Meta heuristic for Minimizing Makespan in a Flow-line Manufacturing Cell with Sequence Dependent Family Setup Times

Behzad Nikjo[a], Javad Rezaeian[b,*]

[a] MSc, Department of industrial engineering, Mazandaran University of Science and Technology, Babol, Iran

[b] Assistant Professor, Department of industrial engineering, Mazandaran University of Science and Technology, Babol, Iran

Received 30 October, 2013; Revised 15 February, 2014; Accepted 06 May, 2014

Abstract

This paper presents a new mathematical model for the problem of scheduling part families and jobs within each part family in a flow line manufacturing cell where the setup times for each family are sequence dependent and it is desired to minimize the maximum completion time of the last job on the last machine (makespan) while processing parts (jobs) in each family together. Gaining an optimal solution for this type of complex problem in large sizes in reasonable computational time using traditional approaches or optimization tools is extremely difficult. A meta-heuristic method based on Simulated Annealing (SA) is proposed to solve the presented model. Based on the computational analyses, the proposed algorithm was found efficient and effective at finding good quality solutions.

Keywords: Scheduling, Simulate annealing, Flow line manufacturing, Setup times, Makespan.

1. Introduction

In a flow shop *N* jobs have to be processed on *M* machines and every job has to be processed at most once on a machine and each machine can only process one job at a time, flow shop is one of the most commonly used arrangements in process and finding an optimal scheduling of jobs (Mehravaran and Logendran, 2013). The flow line manufacturing cell with sequence dependent family setup times is called a pure flow shop. In today's world the wide applications of cellular manufacturing make flow line manufacturing cell scheduling problems (FMCSPs) with sequence dependent family setup times(SDFSTs) is a core topic in the field of scheduling. The FMCSPs with SDFSTs has an exuberance of implications in many industries, such as manufacturing printing circuit boards (PCBs) and TFT-LCD manufacturing; thus, the FMCSPs with SDFSTs is a critical research topic in the field of cellular manufacturing (CM). In fact, CM looks for reaching the efficiency of mass production by identifying and exploiting similarities of different parts (jobs) in their production processes (Bouabda et al., 2011; Saidi Mehrabad and Mirnezami ziabari, 2011). In a CM environment, a variety of machines and/or jobs (parts) are grouped together into part families, each of which is then assigned to a manufacturing cell (MC) (Yang and Liao, 1996). Therefore, MCSPs are especially concerned with sequencing part families and parts within families where each MC is dedicated to producing a specific number of part families (Lin et al., 2009b). When each job is processed on each machine in the same technological order of an MC, this is called a flow line MC (Schaller et al., 2000). Briefly, cellular manufacturing cell is a production system in which parts are grouped into dedicated manufacturing cell, according to a number of similarities in their design and similar characteristics (Solimanpur and Elmi, 2013). In this environment, machineries are located according to similarity of operation and kind of production size in 2 or more groups (Kamali Dolat Abadi et al., 2010). Also, Jeon and Leep (2006) have presented the design of CM as a tool for developing the production environment of machining centering by grouping the part families according to a number of similarities.

In cellular manufacturing systems (CMS), switching between jobs within a part family requires little or no setup time and for this cause it can be included in the processing times of each job. Nonetheless, switching from a job in one part family to a job in another family requires a major setup and, hence, requires an explicit treatment of setup times (Bouabda et al., 2011). And when jobs are grouped, the number of setups in the schedule is minimized; it makes good operational sense, when setups are very costly, in terms of money, time or both (Vakharia and Chang, 1990). In general, scheduling flow line

* Corresponding author Email address: j_rezaeian@ustmb.ac.ir

manufacturing cell problems involving setup times can be categorized into two types; the first type is sequence independent (Schaller, 2000; Schaller, 2001; Skorin-Kapov and Vakharia, 1993; Vakharia and Chang, 1990) and the second type is sequence dependent setup times (Al-Aomar, 2006; Hendizadeh et al., 2008; Lin et al., 2009a, b; Naderi et al., 2009; Ying et al., 2010).

This paper deals with flow line manufacturing cell scheduling problems with sequence dependent family setup times. This problem even in the two-machine case with only one job for each family is NP-hard in the strong sense (Gupta and Darrow, 1986). The problem considered in this paper is NP-hard.

