properties of the traditional parameters of simulated annealing.

Acceptance Probability: There exists a general standard simulated annealing formula for calculation of *acceptance probability*, which is used to measure the potential of the new seed having lower objective function value than the old one. This formula can be differentiated by changing or simplifying parameters or variables also has developed some modifications. In the model generated for this textile problem, the following acceptance probability given in formula 1 is applied in the simulated annealing algorithm that is the most commonly used (Reeves, 1993):

$$P_a\{accept \quad j\} = \begin{cases} 1 & if \quad f(j) \le f(i) \\ e^{-[f(j)-f(i)]/T} & if \quad f(j) > f(i) \end{cases} \tag{1}$$

where

$P_a\{accept\ j\}$: the probability of accepting j^{th} neighbor instead of i^{th} which is the old one during a run of an iteration.

$f(i)$: current fitness (objective) function value (old value).

$f(j)$: fitness (objective) function value of the new move.

T: temperature. This state of the problem can be explained as the objective function which is defined as "total weighted tardiness". The objective function to be minimized starts with an initial value, and the formula 2 is used in general simulated annealing algorithms to determine this value.

Temperature is the main controlling parameter of simulated annealing as mentioned in first four sections. An initial temperature should be determined to start the algorithm, thus this value is not chosen randomly. For the initial temperature the formula 2 has been applied for this model because of its

$$T_{initial} = \frac{f_{\min} - f_{\max}}{\ln(P_{in})} \tag{2}$$

where

$T_{initial}$: initial temperature of the cooling system

f_{min}, f_{max}: minimum and maximum values of fitness functions

P_{in}: a high acceptance probability that is closer to one

Formula 2 says that the desired difference between minimum and maximum value of the objective function, which is the solution range, over natural logarithm of a sufficiently high probability can be used as the initial temperature value. This formula is generated from acceptance probability formula by subtracting the temperature.

Length of markov chain is the step size in which the thermal equilibrium can be reached. As initial temperature approaches to zero, acceptance probability of any moves decreases, and to prevent from this situation, the step size or in other words, number of steps is bounded by a constant value. This constant step size value is called "length of markov chain" in conceptual framework of simulated annealing. The length of markov chain is selected experimentally, or by judgmental evaluation according to the problem under consideration.

Rate of cooling is the most important parameter of the simulated annealing algorithm that is defined as the value by which the temperature of a particular state decreased. The cooling rate should be selected carefully to avoid fast cooling which causes to drop local optimum and stay there. There is a successful formula 3 generated by (Bennage & Dhingra,1995) by using final and acceptance probability definitions, and it is used as the rate of cooling in the algorithm as represented in formula 3 below:

$$\alpha = \left(\frac{\ln P_{in}}{\ln P_f} \right)^{M_l} \qquad (3)$$

This formulation was also used for a lay out problem in (Baykasoglu & Gyndy, 2001)

By using the formula above at each step the temperature is lowered by α multiple of the temperature of that step. Then the final temperature is calculated by formula 4:

$$T_{final} = T_{initial}\alpha^{M_l} \qquad (4)$$

where
α: cooling rate.
P_{in}: high acceptance probability defined at the beginning .
P_f: final acceptance probability .
M_l: length of markov chain.

Values of the parameters should be well adapted for the problem under consideration, and as usual, initial parameters are chosen by trials and according to the experiences to reach the appropriate values to decrease the computational complexity. Detailed literature survey about simulated annealing in all application areas can be found in (Laporte & Osman, 1996).

III. Sample Case Study

A.Problem Definition

The case takes place in small business firm that produces several types of shirts. The manufacturing area includes different departments such as sewing, ironing, button-hole and button, press, control, and packaging. Because of the characteristics of these departments, most of schedules and sequences are made separately, and in this case a sequencing problem has been determined in button-hole and button department.

In the button-hole & button department, there exist four machines; two of them are button machine and the remaining two are button-hole machine. All of them process different operations and serially connected, therefore the scheduling type of the department is flow shop with the definition $n=5/m=4/F\ /\sum w_jT_j\ /\ prmu$ considering the minimization of total weighted tardiness. The Figure 1 shows the structure of this flow shop environment.

If a shirt is considered, there are several button and button-holes on different parts (body, arm, collar, pocket), and these operations are divided into the machines, first button-hole machines complete the operations and button operations come after that (Figure 1).

