Int J Adv Manuf Technol (2016) 84:1841–1860
DOI 10.1007/s00170-015-7715-9

ORIGINAL ARTICLE

A new meta-heuristics for optimum design of loop layout in flexible manufacturing system with integrated scheduling

K. Mallikarjuna[1] · **V. Veeranna**[2] · **K. Hemachandra Reddy**[3]

Received: 6 November 2012 / Accepted: 11 August 2015 / Published online: 22 September 2015

Abstract Flexible manufacturing system is the inception of an innovative manufacturing revolution that will credibly lead the manufacturing trade to levels of automation which is to be taken granted currently in the process-related industries. This paper speaks about multi-objective optimization related to flexible manufacturing systems (FMS) scheduling which act as a constraint in configuring the loop layout in optimum manner by various algorithms, i.e., meta-heuristics like genetic algorithm (GA), simulated annealing (SA), etc. The various loop layout problems are tested for enactment of objective function with respect to computational time and number of iterations involved in GA and SA. A simulation code is generated using programming language and executed using integrated development environment (IDE) tool. A comparative analysis of simulation results of different meta-heuristics with literature results has been done. The performance of this GA is proved to be the best among all the algorithms considered for this work.

Keyword Flexible manufacturing systems · Loop layout · Genetic algorithm · Simulated annealing · IDE tool

1 Introduction

In the current scenario, automated manufacturing industries are under great pressure caused by the rising cost of energy, materials, labors, capital, and intensifying worldwide competition. While these trends will remain for a long time, the problem fronting manufacturing today run much yawning. In many cases, they stem from the very nature of the manufacturing process itself. In order to overcome that, flexible manufacturing systems (FMSs) are regarded as one of the most efficient methods to use in reducing or eliminating manufacturing problems. FMS is more than a technical solution; it is a business-driven solution leading to improve profitability through reducing lead times and inventory levels and improved manufacturing effectiveness through increased operational flexibility, predictability, and control. Flexible manufacturing system [1] combines collection of machine tools which are termed as numerical control machines that can arbitrarily process a cluster of jobs, taking automated material management and workstation control to balance resource exploitation over which the system can accept automatically to variation in jobs manufacture, amalgams, and stages of yield. The objective of FMS is flexibility in production without compromising the quality of products. Flexibility can mean future cost avoidance. This type of flexibility would be common among automotive and manufacturers, where high part volumes are common but future change in market demand are expected and anticipated.

Material handling is important, yet sometimes it is an overlooked aspect of automation. The main function of an MHS is to supply the true materials at the exact locations and at the right time; the cost of material handling has high priority in total cost of production. It means handling

---

✉ K. Mallikarjuna
sadhana_malli@rediffmail.com

[1] Department of Mechanical Engineering, G. Pullaiah College of Engineering and Technology, Kurnool, India

[2] Department of Mechanical Engineering, Brindavan Institute of Technology & Science, Kurnool 518001, Andhra Pradesh, India

[3] Department of Mechanical Engineering, JNTU, Anantapur 515001, India

cost is equal to two-thirds of the total manufacturing cost [2]. This fraction varies depending on type and quantity of production and the degree of automation in the material handling function. Finally, material handling plays an important role in FMS.

The FMS layout involves allocating diverse reserve for attaining full competence. The arrangement has an influence on the make span and cost [3] which should be determined in the inception of the FMS [4]. In practice, the most commonly used type of FMS layouts [5] are as follows:

1. Line or single row layout
2. Loop layout
3. Stepladder layout
4. U-shaped layout

Among the above layouts, this paper focus on loop layout design with integrated scheduling using genetic algorithm (GA) and simulated annealing (SA).

2 Literature survey

During former epoch, FMS layout design with integrate scheduling has got extra emphasis since of its prominence from both hypothetical and real-world points of sight. Early investigation was intense, mainly on the origination and explanation of the problem as the mathematical model, such as branch and bound method and dynamic programing [6], but these approaches can only be useful for small problems. Heuristic methods can solve the small problem and also combinatorial optimization problems. The heuristic methods are usually computationally efficient, but easily trap into local optimal solution and no assurance that they will catch optimal solutions. Recently, meta-heuristic has been applied, such as tabu search, simulated annealing, genetic algorithm, and ant colony optimization. Wei-jun Xia and Zhi-ming Wu [7] developed a new approximation algorithm for the problem of finding the minimum makespan in the job-shop scheduling environment. They combine the two existing algorithm and developed new algorithm which is known as hybrid optimization algorithm. Kumar et al. [4] introduced an ant colony optimization (ACO) algorithm for the layout design with integrated scheduling by applying priority dispatching rules using Giffler and Thompson algorithm. Tiwari and Chang [8] proposed the pareto-optimal block-based EDA using bivariate model for multi-objective flow shop scheduling problem. They apply a bivariate probabilistic model to generate block which have the better diversity along with the non-dominated sorting technique to filter the solutions.