Several researchers have given recognition to FMCSPs with SDFSTs in the manufacturing systems. Although the computational expense necessary to achieve an optimal solution increases exponentially with the problem size (Ying et al., 2012), some studies have developed exact solution algorithms for solving different FMCSPs (Das and Canel, 2005; Gupta and Schaller, 2006; Yang and Chern, 2000). Schaller et al. (2000) developed a branch and bound algorithm for the flow line manufacturing cell with SDFSTs with the objective of minimizing the makespan. Since, branch and bound algorithm can apply to only small problem sets including 3 or 4 families and 3 or 4 machines, they resorted to approximate methods to solve larger instances of the problem. Various heuristic algorithms have been proposed for this purpose. They proposed a composite two-stage scheduling methods for this problem called CMD and showed that it performed best for all problem sets. This algorithm is composed of three heuristic algorithms named *C*, *M* and *D*. Algorithm *C* which is based on CDS procedure of Campbell et al. (1970); it makes a new sequence of parts in each family at the first stage. Then algorithm M or Modified NEH procedure (Nawaz et al., 1983) is performed in the second stage to optimize the sequence of part families. Finally, Schaller et al. (2000) proposed an algorithm *D*, or Descent heuristic, and employed it for improving the solution obtained so far.

Recently, studies of FMCSPs have focused on the development of metaheuristic-based algorithms to minimize makespan. France et al. (2005) considered a flow shop manufacturing cell with SDFSTs and used two evolutionary algorithms (GA, MA) with local search to solve the problem. The performance of the proposed metaheuristics was very analogous to a slight advantage showed by the MA and both algorithms outperformed the CMD algorithm. Hendizadeh et al.(2008) considered the problem of finding a permutation schedule for a flow line manufacturing cell with SDFSTs to minimize makespan. They proposed Tabu Search (TS) based Meta heuristics and used the CMD algorithm (Schaller et al., 2000), the MA algorithm (France et al., 2005) as a benchmark compared with the proposed algorithm. Computational results show that the performance of Tabu search based metaheuristics is better than CMD algorithm and MA. In other research, Lin et al (2009b) dealt with a non-permutation flow line manufacturing cell with SDFSTs to minimize the completion time; three prominent types of metaheuristics (SA, GA, TS) were proposed and empirically evaluated. The empirical results showed that overall the improvement made by non-permutation schedules over permutation schedules for the due-date-based performance criteria were significantly better than that for the completion-time-based criteria. On other hand, Lin et al. (2011) developed an effective multi start simulate annealing (MSA) heuristics for solving the FMCSPs with SDFSTs to minimize makespan. Cheng and Ying (2011) addressed a FMCSPs with SDFSTs; they proposed a two-level iterated greedy (TLIG) heuristics to minimize makespan and then the performance of the proposed algorithm was compared against the eight existing algorithms (France et al., 2005; Hendizadeh et al., 2008; Lin et al., 2009b; Schaller et al., 2000). Bouabda et al.(2011) dealt with a flow line manufacturing cell with sequencing part families and sequence dependent family setup times; they developed an efficient integrative cooperative approach based on a genetic algorithm and a branch and bound procedure to solve the problem. The performance of the proposed method was tested by numerical experiments on a large number of representative problems.

Although in the literature Flow line manufacturing cell scheduling problems with sequence dependent family setup times have become an active area of research in the filed of scheduling, there are few studies about the use of non-permutation schedules (NPS). Accordingly, this paper considers a special type of FMCSPs with SDFSTs, while the defined problem can be a permutation schedule, because the sequence of part families is fixed on all machines. Also, it can be introduced as a non-permutation schedule, because the sequence of parts within each family can be changed between different machines. Thus, in this paper, we work for finding an optimal (near optimal for larger instances) schedule for the part families and jobs within each family in a flow line manufacturing cell with sequence dependent family setup times to minimize makespan. To this end, first, a new mathematical model for to solving the problem in small sizes is presented and then to solve the problem in larger dimensions a metaheuristic algorithm based on Simulated Annealing is suggested and evaluated.

The rest of this paper is organized as follows: In the next section, we explain the new mathematical model. In section 3, we propose a metaheuristic method based on SA to solve the given problem. Section 4 evaluates the computational results. Finally, the conclusion and future research directions are presented in section 5.

2. General Description and Formulation of the Problem

This section first describes the main ideas and assumptions made for this problem. Then the mathematical model is formulated.

2.1. General description

The problem can be classified as $F_m / fmls, S_{i'i}, r_i / C_{max}$: flow shop with m machine; group (family) scheduling problem (fmls); sequence dependent family setup times ($S_{i'i}$); Release date (r); and makespan minimization (C_{max}). Consider a sequencing problem that has a set $N = [j_1, \dots, j_n]$ of n given parts (jobs) to be processed on m machines. All parts to be processed are classified into one of F mutually and collectively exhaustive part families $f = [f_1, \dots, f_F]$ with n_k parts belonging to part family $f_k (k = 1, \dots, F)$ in which all jobs in the same family are processed together.