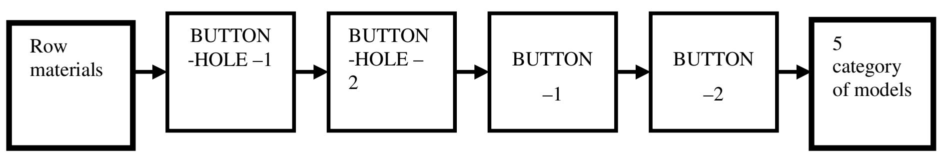


Figure 1: *Machine Flow Of The Button & Button-Hole Department*

According to the model of the shirt, the number of buttons and thus the button-holes differ, this differences affect the process times of the shirts on each machine. During the meeting of the production supervisor, it is claimed that when all shirt models are considered according to the number of buttons on them, there exist 5 main different types of models being produced, and of all orders, shirts have been grouped according to this information. The supervisor wants to obtain a successful sequencing, which minimizes total tardiness of the shirt batch orders as much as possible. But each category of the shirt models has different importance level, therefore the objective can be considered total weighted tardiness. The solution algorithm is developed based on minimizing total weighted tardiness which is commonly preferred in flow shop sequencing problems and is tried to solved using modern heuristic techniques because of the NP-hardness.

B.Assumptions and Constraints

Shirts come to the department by portable textile-cars in-groups, and are transported between the machines and departments by these cars. This produces a constraint in the model that the finished products can not be moved from the machine until the remaining group members complete their processes.

This constraint simplifies the computations; instead of considering processing times per unit, total batch processing times plus setup times can be computed as the total processing time of this model, thus we can act as we have 5 different products and five different processing times when modeling the sequencing algorithm. Due to the data received from production supervisor, sample production period for the button-hole & button department can be summarized as given in Table 1. The cells in Table 1 shows processing times given for each machine representing total operation time of each job in the related machine in terms of hours.

Table 1: *Computed Input Data For Sequencing Problem*

	JOBS				
MACHINES	J1*	J2	J3	J4	J5
M1**(buttonhole1)	8	12	14	15	10
M2(buttonhole2)	2	6	9	16	5
M3(button1)	7	8	8	11	7
M4(button2)	4	6	7	12	5
due date	55	55	75	85	68
job_no	J1	J2	J3	J4	J5
WEIGHTS	3	1	2	5	4

*Job1 (J1)….Jj (jth job)
**Machine1 (M1), …,Mi (ith machine)

As mentioned in the section A in the third chapter, total weighted tardiness has been chosen as the objective function of the model under consideration. According to the information obtained from production environment, a simulated annealing algorithm has been developed, and presented in following sections. Lastly, an assumption has been added to the problem that all products follow same order.

In this case study, the desired solution is to obtain the best sequencing that minimizes the total weighted tardiness. The total weighted tardiness is one of the hardest cases among the other objective functions, and also the flow shop scheduling problems including three or more machines are considered as NP-Hard (Pinedo, 1997). Because of that, heuristic solutions are mostly preferred for these kinds, and simulated annealing algorithm is preferred for this study because of the simplicity.

C. Objective Function

The considered problem is encoded as permutation schedule *permutation encoding*, in which the characteristic numbers in code shows the product number and total code gives the sequence as in the following example: [1 5 4 3 2]. This type of code and its neighborhoods is the search space of the model.

In the most neighborhood search techniques, objective function is also called *"fitness function"*, and denoted as *F(.)* or *f(.)*.

The selected objective function, the total weighted tardiness, can be formulated for this algorithm through formula 5 to 9 (Pinedo, 1997):

$$f(.) = \sum_{j=1}^{5} T_j w_j \tag{5}$$

$$C_{k,j_1} = \sum_{j=1}^{k} p_{k,j_1} \tag{6}$$

$$C_{1,j_k} = \sum_{j=1}^{k} p_{1,j_k} \tag{7}$$

$$C_{i,j_k} = \max\{C_{i-1,j_k}, C_{i,j_{k-1}}\} \tag{8}$$

$$T_j = \max\{(C_{4j_k} - D_j), 0\} \tag{9}$$

Where
T_j: tardiness of job *j*.
w_j: weight given to job *j*.
$C_{k,ji}$:total processing time of k^{th} machine for all products, k= 1,2,…4
$C_{i,jk}$: completion time of job j on i^{th} machine (last machine).
D_j: due date of job *j*. (in hours)
$p_{k,jk}$: total processing time of Job *k* on all machines *k*=1,2,..5,

These variables are all component of scheduling framework, and defined according to the problem type, after these explanations, components of search scheme should be defined.

D.The Search Algorithm (Simulated Annealing)

The components and parameters which are given in the section IIB are applied step by step according to the flow given in the section IIIC with the initial values that are determined at the beginning.