Muthuswamy and Vélez-Gallego [9] propose a mathematical formulation and present a particle swarm optimization (PSO) algorithm. Their objective is to batch the jobs and sequence the batches such that the makespan is minimized. Ayough and Zandieh [10] present a new model dealing with the job rotation scheduling problem. They used new software called Lingo software for simulating two search algorithms, GA and imperialist competitive algorithm (ICA), designed to conquer the algorithmic complexity of model, and their parameters adjusted using Taguchi's method were used. Filho and Barco [11] proposed that a classification system encompasses six main dimensions: FMS type, types of resource constraints, job description, scheduling problem, measure of performance, and solution approach. They analyzed literature using the proposed classification system, which provides the following results regarding the application of GAs to FMS scheduling:

1. Combinations of GAs and other methods were relatively important in the reviewed papers.
2. Although most studies deal with complex environments concerning both the routing flexibility and the job complexity, only a minority of papers simultaneously consider the variety of possible capacity constraints on an FMS environment, including pallets and automated guided vehicles.

Udhayakumar and Kumanan [12] generated an active schedules and optimal sequence of job and tool that can meet minimum makespan schedule for the flexible manufacturing system. They proposed ACO algorithm to derive near optimal solutions which adopt the Extended Giffler and Thompson algorithm for active feasible schedule generation. They used this proposed algorithm to solve the number of problems taken from the literature. Costa and Cappadonna [13] focused on skilled workforce assignment (SWA) to machines of a given shop floor may represent a key issue for enhancing the performance of a manufacturing system. Their literature addressed about the group scheduling problems and identified the effect of human factor on the performance of serial manufacturing systems which was ignored by researchers.

Javadian and Fattahi [14] addressed the hybrid flow shop scheduling problems considering time lags and sequence-dependent setup times. They presented a mathematical model which is capable of solving the small size of the considered problem in a reasonable time. Ranjbar and Razavi [15] proposed a new method to synchronously make the arrangement and planning decisions in a job shop situation. Wangta and Pongcharoen [16] presented the application of SA and TS for minimizing the material handling distance associated with the layout required for manufacturing process of multiple products. They developed a computer-based machine layout designed tool and tested using five datasets from literature. Khamseh and Jolai [17] integrates flexible flow shop group scheduling problem with sequence-dependent setups and preventive maintenance activities in order to minimize the total completion time (makespan). They exploited the Taguchi robust parameter design method. Karthikeyan and Asokan [18] presented a hybrid discrete firefly algorithm to solve the multi-objective flexible job shop scheduling problem with limited resource constraints. They considered the minimization of makespan, maximal workload, and total workload of machines as three different objectives and instead of applying the standard firefly algorithm. Pooranian and Shojafar [19] developed a new hybrid scheduling algorithm GGA that combines GA and the gravitational emulation local search (GELS) algorithm.

3 Problem description

- The problem formulation procedure adopted by Liu and Abraham [20] has been used in this research work. We focus on design of loop layout in flexible manufacturing system with flexible batch scheduling problem (FBSP) as constraint with the following parameters.
- Jobs$J=\{j_1,j_2,\ldots\ldots\ldots\ldots j_n\}$
- Batches $B=\{B_1,B_2,\ldots\ldots\ldots\ldots\ldots\ldots B_n\}$ is a set of n jobs /n batches to be scheduled respectively. Each job J_i consists of a predetermined sequence of operations. $O_{i,j}$ is the operation j of J_i.
- Machines $M=\{M_1,M_2,\ldots\ldots\ldots\ldots\ldots\ldots M_m\}$ is a set of m machines.
- Slots $S=\{S1,S2,S3\ldots\ldots\ldots S_m\}$ is a set of N fixed slots
- Flexible FSP usually is classified into two types as follows:

- Total FBSP {T-FBSP}; every operation can be managed on any machine of M.
- Fractional FBSP {P-FBSP}; every operation can be handled on one machine of set of M.
- Authors adopted P-FBSP integrated with facility layout problems for our research work.

3.1 Multi objective mathematical models

In this section, we introduce the multi-objective function and use it to solve the flexible batch scheduling problems which are integrated with loop layout pattern design leads to minimize the make span and to obtain an optimal layout plan for the machines by minimizing the total transportation cost increased in the system.

3.1.1 Notations

The notations [21] which are used to develop a mathematical model of the design of line layout are defined and interpreted as follows.