2.2. Formulation of the model

To describe the problem more clearly, a non-linear integer programming model is presented. Note that the model is modified by adapting the models, proposed by Stafford and Tseng (2002) for solving four different flowshop sequencing problems and Ying et al .(2012) for scheduling a no-wait flowshop manufacturing cell. The major assumptions made in this research are summarized as follows:

2.2.1. Model assumptions

- Let where n_k denotes the number of jobs in each family f_k and let the jobs be numbered sequentially such that the first n_1 jobs belong to the first family f_1, the next n_2 jobs belong to f_2 , and so on .Therefore, $n = n_1 + \cdots + n_F$.
- The individual job setup times are known and included in the job processing times.
- Pre-emption is not allowed, meaning once a job starts to be processed on a machine, the process cannot be interrupted before completion.
- The ready time of each job can be not zero; meaning that all parts are not available for processing at the start time.
- The number of jobs, their ready times, their processing times, the number of families, and the SDFSTs are non-negative integers and are known in advance.
- Each job can be processed by at most one machine at any given time. Furthermore, each machine can handle only one job at a time and is continuously available to process all scheduled jobs when required.
- The SDFSTs $S_{i'i}$ are incurred when job i belongs to part family f_y, is processed immediately after job i' that belongs to part family f_x on machine j .
- There is not any per specified priority between jobs and part families.
- The jobs' sequence within each family can to be changed between different machines, while the part families' sequence remains unchanged between different machines.

2.2.2. Model inputs (Parameters)

n : Number of jobs
m: Number of machines
F: Number of families
P_{ij}: processing time of job i on machine j
r_i: Release date of job i
$S_{i'i}$: set up times for job i processed immediately after job i' on all machines

2.2.3. Model outputs (Decision variables)

f_{kp} Binary variable taking value 1 if family k is assigned to sequence p and 0 otherwise.
f_{ikqj} Binary variable taking value 1 if job i belongs to qth position of family k on machine j and 0 otherwise.
g_{itj}= If job i is assigned to sequence position t on Machine j. (as following): Binary variable taking value 1 if $JP(i, j)$ is equal to t and 0 otherwise.
$W_{i'itj}$ Binary variable taking value if job i be in position t and processed immediately after job i' on the machine j and 0 otherwise.
V_{t1}: starting time of the tth job in the sequence on machine 1
C_{tj}: the completion time of the tth job in the sequence on machine j
C_{max}: the completion time of the nth job in the sequence on machine m

Equation (1) denotes minimizing the makespan. Equations (2) and (3) defined that every position of the family sequence is established by exactly one family and every family is assigned to exactly one position of the family sequence. Equation (4) determines the position of each family. Equations (5) and (6) ensure that every job is assigned to exactly one position in the sequence of its associated family and every position within the sequence of each family is established by exactly one job. Equation (7) determines the position of each job i on machine j. Equation (8) determines the ready time of jobs on first machine. Equations (9) to (11) compute the completion time for the jobs on the first machine. The job completion time in each position at each machine in the manufacturing cell is represented by Equations (12) to (14), and Equation (15) defines the completion time of the last job on the last machine (makespan). Equation (16) shows the binary variables and other variables.

3. The Proposed SA Based Meta-Heuristics

Simulated annealing (SA) based meta heuristics algorithms is one of the most popular efficient procedures for addressing combinatorial optimization problems. SA

3.1.1. Objective function and the constraints

$$\boldsymbol{minC_{max}} \tag{1}$$

Subject to:

$$\sum_{p=1}^{F} f_{kp} = 1 \qquad k = 1, \dots, F \tag{2}$$

$$\sum_{k=1}^{F} f_{kp} = 1 \qquad p = 1, \dots, F \tag{3}$$

$$FP(k) = \sum_{p=1}^{F} f_{kp} \times p \qquad k = 1, \dots, F \tag{4}$$

$$\sum_{i=a(k)}^{b(k)} f_{ikqj} = 1 \qquad k = 1, \dots, F; \quad q = 1, \dots, n_k; \; j = 1, \dots, m \tag{5}$$

$$\sum_{q=1}^{n_k} f_{ikqj} = 1 \qquad k = 1, \dots, F \;; \; a(k) \le i \le b(k); \; j = 1, \dots, m \tag{6}$$

$$JP(i,j) = \sum_{k=1}^{F} \sum_{p'=0}^{FP(k)-1} f_{kp' \times n_{p'}} + \sum_{q=1}^{n_k} f_{ikqj} \times q \qquad k = 1, \dots, F \;; \; a(k) \le i \le b(k); \; j = 1, \dots, m \tag{7}$$

$$V_{t1} \ge \sum_{i=1}^{n} R_i \times g_{it1} \qquad t = 1, \dots, n \tag{8}$$

$$C_{11} = V_{11} + \sum_{i=1}^{n} P_{i1} \times g_{i11} \tag{9}$$

$$C_{t1} \ge C_{t-1,1} + \sum_{i'=1}^{n} \sum_{i=1}^{n} S_{i'i} \times W_{i'it1} + \sum_{i=1}^{n} P_{i1} \times g_{it1} \qquad t = 2, \dots, n \tag{10}$$