In this study length of markov chain has been chosen as M_l=10 because of the small problem size. Trials show that the selected value is sufficient for this problem, more than this value results in just an inefficient computational effort and larger number of iterations. This limitation of step size in each temperature also fixes the last acceptance probability (0.95), which is called *final acceptance probability,* and the value of the probability is important for cooling. *The initial acceptance probability*, chosen with a 0.96 high value is determined by trials to reach the best solution.

In addition to length of markov chain, each temperature has been iterated for 10 times, and the initial range between minimum and maximum values of objective function is taken as –20 according to the trials, and after the range of 20 the objective function value does not change. It is negative because the objective is to minimize the total weighted tardiness, and should be decreased during the algorithm.

Following steps are integrated according to the general calculation phases of simulated annealing which is used during the development of VBA codes in Excel :

Step1. Determine the initial values.

a) Randomly chose an initial solution.

b) Define solution range (f_{min}-f_{max}), final probability value (P_f), length of markov chain (M_l).

c) Calculate initial temperature ($T_{initial}$), cooling rate(α), final temperature (T_{final}).

d) Set Temperature(T)= $T_{initial}$

e) Set *length=0, f(j)=0.*

Step 2. Calculate the initial objective function value.

a) Calculate the total weighted tardiness of the initial solution.

b) Set this value as *f(i)* and continue.

Step 3. Test the termination and temperature reduction condition.

a) If $length=M_l$ reduce the temperature as $T=T-\alpha T$ and set *length=0* and go to step 3b , else continue with current *length* and same temperature (T) and go to step

b) If $T = T_{final}$ then, stop the algorithm and set the objective function value at this level as the best solution, else go to step 4.

Step 4. Generate the new sequence.

a) Randomly select two integers between 1 and number of products.

b) Exchange the position of these numbers on the current sequence. (e.i, if the first sequence : 2-3-5-1-4, and randomly 5 and 2, then the new candidate sequence is figured out as 5-3-2-1-4)

c) Calculate the objective function value of the new sequence.

d) Set this value as $f(j)$.

Step 5. Compare the objective function values.

a) If $f(j)<f(i)$ then replace $f(j)$ with $f(i)$ and *length*=*length*+1, go to step 3a. Else go to step 5b.

b) If $f(j)\geq f(i)$ but $P_a(j)\geq P_f$, then replace $f(j)$ with $f(i)$ and *length*=*length*+1, go to step 3a. Else go to step 5c.

c) If $f(j)\geq f(i)$ but $P_a(j)<P_f$, then continue with old $f(i)$ and *length*=*length*+1 go to step 3a.

The most important step of the simulated algorithm is the moving policy given in step 5. After the objective function value calculated for the new sequence, this value is compared to the old one. If the objective function value of the new sequence is smaller than the old one, the old sequence is canceled from the system and, new sequence continues to be processed. However, new sequence can be worse than the old one, if such a situation exists, then acceptance probability (P_a in formula 1) is calculated for the new sequence. If the acceptance probability of the new sequence is higher than the threshold value chosen at the beginning (P_f), that means, this sequence is not better than the old one, but has a potential performance to reach better solutions, then algorithm is continued with the new one, otherwise, algorithm continues with the old one to generate new sequences. The acceptance probability limit prevents the algorithm to stay in the local optimum points and rescue the iterations to jump to the new sequences out of the local minimum points.

Using this algorithm simulated annealing has been applied to the sequencing problem through an excel application sheet developed by using visual basic codes. Section IIIE presents these computational results, solutions and comparisons with traditional sequencing rules implemented by using LEKIN scheduling software.

E.Computational Results- Simulated Annealing Solutions

The algorithm has been executed on three macros application on Ms-Excel 11.0 by using 5 -type –product and four machines. The first macro clears the input table for new solution, the second macro generates the initial solution and the last and the most important one implements the search algorithm until termination. Following Figure 2 represents the user interface view of worksheet:

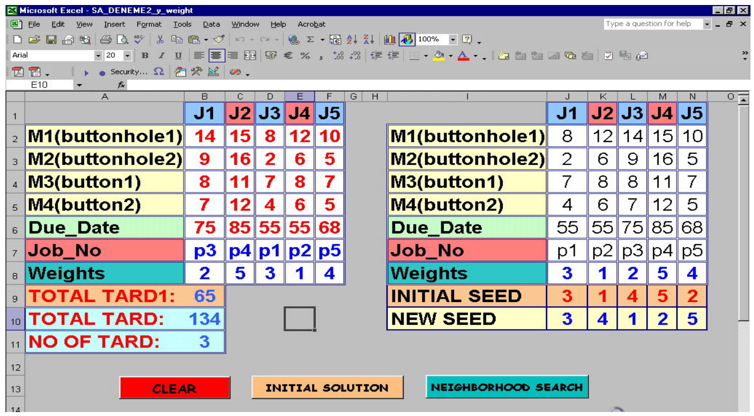


Figure 2: *User-interface of Excel Application Sheet*

Solution at each move has been traced carefully and it can be said that the algorithm has been generated successful results. If the problem size considered, there exist 5! different sequencing, and if fully searched it has rather high computational effort which grows highly while the number of products increases, but here the generated algorithm achieves a successful solution by 57 moves, this changes according to the initial solution. A sample solution list is summarized here with moves started with random solution and function values as given in the Table 2.

Table 2: *No of Moves and Sequences Obtained in The Execution of The Algorithm*

Move no:	SEQUENCE					Tot. Weig. Tardiness	No of tardy jobs	Make span
1	j5	j3	j1	j4	j2	42	2	92
2	j5	j3	j1	j4	j2	42	2	92
3	j1	j3	j5	j4	j2	42	2	92
4	j1	j5	j3	j4	j2	42	2	92
5	j5	j1	j2	j3	j4	65	1	98
11	j5	j1	j3	j4	j2	42	2	92
12	j5	j1	j3	j4	j2	42	2	92
13	j5	j1	j2	j4	j3	32	1	91
31	j1	j5	j3	j4	j2	42	2	92
54	j1	j4	j5	j3	j2	25	1	80
55	j1	j4	j5	j3	j2	25	1	80
56	j1	j4	j5	j3	j2	25	1	80
57	j1	j4	j5	j3	j2	25	1	80

Table 2 includes the partly print table of the Ms-Excel module results and represents all steps. Some sequences can be seen more than once, because it can be the point where the new seed is worse, and where the algorithm searches for a new seed to jump into.

For the sample execution, totally 57 moves are seen when begins with the sequence j5-j3-j1-j4-j2 , and the last solution row which is best f the algorithm is the sequence j1-j4-j5-j3-j2 with 25 weighted-hours totally with one tardy job and 80 hours make span. Several computations have been implemented and the algorithm has given the same solution for different initial sequences. If the objective function values of each move are examined, it is easy to see the waves of changes; these changes are presented in the Figure 3.

Figure 3 also indicates the local minimum points and the performance of the

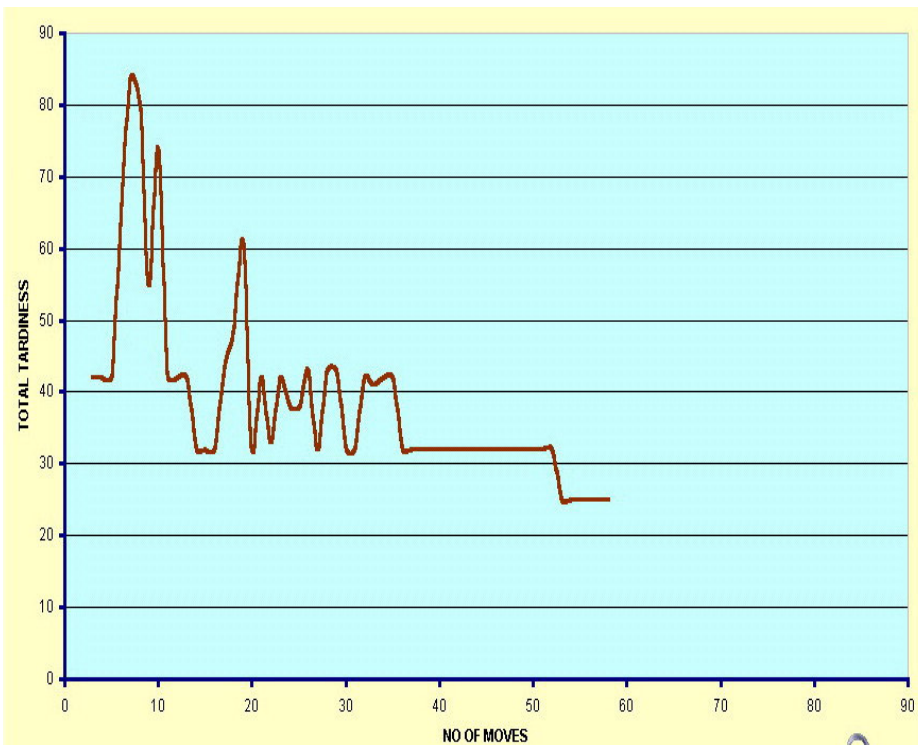


Figure 3: *Changes On Total Weighted Tardiness During The Neigborhood Search*

algorithm to jump to the new values. This algorithm is developed to minimize the objective function, but Figure 3 shows that objective function increasing at the beginning. This situation is the advantage of simulated algorithm, if acceptance probabilities are high although the objective function value is smaller. The solutions/candidate sequences at the beginning forwards the algorithm to continue from the right point.