i	Part type index i, $i'=1,2,3,\ldots\ldots,n$
j	Process index $j=1,2,3,\ldots\ldots,p$
k	Machine index k, $k'=1,2,3,\ldots\ldots,m$
n	Number of batches / job
m	Number of machines
S_{maxi}	Make span of system maximum completion time
$s_{\mathrm{n,m}}$	Make span of system
$S_{\mathrm{i,j,k}}$	Partial make span without predecessors
$s_{\mathrm{i,j+1,k}}$	Enhanced make span with predecessors
$T_{\mathrm{i,j}}$	The duration (processing time) of operation j of job i
$T_{\mathrm{i,j+1}}$	The duration of operation $j=1$ of job i
$O_{\mathrm{i,\ k}}$	Operation of job on corresponding machine
$B_{\mathrm{i,\ k}}$	Batch processing on corresponding machine
$f\,(i,j,k)$	Vector representing corresponding operation of job on specified machine
X	Corresponding layout
M	Total number of machines contained in the manufacturing system
m_{i}	Machine in slot n_1
m_{j}	Machine in slot n_{N}
N	Number of slots
$\mathrm{MH}_{m1,m2}$	Material handling cost between machines m_1 and m_2 (m_1 $m_2=1,2,3,\ldots\ldots.,M$)
$\mathrm{RD}_{n1,n2}$	Rectangular distance between machinery locations n_1 and n_2 (n_1 $n_2=1,2,3,\ldots\ldots N$)
$\mathrm{MF}_{m1,m2}$	Amount of material flow among machines m_1 and m_2 (m_1 $m_2=1,2,3,\ldots.M$)
$\mathrm{LOC}_{\mathrm{mi}}$	Loading cost from loading station to machines
$\mathrm{ULOC}_{\mathrm{mi}}$	Unloading cost from unloading station to machines

3.2 Objective functions

(I) Minimize make span $F\,(S_{\mathrm{maxi}})$

Minimize, $F(S_{\max})=S_{n,m}$

Sub to

1. conjunctive constraints

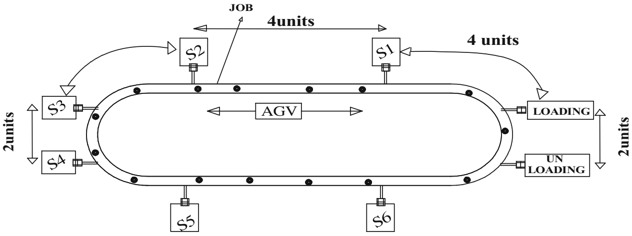

Fig. 1 Loop layout arrangements of FMS for six machines

$S_{i,j,k} \leq S_{i,j+1,k} - T_{i,j+1}$, for $j = 1, 2, 3 \ldots p$
$S_{i,j,k} \geq 0$, for $j = 1, 2, 3 \ldots p$

2. Resource constraints

$O_{I,j\,k}=1$ if job i scheduled before job i' on machine k=0, otherwise for O € S (i,j,k)
for j=1, 2, 3…p

3. Disjunctive constraints

$B_{i,\,k}=1$ if job i processed only once on machine k=0, otherwise for B € S (i,j,k)
for i, i'=1, 2, 3….. n
k, k'=1, 2, 3 …m

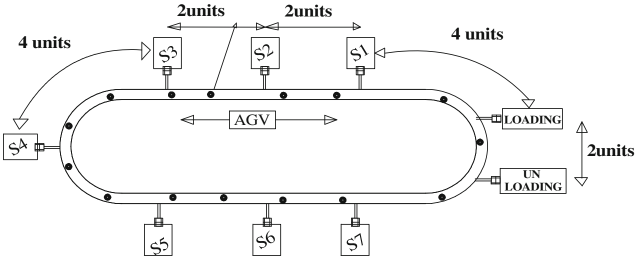

Fig. 2 Loop Layout Arrangements of FMS for 7 machines

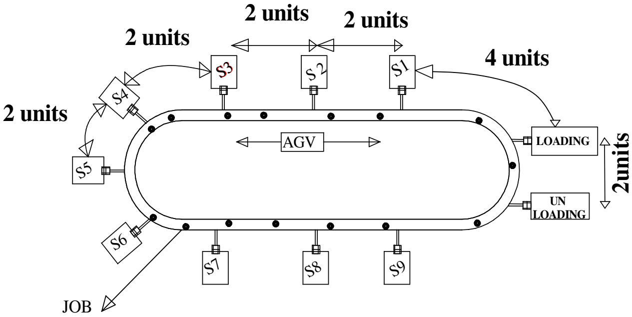

Fig. 3 Loop layout arrangements of FMS for nine machines

(II) Minimize total transportation cost (Z) =

$$\left[\sum_{m_i=1}^{M}\sum_{m_j=1}^{M}\left(MF_{m1m2} {*} MH_{m1m2} {*} RD_{n1n2}\right)+LOC_{mi}+ULOC_{mj}\right]$$

Sub to

$\sum_{m_i=1}^{M} X_{m_i m_j} = 1$; if machine m_{i} is at assigned to slot N=0, otherwise

$\sum_{m_j=1}^{M} X_{m_i m_j} = 1$; if machine m_{j} is at assigned to slot N=0, otherwise

$X_{m_i m_j} \overset{\grave{E}}{o} \{0,1\},\ m_i,\ m_j = 1, 2, \ldots\ldots\ldots\ldots N$

3.2.1 Configuration of loop layout

Figures 1, 2, and 3 shows the loop layout configurations of FMS of six, seven and nine machines respectively.

3.2.2 Procedure for neighborhood creation

The vicinity may be created by any one of the following methods.