$$C_{t1} \ge V_{t1} + \sum_{i=1}^{n} P_{i1} \times g_{it1} \qquad t = 2, \dots, n \tag{11}$$

$$C_{1j} \ge \sum_{i=1}^{n} g_{i1j} \times \sum_{t'=1}^{n} C_{t',j-1} \times g_{i,t',j-1} + \sum_{i=1}^{n} P_{i1} \times g_{i1j} \qquad j = 2, \dots, m \tag{12}$$

$$C_{tj} \ge \sum_{i=1}^{n} g_{itj} \times \sum_{t'=1}^{n} C_{t',j-1} \times g_{i,t',j-1} + \sum_{i=1}^{n} P_{ij} \times g_{itj} \qquad t = 2, \dots, n \,; \; j = 2, \dots, m \tag{13}$$

$$C_{tj} \ge C_{t-1,j} + \sum_{i'=1}^{n} \sum_{i=1}^{n} S_{i'i} \times W_{i'itj} + \sum_{i=1}^{n} P_{ij} \times g_{itj} \qquad t = 2, \dots, n \,; \; j = 2, \dots, m \tag{14}$$

$$C_{max} = C_{nm} \tag{15}$$

$$f_{kp}, f_{ikqj}, g_{itj}, w_{i'itj} = \{0,1\} \quad ; W_{i'itj} = g_{i',t-1,j} \times g_{itj} \;; \forall i', i, t, j \,; a(k) = 1 + \textstyle\sum_{k'=0}^{k-1} n_{k'} \;; b(k) = \textstyle\sum_{k''=1}^{k} n_{k''} \tag{16}$$

is based on the simulation of the energy changes in a physical annealing process where solids is heated and cooled to gain a crystalline structure, in which slow cooling of metal produces a good, low energy state crystallization, whereas fast cooling produces poor crystallization. This algorithm was introduced by metropolis et al. (1953), and applied to optimization problems by Krikpatrick et al. (1983). The solution representation, the initial solution and neighborhood and the SA procedure and parameters are discussed in the following part:

3.1. Solution representation

Since the operation sequence of part families and the sequence of parts within each family on each machine can be different, F part families to be processed on m machines, the solution representation consists of m+mf (or m(1+F)) section. The first m section indicates the operation sequence of part families on each machine, and the mf section corresponds to the operation sequence of part within each part family on each machine. For example, there are three machines, three families and a total of 11 parts to be scheduled; a solution representation as shown in Table 1 can be decoded as below. The operation sequence of part families for machines 1, 2, and 3 are 2-3-1, 3-1-2, and 2-1-3, respectively. Meanwhile, the operation sequence for parts in part families 1,2, and 3 are on machine 1: 1-2-3-4,4-2-3-1,1-2-3-4; on machine 2: 5-6-7,6-7-5,6-7-5; on machine 3: 9-8-10-11,8-9-10-11,11-10-9-8, respectively

3.2. Initial solution & neighborhood

The initial solution is generated by randomly ordering the sequence of part families on each machine and the sequence of the parts within each family on each machine. Let X be as current solution, where the set N(X) is the set of solutions neighboring X. N(X) is obtained by a swap operation on the sequence of families and the sequence of the parts within each family. However, in the same family, two jobs are randomly selected and swapped with each other. Similarly, for the sequence of families, two families are randomly selected and swapped directly.

3.3. SA procedure and parameters

The proposed SA approach is briefly described as such: first, SA starts from an initial solution X as Incumbent solution. After setting parameters, a series of moves are made until the final temperature (stopping criterion) is met. At each iteration, the next solution S is generated from the neighborhood of the current solution X. the new solution is accepted or rejected by another random rule. A parameter t, called the temperature that controls the acceptance rule. Let $C_{max}(x)$ denote the calculation of the objective function value of X, and ΔC the variation between $C_{max}(x)$ and $C_{max}(s)$; that is computed $\Delta C= C_{max}(s)- C_{max}(x)$.if $\Delta C\le 0$, solution S is accepted, otherwise, solution S is accepted with some probability depending on the current temperature (t_i) and the amount of degradation of the objective function $e^{-\Delta C / t_i}$. The algorithm proceeds by attempting a determined number of neighborhood moves at each temperature t_i, while temperature is gradually decreased under an especial mechanism called the cooling schedule. The cooling schedule used in this paper is as follows (Lundy and Mees, 1986):

$$t_i = \frac{(t_0 - t_f)}{N(i+1)} + t_0 - \frac{(t_0 - t_f)(\mathrm{N}+1)}{N} \quad i = 1, \dots, N$$

Where, t_0 , t_f and N are initial temperature, final temperature and the number of temperature levels between t_0 and t_f . According to the fact that the performance of simulated annealing forcefully depends on the proper selection of its parameter values, we applied the Taguchi method for algorithm calibration.
The general scheme (pseudo code) for our proposed SA is presented in Figure.1.