After the last decrease on graph, the solution remains constant in the threshold value and this algorithm could not find better solution, thus this solution is accepted as the near minimum value of total weighted tardiness of the sequence.

To see the full performance of the algorithm, initial solution is generated randomly rather than any sequencing rule, of course, as expected, if another solution is integrated in to the initial solution, then simulated annealing finds the same solution faster. In this study, the purpose is to represent only the performance of simulated annealing algorithm, clearly, without any support by another solution, thus algorithm is started by a random solution.

If required, pre-sequenced seeds can also be written into the initial solution area. Results show that simulated annealing algorithm generates nearly optimal and mostly the exact optimal solution for this type permutation scheduling problems. This execution may be expanded by increasing the problem size and the properties or constraints of the flow shop problem; these are only the simple modifications on macros, not on the algorithm. Following sections include the comparison of the solutions on LEKIN software with simulated annealing solution developed for this case.

F.*Computational Results - LEKIN Solutions*

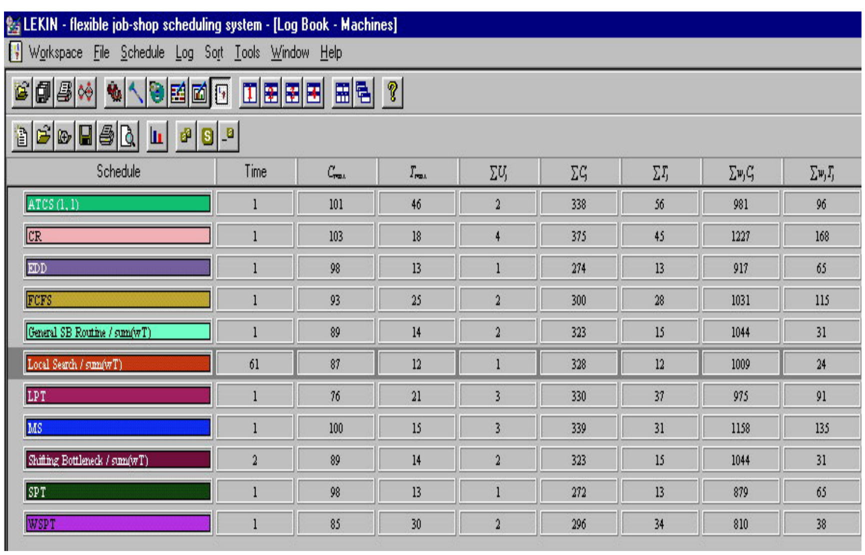
Simulated annealing algorithm produces near optimum solution and does not guarantee that it is exact optimum point. Under these circumstances, it is considered that the performance of the algorithm can be compared with another algorithms, and so that the LEKIN is well known and commonly used scheduling software, the results are compared with the LEKIN's.

The same problem has been solved by using software LEKIN Scheduler with different types of sequencing rules and algorithm. Following table (Table 3) shows the values of each rule and the other heuristics:

Table 3: *Objective Function Values of Rules and Algorithms*

Sequencing Rule	Total weighted tardiness of the flow shop (work hours)
Simulated Annealing	25
Local Search (LEKIN)	24
ATCS	96
CR	168
EDD	65
SHIFTING BOTTLENECK	31
MS	135
GENERAL SB ROUTINE	31
FCFS	115
LPT	91
SPT	65
WSPT	38

All sequencing rules and integrated heuristic algorithms are implemented on this software and results are summarized in the Figure 4 and Figure 5. According to the results in Figure 4, among all rules, local search model of LEKIN is given the best objective function which gives the total weighted tardiness value of 24 with 1 tardy job and 87 hours make span. The local search algorithm of LEKIN produces the following Gantt Chart and sequences given in the Figure 5 and Figure 6, simultaneously.