- Couple wise interchange of neighboring jobs
- Random exchange of operation sequence with repair function

Out of two methods, we used random exchange of operations sequence with repair function which is considered for neighborhood creation.

3.2.3 Procedure for batch scheduling methodology with repair function

Example: A scheduling comprises 3 machines and 3 batches (each batch contains 100 jobs) and 3 operations are considered. The total there are 9 operations, and the chromosome consists of 9 genes

No. of batches	1			2			3		
No of jobs per batch 100	100			100			100		
No. of Operations	1	2	3	1	2	3	1	2	
Machines	M1	M2	M3	M2	M1	M3	M1	M3	M2
Representation	1	2	3	4	5	6	7	8	9
Unfeasible chromosome	5	2	1	6	3	8	4	7	9
Feasible chromosome	1	4	7	2	3	5	8	6	9

Table 1 Outline of production system

Layout pattern	No. of machines	No. of batches	No of operations	Load/unload stations	No of AGV
Loop	6	6	6	2	1

Batch scheduling methodology Batches are scheduled based on the batch permutation sequence derived by the priority dispatch rule based on precedence constraint. Initially, jobs are carried from the load/unload station to the respective workstations where the operations of all jobs are scheduled as per priority dispatch rule within the precedence relations. Finally, optimum makespan is determined. This type of scheduling methodology helps in reducing the waiting times and thus helps in improving the resource utilization and the throughput. As mentioned in above table, there are two types of chromosomes, one is unfeasible and the other is feasible. There is a procedure to convert unfeasible to feasible chromosomes by repair function.

Repair function A repair function [22] is used to repair the unfeasible chromosome which violate the precedence constraints and convert into feasible one

Step 1: Find the position of the batch operations, which violate the precedence relations.
Step 2: Compute the inter slot distance between the machines of violating operations.
Step 3: If the inter slot distance between them is less than half the chromosome length then swap the operations.
Step 4: Otherwise, randomly pick any one operation and insert it before or after the other depending on the precedence.

4 Proposed methodology

The general explanation of the suggested procedures is shared out as follows.

Table 2 Outline of production system

Layout pattern	No. of machines	No. of batches	No of operations	Load/unload stations	No of AGV
Loop	7	7	7	2	1

Table 3 Outline of production system

Layout pattern	No. of machines	No. of batches	No of operations	Load/unload stations	No of AGV
Loop	9	9	9	2	1

4.1 Genetic algorithm

Genetic algorithms are population-based optimization algorithms centered on the procedure of regular inheritances and expected choice. It is also called as a stochastic search procedure for combinational optimization problems. These search technique is commonly used to generate fruitful solutions to optimization and search problems by using the principle of Charles Darwin of “survival of the fittest,” where weak individuals die before reproducing, while healthier ones breath longer and bear many offspring and breed children, who often inherit the qualities that enabled their parents to survive the reproduced children are in most cases stronger than their parents. The element and mechanism of genetic algorithm are representation, population, evaluation, selection operator, and parameter.

4.1.1 GA parameters

Description	Parameters	Values
Population size	Pop	Multiple
Crossover probability	p_c	0.95
Mutation probability	p_m	0.05
Stopping criteria	T.C	100 generation

4.2 Simulated annealing algorithm

Simulated annealing (SA) is a meta-heuristic for the overall optimization problem of applied mathematics, namely locating a good estimation to the global minimum of a given function in large search space. Simulated annealing was first introduced by Kirkpatrick. Gelett, and Beechi in

Table 4 Batch varieties with batch sizes of the loop layout with six machines with six jobs

Batch number		B1	B2	B3	B4	B5	B6
Batch varieties	CBS	100	100	100	100	100	100
	VBS	50	40	60	30	10	25

Table 5 Batch varieties with batch sizes of the loop layout with seven machines with seven jobs

Batch number		B1	B2	B3	B4	B5	B6	B7
Batch varieties	CBS	100	100	100	100	100	100	100
	VBS	50	40	60	30	10	25	90

1983 and Cerny in 1985 to solve optimization problem. It is based on the comparison between finding an optimal solution in solving optimization problems and finding a low-energy state in the annealing process of solids. Annealing is a physical process for obtaining a low-energy state of a solid in two steps:

1. The metal is heated up to the recrystallization point.
2. The temperature of the metal is reduced slowly by cooling, allowing it to attain thermal balance at each temperature.

The integral part of an annealing algorithm is its neighborhood generation scheme, on the basis of which different annealing algorithms are developed.