Table 1
An illustration of the solution representation

	Machine1	Machine2	Machine3
Sequence of the part families in	2-3-1	3-1-2	2-1-3
Sequence of the Parts in part familiy1 in	1-2-3-4	4-2-3-1	1-2-3-4
Sequence of the Parts in part familiy2 in	5-6-7	6-7-5	6-7-5
Sequence of the Parts in part familiy3 in	9-8-10-11	8-9-10-11	11-10-9-8

Procedure Simulated Annealing

```
Initialize algorithm parameters (T0,N)
X=S0                    /Generation of the initial solution/
T=T0                    /Starting temperature/
for a=1 to N  do        /temperature counter/
Generating a neighbor solution(S)  from current solution  (X) at random
     if Cmax(S) < Cmax(X)    then        /Acceptance criterion/
        X=S
     else
        if  random(0,1) <exp (-Cmax(S)+Cmax(X) /T  )  then
         X=S
        endif
     endif
  T=g(T)          /temperature update/
endfor
```

Fig. 1. pseudo code of the proposed SA

3.4. Algorithm's calibration

In this section, we investigate the impact of different parameters on the performance of our SA by means of the Taguchi method. Taguchi method was introduced by Taguchi (1986). Taguchi method is more effective; in fact, it is an optimization technique which can study a large number of decision variables with a minimal number of experiments. In Taguchi method, the factors are categorized into two main groups: controllable and noise factors. Controllable factors are set and controlled by designers, whereas noise factors are varied by the environment and usage, while we have no direct control (Naderi et al., 2011). Due to the fact that elimination of the noise factors is impractical and often impossible, Taguchi method pursues to minimize the effect of noise and to determine the optimal level of the important controllable factors base on the concept of robustness (Al-Aomar, 2006; Tsai et al., 2007). In addition to determining the optimal levels, Taguchi establishes the relative importance of each factor in terms of their main effect on the performance of the algorithm (Naderi et al., 2009). In this method, there is a transformation of the values of response variable into a ratio which is the measure of variation called the signal-to-noise(S/N).
Here, the term " signal" denotes the desirable value (mean response variable) and "noise" denotes the desirable value (standard division) as the S/N ratio indicates the amount of variation existing in the response variable. The objective is to maximize the signal- to- noise ratio. Taguchi classifies all objective functions into three groups: the smaller-the-better type, the larger-the-better type, and nominal. Each group has its own appropriate formulas for calculating the S/N ratio. Since, the objective function in this research is minimizing the makespan and it is categorized in the smaller-the-better type, its corresponding S/N ratio is (Roy, 1990):

$$S/_N = -10\log_{10}[objective\ function]^2$$

For the parameter design procedure using Taguchi method, we refer the readers to the textbooks of Roy (1990) and the paper of Cheng and Chang (2007).
In this study, the SA factors that need to be tuned are as follows: number of iteration (N), cooling schedule (annealing) and initial temperature (T_0). These parameters and their levels are shown in table2. We run the SA for each trial of Taguchi experiment. The results are transformed into S/N ratio. Figure 2 shows the mean S/N ratio obtained for each level of the factors. Finally, the optimal levels of factors are shown in Table 3.

4. Computational Results and Discussion

4.1. Test problems

To put it briefly, in this study, the processing times for each machine and release dates for jobs were randomly generated using a uniform distribution as follows:

- Processing times~[1,10]
- Release dates~[1,10]

Table 2
Factors and their levels

factor	level	candidate Values
N	1	2000
	2	5000
	3	10000
Annealing cooling) (schedule	1	Linear
	2	Exponential
	3	Hyperbolic
Initial F	1	30
	2	50
	3	100

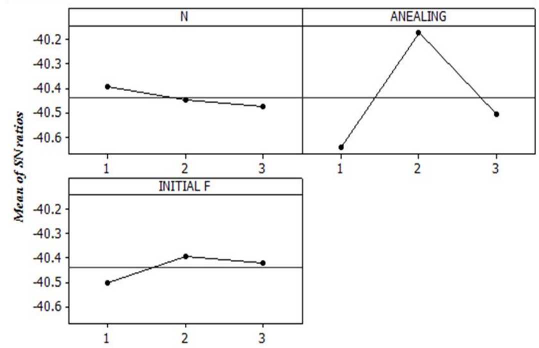

Fig. 2. The mean S/N ratio plot for each level of the factors

Table 3
Optimal level of factors

factor	Optimal level	Best value
N	1	2000
Annealing (cooling schedule)	2	Exponential
Initial F	2	50

Since problem difficulty is presumably to depend on whether there is a balance between average processing times and average family setup times, hence, three different classes of family setup times were randomly generated from the following uniform distributions:

- Small setup (SSU)~[1,20] implies that the ratio of average family setup time to average job processing time is almost 2:1.
- Medium setup (MSU)~[1,50] implies that the ratio of average family setup time to average job processing time is almost 5:1.
- Large setup (LSU)~[1,100] implies that the ratio of average family setup time to average job processing time is almost 10:1.