LEKIN - flexible job-shop scheduling system - [Log Book - Machines]

Workspace File Schedule Log Sort Tools Window Help

Schedule	Time	C_{max}	T_{max}	$\sum U_j$	$\sum C_j$	$\sum T_j$	$\sum w_j C_j$	$\sum w_j T_j$
ATCS (1, 1)	1	101	46	2	338	56	981	96
CR	1	103	18	4	375	45	1227	168
EDD	1	98	13	1	274	13	917	65
FCFS	1	93	25	2	300	28	1031	115
General SB Routine / sum(wT)	1	89	14	2	323	15	1044	31
Local Search / sum(wT)	61	87	12	1	328	12	1009	24
LPT	1	76	21	3	330	37	975	91
MS	1	100	15	3	339	31	1158	135
Shifting Bottleneck / sum(wT)	2	89	14	2	323	15	1044	31
SPT	1	98	13	1	272	13	879	65
WSPT	1	85	30	2	296	34	810	38

Figure 4: *Results Of Computations For Different Rules*

Figure 4 shows the sequences calculated with the different rules algorithms within LEKIN's modules covering the traditional scheduling rules with the different performance measures. The first column (schedule time) shows the total number of iterations to find the given results and the following columns represents the performance measures of the schedule; make span (C_{max}), maximum tardiness (T_{max}), total number of tardy jobs ($\sum U_j$), total completion time ($\sum C_j$), total tardiness ($\sum T_{j...}$), total weighted completion time ($\sum w_j C_{j...}$), and total weighted tardiness ($\sum w_j T_j$) for j jobs where $j=1,..n$,, respectively. The best solution among all scheduling rules is LEKIN's local search algorithm which finds the solution with 87 hour- makespan, 1 tardy job, 24 hour total weighted tardiness values.

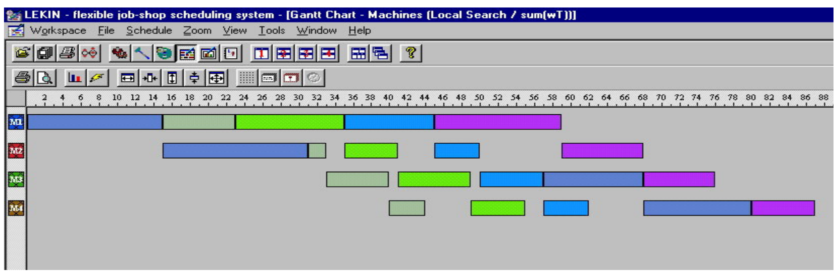


Figure 5: *Gantt Chart of Local Solution of LEKIN*

Figure 5 represents the Gantt chart of the sequence/schedule generated by the local search algorithm of LEKIN according to the machines. In the schedule of first machine, the jobs are ordered with the different color with respect to the sequence given by LEKIN. The colors are also represented in Figure 6 with related jobs. These jobs are started on the first machine and handled to the next one after completed the operation. The first machine has the highest utilization rate, but the other machines have idle times during the operations. There exists a mechanism to control the job sequences within the LEKIN modules, and after this control, the sequence of the jobs in each machine can be changed. Figure 6 shows this situation that the sequences in the third and fourth machine are different from the others. Because of that, it is not possible to compare the performance of simulated annealing algorithm with the LEKIN's local search approach, but still supports that simulated annealing algorithm produces successful results near to the LEKIN which is used in many large enterprises.

However, the proposed model generated only one hour difference form local search algorithm of LEKIN with more successful sequencing and 7 hours less make span. Table 4 presents the final results from simulated annealing and from LEKIN's local search algorithm. LEKIN's local search algorithm produces two different sequences for the first two machines and last two machines, respectively whereas simulated annealing uses the same sequence for all machines.

Table 4: *Comparison of Results*

Simulated Annealing (∑Tw= 25 hours, Cmax=80 hours, no.of moves=57)	LEKIN's Local search algorithm (∑Tw= 24 hours Cmax=87 hours, no.of moves=61)				
J1 J4 J5 J3 J2	J4	J1	J2	J5	J3
	J1	J2	J5	J4	J3