4.2.1 SA parameters

Description	Parameter	Values
Population size	Pop	Single
Initial temperature	T	100c
Final temperature	T	Depends on termination criteria
Cooling rate	K	0.95
Probability of acceptance	P	0.95
Neighborhood exchanges	N E	R E O S W R F
Stopping criteria	S.C	200

REOSWRF random exchange of operation sequence with repair function

Table 6 Batch varieties with batch sizes of the loop layout with nine machines with nine jobs

Batch number		B1	B2	B3	B4	B5	B6	B7	B8	B9
Batch varieties	CBS	100	100	100	100	100	100	100	100	100
	VBS	50	40	60	30	10	25	90	15	70

Table 7 Processing time and process routing matrices for configurations of loop layout with seven machines and seven jobs or batches

Batch	O1		O2		O3		O4		O5		O6		O7	
	M	T	M	T	M	T	M	T	M	T	M	T	M	T
B1	2	10	4	12	6	11	5	9	7	7	1	7	3	5
B2	5	4	4	2	7	4	3	6	1	6	6	5	2	3
B3	3	7	5	6	1	4	6	9	7	10	2	4	4	3
B4	2	9	4	2	7	9	6	1	5	9	3	4	1	3
B5	7	4	6	7	5	6	7	2	6	1	10	3	6	
B6	2	9	1	3	6	4	7	3	5	6	4	6	3	6
B7	4	5	2	4	7	3	6	2	5	7	1	7	3	6

4.2.2 SA algorithm

The procedure as follows:

Step 1: Choose an initial point $a^{(0)}$, a stop criterion (S.C). Set *T* a sufficiently high valve, number of iterations to be performed at a particular temperature.
Step 2: Calculate a neighboring point $a^{(t+1)}=N(a^{(t)})$ usually, a random point in the neighborhood is created.
Step 3: If $\Delta HE=HE[a(^{t+1})]-HE(a^{(t)}) <0$,
Step 4: Set $t=t+1$, else create a random number (¥) in the range (0,1). If $r\leq\exp(\Delta HE/T)$, set $t=t+1$; else go to step 2.
Step 5: If $\int a^{(t+1)}-a^{(t)}/ <$(S.C) and *T* is small terminate, else if (t mode n)=0, then lower *T* according to a cooling schedule, else go to step 2.

Table 8 Processing time and process routing matrices for configurations of loop layout with six machines and six jobs

Batch	O_1		O_2		O_3		O_4		O_5		O_6	
	M	T	M	T	M	T	M	T	M	T	M	T
B_1	1	8	2	7	3	14	4	9	5	3	6	4
B_2	2	10	3	17	6	6	4	13	1	4	5	3
B_3	5	18	1	16	4	11	2	12	6	3	3	3
B_4	1	16	6	7	3	11	5	4	4	4	2	13
B_5	4	12	2	15	5	9	6	11	3	3	1	4
B_6	3	8	2	7	6	9	1	6	5	11	4	12

Table 9 Processing time and process routing matrices for configurations of loop layout with nine machines and nine jobs

Batch	O_1		O_2		O_3		O_4		O_5		O_6		O_7		O_8		O_9	
	M	T	M	T	M	T	M	T	M	T	M	T	M	T	M	T	M	T
B_1	2	11	4	10	6	7	5	19	7	8	1	7	9	9	3	10	8	13
B_2	5	4	4	12	9	14	3	6	1	2	6	4	8	3	2	8	7	9
B_3	3	13	5	16	1	4	6	9	7	10	2	4	8	3	4	7	9	2
B_4	2	9	4	12	7	8	6	1	5	9	3	4	9	3	8	9	1	7
B_5	8	14	6	7	5	16	4	7	2	6	9	10	7	4	1	5	3	6
B_6	9	9	8	13	6	4	7	2	5	6	4	6	3	4	1	7	2	3
B_7	4	5	2	14	7	3	9	12	5	17	8	7	3	16	6	5	1	6
B_8	1	6	2	4	6	3	7	2	8	7	5	5	3	6	4	5	9	9
B_9	4	15	1	14	8	6	9	12	5	7	7	2	3	16	6	2	2	6

4.3 Arithmetical illustrations

An attempt is made to apply the GA and SA algorithms on FMS scheduling to determine best solution in terms of minimum completion time and obtaining the batch sequence on each machine which facilitates in arranging machines in optimum manner in loop layout to determine non intersecting arrangement of machine owing to that total transportation cost of making necessary mobility of parts are reduced.

4.3.1 The arithmetical illustration of suggested genetic algorithm for case problem (2) is styled below

1. Choose a feasible chromosomes based on number of operations in case FMS scheduling and based on the number of machines in single row layout of as shown below

   - GA applied to loop layout
   - Chromosomes 1 and 2 are randomly selected

Feasible chromosome 1 2 4 1 5 6 3

total transportation cost$=f(x)=84$ Rs

chromosome 2 4 6 3 1 5 2

total transportation cost$=f(x)=154$ Rs

2. Roulette wheel selection procedure

   - Calculate raw fitness for above chromosomes
   - Develop the mating pool

Feasible chromosome 1 2 4 1 5 6 3

raw fitness$=F(x)=1/(1+f(x))$
$=1/(1+84)=0.0117$

chromosome 2 4 6 3 1 5 2

raw fitness$=F(x)=1/(1+f(x))$
$=1/(1+154)=0.00645$

Finally in mating pool, we got different chromosomes than previous due to reproduction, suppose we got