Respectively, the number of families varies between [3 10], the number of jobs varies between [1 10] and the number of machines varies between [3 10]. With respect to the existing data, ten instances were generated for each class. As an example, FMCMSU56 denotes an instance of a flow line manufacturing cell problem with medium family setup times and five families that is processed on six machines.

4.2. Computational results

The problem considered in this paper can be solved in small sizes with mathematical model by lingo9. We solve six small instances of this problem and the same solution is gained (mathematical model and proposed SA). But, since gaining an optimal solution for this type of complex problem in large sizes in reasonable computational time using traditional approaches or optimization tools is extremely difficult. A prominent type of metaheuristics - a simulated annealing- is proposed and empirically evaluated. The obtained computational results of lingo9 and SA in small sizes are shown in Table 4. This table shows that mathematical model can solve small instances exactly, but when the size of instances increased, the mathematical model is unable to obtain the optimal or near optimal solution in a reasonable time. Thus, the proposed metaheuristic algorithm was implemented using Matlab 7.12 and run on a laptop with an Intel core (i7)-2630QM (2 GHz) CPU and 8 gig memories. The minimum value of each instance obtained by the proposed algorithm was compared with its average value that is computed as follows:

$$\%\,Gap_{SA} = \frac{Ave_{SA} - Min_{SA}}{Min_{SA}} \times 100$$

Table 5 lists the minimum value (as best solution), maximum value, average value, average computational time (CPU time in seconds), and the gap percentage (a measure for the variation in a set of data that looks at the variation as a proportion of the average or target value) of each instance with three classes of family setup times (small, medium, and large). These results demonstrate that when the dimensions of problems are increased, the gap percentage is decreasing. Also, this table indicates that SA has reasonable time to gain the optimal or near optimal solution. Finally, Figure 3 shows the gap percentage of each instance of each class.

A review of the results illustrated in Table 5 and Figure 3 also show that the majority of Gap_{SA} levels of the problems are less than 5%, that expresses convergence of SA. Therefore, the obtained results show that the proposed algorithm is efficient, effective and reliable for the operation managers in minimizing makespan for the flow line manufacturing cell problem with sequence dependent family setup times.

Table 4
The Computational results of lingo9 and SA for small sizes problems

instance	Number of jobs	Number of family	Number of machines	Solution by lingo	Solution by SA	%gap
1	3	2	2	12	12	0
2	4	2	2	14	14	0
3	3	2	3	15	15	0
4	4	3	3	17	17	0
5	5	3	3	22	22	0
6	6	3	3	27	27	0

Table 5
The computational results of proposed SA for different classes of family setup times

instances	N	m	F	Setup times class*	Proposed SA			Ave CPU time(s)	gap SA(%)
					Min (best)	Max	Ave		
				S	67	67	67	0.348	0.00
FMCSSU33	5	3	3	M	67	67	67	0.334	0.00
				L	100	100	100	0.338	0.00
				S	83	94	88.4	0.441	6.51
FMCSSU34	7	4	3	M	83	96	89.1	0.443	7.35
				L	89	107	93	0.444	4.49
				S	86	98	90.6	0.568	5.35
FMCSSU44	9	4	4	M	129	143	135.2	0.560	4.81
				L	142	163	150.2	0.559	5.77
				S	132	146	137.4	0.817	4.09
FMCSSU55	12	5	5	M	165	193	185.1	0.819	12.18
				L	225	240	232.7	0.822	3.42
				S	143	157	148.1	0.981	3.57
FMCSSU56	13	6	5	M	206	232	219.8	0.975	6.70
				L	286	315	303.6	0.982	6.15
				S	171	192	181.7	1.01	6.26
FMCSSU65	16	5	6	M	231	255	241.8	1.02	4.68
				L	317	356	343	1.02	8.20
				S	185	201	194.2	1.25	4.97
FMCSSU66	18	6	6	M	264	282	275.5	1.26	4.36
				L	367	432	411.3	1.26	12.07
				S	295	315	305.7	2.06	3.63
FMCSSU88	24	8	8	M	391	443	427.2	2.05	9.26
				L	515	577	559.8	2.07	8.70
				S	356	373	363.9	2.51	2.22
FMCSSU108	29	8	10	M	439	477	454.1	2.49	3.44
				L	711	763	735.9	2.51	3.50
				S	457	483	469	3.48	2.63
FMCSSU1010	35	10	10	M	589	620	603.3	3.47	2.43
				L	799	848	821.3	3.45	2.79

*S= small setup times; M= medium setup times; L= large setup times.