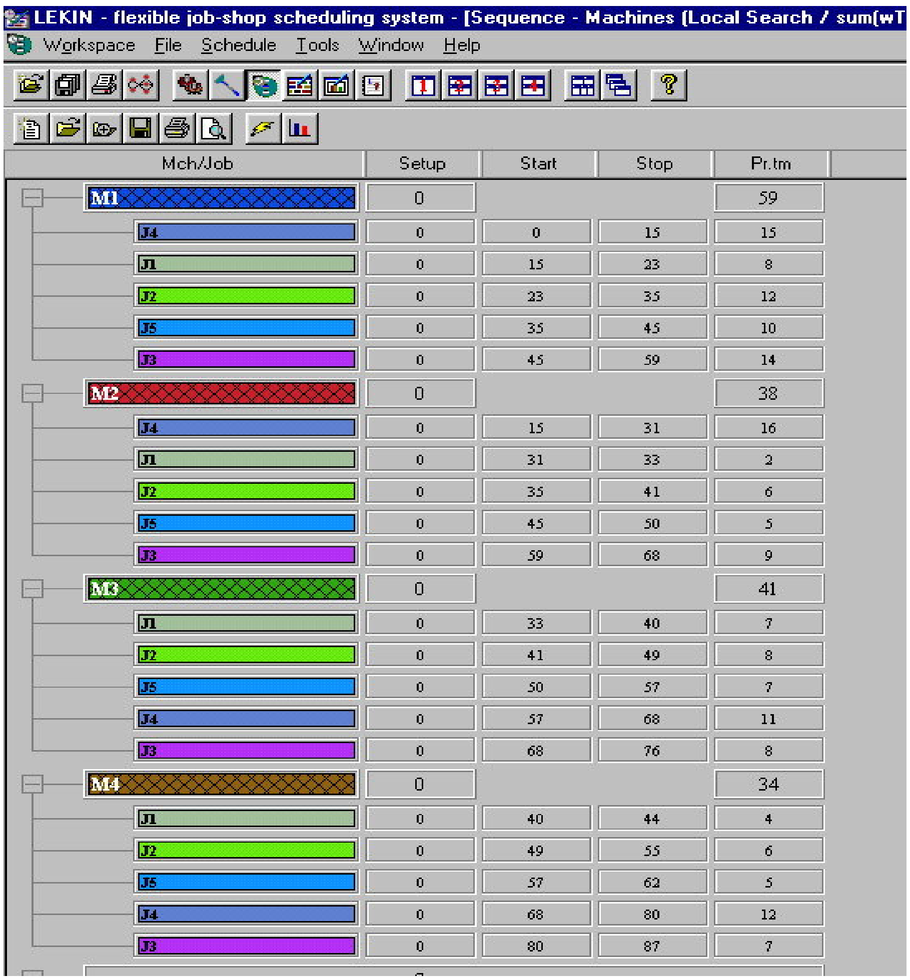


Figure 6: *Seqeunce Scheme of the Local Search Algorithm*

The best solution generated by LEKIN shows that the sequence should be J1-J2-J5-J4-J3, with 24 hours weighted total tardiness and 87 hours make span. In the previous section, simulated annealing algorithm has found this value 25 hours with 80 hours make span. LEKIN seems better, but as mentioned before, it can not be an exact comparison. For total weighted tardiness LEKIN finds only one tardy job and has 12 hours tardiness with weight value of 2. However, we have an assumption that all jobs will follow same sequence, but if the Gantt chart is examined carefully, it can be seen that the algorithm changes the sequence on the last two machines in Figure 5, so it can not be compared with the simulated annealing algorithm generated in the scope of this study.

IV.Conclusion

Simulated annealing algorithm was developed by the natural annealing process that covers the cooling scheme of materials, their thermal characteristics, energy and temperature flows has been translated into optimization language, behavior of the material has been taken into consideration for the variables to be optimized. And recently, there is still increasing trend on the annealing algorithm and estimation of its parameters.

Simulated annealing algorithm is one of the most useful meta/modern heuristic techniques that are used in many fields of study such as function optimization, statistical analysis, finance, production, physical examinations and etc. The main logic of the algorithm is escaping from local optimal points to achieve the global optimum. Therefore, small peaks should be passed, and for this reason some sequences can be accepted even if they are worse than old one.

In this study, scheduling applications has been handled relatively more than the other fields of studies. Finally, a real case study from production environment has been examined including permutation schedule for four-machine and five-job flow shop problem as an application example. For this sequencing problem, the proposed simulated annealing model has been compared with the traditional sequencing rules and some heuristic techniques and among them it can be said that the simulated annealing algorithm is preferably successful. The computation for other heuristics and rules, LEKIN software has been used and, some contradictions have been found in heuristic algorithms. However, the proposed model generated only one hour difference form local search algorithm of LEKIN with more successful sequencing and 7 hours less make span. The main purpose of the case to show the performance of simulated annealing algorithm on flow shop scheduling problem, thus assumptions and constraints, and also objectives can be chosen in any way. These changes do not affect the algorithm. In other words the changes make only the difference in formula of objective and/or sequence computations and the structure of the algorithm is independent of the situation.