Table 10 Inter-slot distance matrix for loop layout with six machines

Slots	S_1	S_2	S_3	S_4	S_5	S_6
S_1	0	4	8	10	14	10
S_2	4	0	4	8	10	14
S_3	8	4	0	4	8	10
S_4	10	8	4	0	4	8
S_5	14	10	8	4	0	4
S_6	10	14	10	8	4	0

Table 11 Inter-slot distance matrix for loop layout with seven machines

Slots	S_1	S_2	S_3	S_4	S_5	S_6	S_7
S_1	0	2	4	8	12	12	10
S_2	2	0	2	4	8	12	12
S_3	4	2	0	2	4	8	12
S_4	8	4	2	0	2	4	8
S_5	12	8	4	2	0	2	4
S_6	12	12	8	4	2	0	2
S_7	10	12	12	8	4	2	0

Table 12 Inter-slot distance matrix for loop layout with nine machines

Slots	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0	2	4	6	8	10	12	10	8
S_2	2	0	2	4	6	8	10	12	10
S_3	4	2	0	2	4	6	8	10	12
S_4	6	4	2	0	2	4	6	8	10
S_5	8	6	4	2	0	2	4	6	8
S_6	10	8	6	4	2	0	2	4	6
S_7	12	10	8	6	4	2	0	2	4
S_8	10	12	10	8	6	4	2	0	2
S_9	8	10	12	10	8	6	4	2	0

chromosome 1: **3 5 4 6 1 2**
chromosome 2: **2 4 1 5 6 3**

- *Cross over* Here, we choose single point cross over with two point crossing site given below

chromosome 1: **3 5 4 6 1 2**

(here, condition is machine numbers should not be repeated while crossing)

chromosome 2: **2 4 1 5 6 3**
offspring 1: **3 5 1 4 6 2**
offspring 2: **2 4 5 6 1 3**

- *Mutation* swap mutation is used, which chooses two random positions on offspring chosen and change the machine associated with those position

offspring 1: **3 5 1 4 6 2**
offspring 2: **3 6 1 4 5 2**
(New chromosome)

Likewise, the new chromosome is replaced with older one; go for next generation for evaluating objective function.

4.3.2 The arithmetical illustration of suggested simulated annealing for case problem (10) is styled below

Let TTC=84 Rs

- set HE(1)=84 at T=100^0c
- best solution so for is E(1)=84

At next iteration,

Table 13 Load and unload matrices for loop layout with six machines

Slots	S_1	S_2	S_3	S_4	S_5	S_6
Load station	4	8	12	14	10	6
Unload station	6	10	14	12	8	4

Table 14 Load and unload matrices for loop layout with seven machines

Slots	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Load station	4	6	8	12	10	8	6
Unload station	6	8	10	12	8	6	4

- If neighboring valve (TTC)=85 Rs
- set HE(2)=85 at T=100^0c
- calculate the difference b/n two energy levels, i.e.,
- H HE(2)-HE(1)
- Now H f H Then Calculate the probability of acceptance (R)=exp(-ΔHE/T)
- If R<exp(-ΔHE/T), Then reject the current solution and do not change the best solution

5 Data set details for loop layout with FBSP

The combination of the scheduling of parts into a flexible manufacturing system layout, succeeded by the automated material handling and by means of computer control, can effect in manufacturing systems described by flexibility, great productivity, and little cost per unit formed. The response to the FMS schedule is a best routing of parts acquired from the production schedule that let off the transfer activities of the FMS. Here, the transfer activity is included. The transportation cost depends on the inter slot distance between machines, incidence of trips of parts from machine to machine and loading/unloading station, and material handling cost. The AGV moves in forward and reverse direction, i.e., loading station-machines-unloading station vice versa.

A production system with the summary and batch sizes and the layout of FMS are shown in Tables 1, 2, and 3. The data set details of batch varieties and sizes are given in Tables 4, 5, and 6. Let there be parts to be processed on machine for various operations which requires the processing time and part routing with the operation sequence of parts which steers the parts on various machines are depicted in Tables 7, 8, and 9.

Table 15 Load and unload matrices for loop layout with nine machines

Slots	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
Load station	4	6	8	10	12	12	10	8	6
Unload station	6	8	10	12	12	10	8	6	4

Transportation cost per unit distance=1 Rs
Load and unload cost per unit distance=1 Rs

Table 16 Comparison of arithmetical results of the proposed evolutionary algorithms (for CBS=100 numbers in a batch and same quantity in all batches with number of iterations=100)