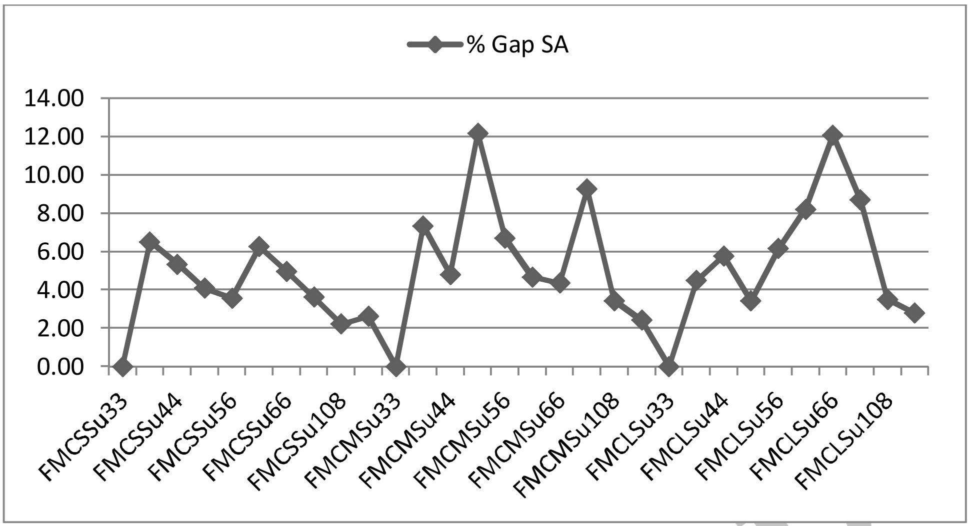

Fig. 3. The gab percentage for all instances

5. Conclusions

In this paper, the flow line manufacturing cell scheduling problems was investigated with sequence dependent family setup times. Minimizing the maximum completion time of the last job on the last machine (makespan) was considered as objective function. At first, the problem under consideration and assumptions were formulated as a non-linear integer programming model. This model is capable of solving small instances. Then, an effective simulated annealing algorithm was applied to tackle the problem. Computational experiments were performed to probe whether the proposed SA is efficient, effective and flexible. Many topics in the area of FMCSPs with SDFSTs remain for future research.

References

[1] Al-Aomar, R., (2006) Incorporating robustness into Genetic Algorithm search of stochastic simulation outputs. Simul Model Pract Theory 14:201–223.

[2] Bouabda, R., Jarboui, B., Eddaly, M.,&Rebai, A.,(2011) A branch and bound enhanced genetic algorithm for scheduling a flowline manufacturing cell with sequence dependent family setup times. Computers & Operations Research 38,387–393.

[3] Campbell, H.G., Dudek, R.A., Smith, M.L., (1970) A heuristic algorithm for the n-job, m- machine sequencing problem. Management Science 16:630–637.

[4] Cheng, B.W., Chang, C.L., (2007) A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm. Expert Syst Appl 32:415–421.

[5] Cheng, H.M., Ying, K. C., (2011) Minimizing makespan in a flow-line manufacturing cell with sequence dependent family setup times .Expert Systems with Applications, 38, 15517-15522.

[6] Das, S.R., Canel, C., (2005) an algorithm for scheduling batches of parts in a multi-cell flexible manufacturing system. Int J Prod Econ 97:247–262.

[7] França, P.M., Gupta, J.N.D., Mendes, A.S., Moscato, P., & Veltink, K.,(2005) Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. Computers and Industrial Engineering, 48, 491–506.

[8] Gupta, J.N.D., Darrow, W.P., (1986) the two machine sequence dependent flowshop scheduling problem. European Journal of Operational Research 24, 439-446.

[9] Gupta, J.N.D., Schaller, J.E., (2006) Minimizing flow time in a flow-line manufacturing cell with family setup times. Journal of the Operational Research Society 57:163–76.

[10] Hendizadeh, H., Faramarzi, H., Mansouri, S.A., Gupta, J.N.D, Elmekkawy, T.Y., (2008) Meta- heuristics for scheduling a flow shop manufacturing cell with sequence dependent family setup times. International Journal of Production Economics 111:593–605.

[11] Jeon, G., Leep, H.R., (2006) Forming part families by using genetic algorithm and designing machine cells under demand change. Computers and Operations Research, 33:263–83.

[12] Kamali Dolat Abadi, A.h., Pasandideh, S.H.R., Abdi Khalife, M., (2010) Layout of Cellular Manufacturing System in Dynamic Condition. Journal of Industrial Engineering 5, 43-54.

[13] Kirkpatrick, F., Gelatt, C., Vecci, M., (1983) Optimization by simulated annealing. Science, 220(4598), 671-680.

[14] Lin, S. W., Gupta, J. N. D., Ying, K. C., & Lee, Z. J.,(2009a) Using simulated annealing to scheduling a flowshop manufacturing cell with sequence dependent family setup times.International Journal of Production Research, 47, 3205– 3217.