The simulated annealing algorithm can be applied for many problems from production and service fields, and the macro developed for the execution of simulated annealing algorithm can be improved for the further studies to develop an integrated decision support system for the industries.

References

Ahmed, A.M, Alkhamis, T.M.,(2002). Simulation Based Optimization using Simulated Annealing with Ranking and Selection. *Computers and Operations Research,* 29, p.387-402.

Aldowaisan, T.; Allahverdi, A. (2003). New heuristics for no-wait flow shops to minimize make span, *Computers & Operations Research*, 30, p. 1219–1231.

Allahverdi, A.; Aldowaisan, T. (2004). No-Wait Flow shops With Bicriteria of Make span And Maximum Lateness, *European Journal of Operational Research*, 152, p. 132–147.

Baykasoglu,A, Gyndy, N, (2001). A Simulated Annealing Algorithm For Dynamic Layout Problem, *Computers & Operations Research*, 28, p.1403-1426.

Bennage WA, Dhingra AK.(1995). Single And Multi-objective Structural Optimization in Discrete-Continuous Variables Using Simulated Annealing. *International Journal for Numerical Methods in Engineering*, 38, p. 2753-73.

Bounds, D. G. (1987). New Optimization Methods from Physics and Biology, Nature, 329, p. 215-218.

Brandimarte, P.(1995). Advance Models for manufacturing Systems Management. Florida:CRC Press

Burdett, R.L.; Kozan, E. (2000). Evolutionary algorithms for flow shop sequencing with non-unique jobs, *International Transactions in Operations Research*, 7, p.401-418.

Dorigo, M.; Gambardella, L.M.(1997) Ant Colonies for the Travelling Saleman Problem, *Biosystems*, 3 (2), p. (73-81).

Glover F. (1986). Future paths for integer programming and links to artificial intelligence. Computers & Operations Research, 1(3), p.533–49.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass.

Held, M.; Karp, R.M. (1970) The Traveling Salesman Problem and Minimum Spanning Trees.

Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor.

Hopfield, J.J.(1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Science, USA, 79, p. 2554.

http://csep1.phy.ornl.gov/CSEP,25.12.2003

Katayama, K, Narihisa, H.(2001). Performance of Simulated Annealing –Based Heuristic for the Unconstraint Binary Quadratic Programming Problem. *European Journal of Operations Research*, 134, p.103-119.

Kirkpatrick, S., Gerlatt, C. D. Jr., and Vecchi, M.P.(1983). Optimization by Simulated Annealing, *Science*, 220, p.. 671-680.

Kubotani, H., Yoshimura, K.(2003). Performance Evaluation of Probability Functions for Multi-Objective Simulated Annealing. *Computers and Operations Research,* 30, p. 427-442.

Laporte G, Osman I. (1996). Metaheuristics: A bibliography, *Annals of Operations Research,* 63, p.513-623.

Lejeune, A.M.,(2003). Heuristic Optimization of Experimental Designs. *European Journal of Operations Research*, 147, p. 484-498.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M. N., Teller, A.H. and Teller, E.(1958) Equations of State Calculations by Fast Computing Machines, *J. Chem. Phys.* 21, p. 1087- 1092.

Pincus, M.(1970), A Monte Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems, *Operations Research,* 18, p. 1225-1228.

Pinedo, M. (1997). Scheduling Theory, Algorithms and Systems, New Jersey, Prentice-hall.

Randelman, R. E., and Grest, G.S.. (1997) N-City Traveling Salesman Problem - Optimization by Simulated Annealing, *J. Stat. Phys.* 45, p. 885-890.

Reeves, C.R (1993). Modern Heuristic Techniques for Combinatorial Problems (Blackwell, Oxford)

Singh,N, Ramajani D.,(1996). Cellular Manufacturing Systems, Design, Planning and Control,1st edition,Chapman and Hall.

T'kindt, V.; Monmarch, N.; Tercinet, F.; Lagugt, D.(2002). An Ant Colony Optimization algorithm to solve a 2-machine bicriteria flow shop scheduling problem, *European Journal of Operational Research,* 142, p.250–257.

Varadharajan, T.K.; Rajendran, C., (2005). A multi-objective simulated-annealing algorithm for scheduling in flow shops to minimize the make span and total flow time of jobs, *European Journal of Operational Research*, 167, p. 772–795.

Wolfram, S.(1984) Universality and Complexity in Cellular Automata, *Physica D: Nonlinear Phenomena*, 10(1-2),p. 1-35.