Instan-M/c×J/B×Oper	GA			SA		
	MAKSP (min)	TTC (Rs)	CPU (s)	MAKSP (min)	TTC (Rs)	CPU (s)
KMN 1-(6×5×5)	7100	9400	14	7400	10,000	0
KMN 2-(6×6×6)	7400	13,000	14	8100	13,200	0
KMN 3-(7×7×7)	6900	9600	16	7300	9800	0
KMN 4-(7×6×6)	6300	8400	14	6700	8800	0
KMN 5-(7×7×4)	4300	9600	14	4300	11,400	0
KMN 6-(7×7×5)	7500	9200	31	7500	9600	0
KMN 7-(7×7×6)	7300	10,600	14	8200	11,000	0
KMN 8-(9×5×5)	5600	8000	13	5800	8200	0
KMN 9-(9×6×5)	6200	9800	14	6300	10,200	0
KMN 10-(9×7×5)	5100	9600	14	5100	12,000	0
KMN 11-(9×7×6)	6200	10,600	15	6500	12,200	0
KMN 12-(9×7×7)	8300	11,000	15	8800	12,000	0
KMN 13-(9×8×5)	7300	12,600	14	7200	12,800	0
KMN 14-(9×8×6)	9700	13,800	15	9800	14,200	0
KMN 15-(9×8×7)	10,200	13,400	15	10,500	13,800	0
KMN 16-(9×8×8)	9500	13,800	16	9700	13,000	0
KMN 17-(9×9×5)	6600	13,400	15	6600	15,000	0
KMN 18-(9×9×6)	6000	15,200	15	6200	14,600	0
KMN 19-(9×9×7)	8800	15,600	16	9000	15,600	0
KMN 20-(9×9×8)	8400	15,000	17	8900	15,400	0
KMN 21-(9×9×8)	11,200	15,600	18	11,200	16,400	0

6 Data set details of batch varieties and sizes

(CBS = constant batch size, VBS = variable batch size)

The way of part/batch moves over the machines is given in the same Tables 7, 8, and 9 as an input for FMS scheduling, where the objective is to arrive at a layout, which determines non-intersecting best arrangement of machines such that total transportation cost of making necessary mobility of parts are reduced.

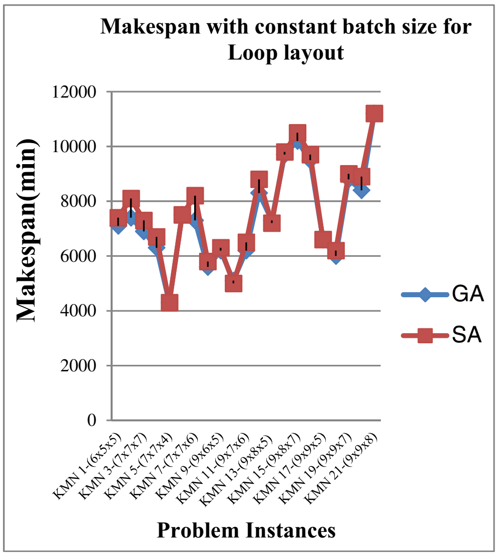

Fig. 4 Comparison of proposed algorithm for makespan with constant batch size for loop layout

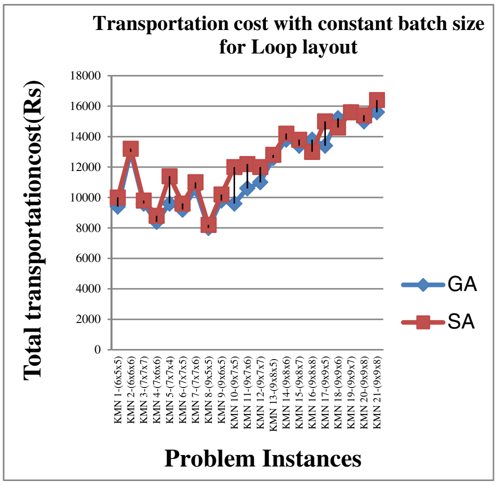

Fig. 5 Comparison of proposed algorithm for makespan with constant batch size for loop layout

Table 17 Comparison of arithmetical results of the proposed evolutionary algorithms for CBS=100 numbers in a batch and same quantity in all batches