[15] Lin, Sh.W. Ying, K.Ch. Lee, Z.J., (2009b). Metaheuristics for scheduling a non-permutation flowline manufacturing cell with sequence dependent family setup times. Computers & Operations Research 36:1110 – 1121.

[16] Lin, S. W., Ying, K. C., Lu, C.C& Gupta, J. N. D.,(2011) Applying multi-start simulated annealing to schedule a flow line manufacturing cell with sequence dependent family setup times. International Journal of Production Economics, 130, 246-254.
[17] Lundy, M., Mees, A., (1986) Convergence of an annealing algorithm. Math Program 34(1), 111–124.
[18] Mehravaran, Y., Logendran, R., (2013) Non permutation flow shop scheduling in a supply chain with sequence dependent setup times. International Journal of Production Economics, 141,639-645.
[19] Metropolis, N., Rosenbluth, A. Rosenbluth, M., Teller, A., (1953) Equations of State Calculations by Fast Computing Machines. Journal of Chemical Physics 21(6), 1087-1092.
[20] Naderi, B., Zandieh, M., Khaleghi Ghoshe Balagh, A., Roshanaei, V., (2009) an improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. Expert Syst Appl 36, 9625–9633.
[21] Naderi, B., Zandieh, M., Fatemi Ghomi, S.M.T., (2009) an Iterated Greedy Algorithm for Flexible Flow Lines with Sequence Dependent Setup Times to Minimize Total Weighted Completion Time. Journal of Industrial Engineering 3, 33–37.
[22] Naderi, B., Fatemi Ghomi, S.M.T., Aminnayeri, M., Zandieh, M., (2011) Scheduling open shops with parallel machines to minimize total completion time. Journal of Computational and Applied Mathematics 235, 1275–1287.
[23] Nawaz, M., Enscore, E., Ham, I., (1983) A heuristic algorithm for the m-machine n-job flow shop sequencing problem. OMEGA, International Journal of Management Science, 11, 91–95.
[24] Roy, R.K., (1990) A Primer on the Taguchi Method. Van Nostrand Reinhold, New York.
[25] Saidi Mehrabad, M., Mirnezami ziabari, S.M., (2011) Developing a Multi-objective Mathematical Model for Dynamic Cellular Manufacturing Systems. Journal of Industrial Engineering 7, 1-9.
[26] Schaller, J.,(2000) A comparison of heuristics for family and job scheduling in a flow- line manufacturing cell. International Journal of Production Research, 38(2):287–308.
[27] Schaller, J., (2001) a new lower bound for the flow shop group scheduling problem. Computers and Industrial Engineering, 41(2):151–61.
[28] Schaller, J.E., Gupta, J.N.D., Vakharia, A.J., (2000) Scheduling a flowline manufacturing cell with sequence dependent family setup times. European Journal of Operational Research, 125:324–339.
[29] Skorin-Kapov, J., Vakharia, A.J., (1993) Scheduling a flow-line manufacturing cell: a tabu search approach. International Journal of Production Research, 31(7): 1721–1734.
[30] Stafford, E.F., Tseng, F.T., (2002) two models for a family of flowshop sequencing problems. Eur J Oper Res 142:282–293.
[31] Solimanpur, M., Elmi, A., (2013) A tabu search approach for cell scheduling problem with makespan criterion. Int J Production Economics 141,639–645.
[32] Taguchi, G., (1986) Introduction to quality engineering, White Plains: Asian Productivity Organization/UNIPUB.
[33] Tsai, J.T., Ho, W.H., Liu, T.K., Chou, J.H., (2007) Improved immune algorithm for global numerical optimization and job shop scheduling problems. Appl Math Comput 194(2), 406–424.
[34] Vakharia, A.J., Chang, Y.L., (1990) a simulated annealing approaches to scheduling a manufacturing cell. Naval Research Logistics, 37,559–577.
[35] Wilson, A.D., King, R.E., Hodgson, T.J.,(2004) Scheduling non-similar groups on a flow line: multiple group setups. Robotics and Computer-Integrated Manufacturing 20, 505–515.
[36] Yang, D.L., Chern, M.S., (2000) Two-machine flowshop group scheduling problem. Computers & Operations Research 27,975–985.
[37] Yang, W.H., Liao, C.J., (1996) Group scheduling on two cells with intercell movement. Computers & Operations Research 22:997–1006.
[38] Ying, K. C., Gupta, J. N. D., Lin, S. W., & Lee, Z.J., (2010) Permutation and Non permutation schedules for the flowline manufacturing cell with sequence dependent family setups. International Journal of Production Research 48, 2169–2184.
[39] Ying, K.C., Lee, Z.J., Lu, C.C., & Lin, S.W., (2012) Metaheuristics for scheduling a no-wait flowshop manufacturing cell with sequence-dependent family setups. International Journal of Advanced Manufacturing Technology 58,671–682