Instance (M×J/B×O)	GA			SA		
	BWT (min)	MASEQ	MWT (min)	BWT (min)	MASEQ	MWT (min)
KMN 1 (6×5×5)	B_1: 3000 B_2: 2100 B_3: 1100 B_4: 2900 B_5: 2100	6, 2, 5, 4, 3, 1	M_1: 3300 M_2: 2800 M_3: 800 M_4: 1500 M_5: 5500 M_6: 4400	B1: 3300 B2: 2400 B3: 1400 B4: 3200 B5: 2400	6, 2, 5, 4, 3, 1	M1: 3600 M2: 3100 M3: 1100 M4: 1800 M5: 5800 M6: 4700
KMN 2 (6×6×6)	B1: 2900 B2: 2100 B3: 1200 B4: 1900 B5: 2000 B6: 2100	5, 6, 4, 3, 1, 2	M1: 2000 M2: 1000 M3: 1900 M4: 1300 M5: 2600 M6: 3400	B1: 3600 B2: 2800 B3: 1900 B4: 2600 B5: 2700 B6: 2800	1, 2, 3, 4, 5, 6	M1: 2700 M2: 1700 M3: 2600 M4: 2000 M5: 3300 M6: 4100
KMN 3 (7×7×7)	B1: 800 B2: 3900 B3: 2600 B4: 3200 B5: 2300 B6: 3200 B7: 3500	7,5,2,1,4,6,3	M1: 2900 M2: 2400 M3: 2900 M4: 3200 M5: 2200 M6: 3000 M7: 2900	B1: 1200 B2: 4300 B3: 3000 B4: 3600 B5: 2700 B6: 3600 B7: 3900	1, 4, 6, 7, 5, 2, 3	M1: 3300 M2: 2800 M3: 3300 M4: 3600 M5: 2600 M6: 3400 M7: 3300
KMN 4 (7×6×6)	B1: 1800 B2: 800 B3: 2700 B4: 2800 B5: 1200 B6: 2700	7,4,6,2, 3, 1, 5	M1: 1500 M2: 3900 M3: 3500 M4: 3400 M5: 1700 M6: 1800 M7: 2500	B1: 2200 B2: 1200 B3: 3100 B4: 3200 B5: 1600 B6: 3100	7, 4, 5, 1, 3, 6, 2	M1: 1900 M2: 4300 M3: 3900 M4: 3800 M5: 2100 M6: 2200 M7: 2900
KMN 5 (7×7×4)	B1: 200 B2: 2700 B3: 1800 B4: 1600 B5: 2600 B6: 2400 B7: 2300	4,6,5,2, 3, 1, 7	M1: 3600 M2: 1500 M3: 1700 M4: 400 M5: 1700 M6: 2900 M7: 1800	B1: 200 B2: 2700 B3: 1800 B4: 1600 B5: 2600 B6: 2400 B7: 2300	4, 6, 5, 2, 3, 1, 7	M1: 3600 M2: 1500 M3: 1700 M4: 400 M5: 1700 M6: 2900 M7: 1800
KMN 6 (7×7×5)	B1: 3600 B2: 3300 B3: 3900 B4: 2600 B5: 4500 B6: 3300 B7: 4600	3,6,2, 5, 4, 1, 7	M1: 5800 M2: 3200 M3: 5500 M4: 6000 M5: 200 M6: 1800 M7: 3300	B1: 3600 B2: 3300 B3: 3900 B4: 2600 B5: 4500 B6: 3300 B7: 4600	7, 1, 2, 6, 3, 4, 5	M1: 5800 M2: 3200 M3: 5500 M4: 6000 M5: 200 M6: 1800 M7: 3300
KMN 7 (7×7×6)	B1: 1700 B2: 2800 B3: 3100 B4: 4500 B5: 600 B6: 4900 B7: 1500	3,4,2, 6, 1, 5, 7	M1: 3600 M2: 4100 M3: 1400 M4: 2400 M5: 3400 M6: 2900 M7: 1300	B1: 2600 B2: 3700 B3: 4000 B4: 5400 B5: 1500 B6: 5800 B7: 2400	7, 5, 3, 4, 2, 6, 1	M1: 4500 M2: 5000 M3: 2300 M4: 3300 M5: 4300 M6: 3800 M7: 2200
KMN 8 (9×5×5)	B1: 1500 B2: 2500 B3: 0 B4: 100 B5: 600	1, 6, 7, 3, 5, 4, 2, 8, 9	M1: 3100 M2: 3400 M3: 3400 M4: 2300 M5: 3500 M6: 1700 M7: 2300 M8: 4100 M9: 3300	B1: 1700 B2: 2700 B3: 200 B4: 300 B5: 800	9, 1, 5, 4, 2, 3, 8, 7, 6	M1: 3300 M2: 3600 M3: 3600 M4: 2500 M5: 3700 M6: 1900 M7: 2500 M8: 4300 M9: 3500
KMN 9	B1: 2600	1, 4, 2, 6, 8, 9, 5, 3, 7	M1: 3000	B1: 2500	2, 7, 5, 1, 9, 8, 3, 6, 4	M1: 3700

Table 17 (continued)

Instance (M×J/B×O)	GA			SA		
	BWT (min)	MASEQ	MWT (min)	BWT (min)	MASEQ	MWT (min)
(9×6×5)	B2: 2100 B3: 300 B4: 1000 B5: 2100 B6: 1500		M2: 2900 M3: 4300 M4: 2800 M5: 1800 M6: 2300 M7: 3300 M8: 4700 M9: 3100	B2: 1100 B3: 1000 B4: 2100 B5: 400 B6: 2000		M2: 1100 M3: 2400 M4: 2100 M5: 3100 M6: 3700 M7: 2100 M8: 3200 M9: 3300
KMN10 (9×7×5)	B1: 1200 B2: 1700 B3: 2400 B4: 2200 B5: 1400 B6: 2000 B7: 2600	2, 4, 1, 5, 8, 7, 3, 6, 9	M1: 3400 M2: 2000 M3: 3000 M4: 2600 M5: 3800 M6: 3200 M7: 900 M8: 3900 M9: 900	B1: 1100 B2: 1600 B3: 2300 B4: 2100 B5: 1300 B6: 1900 B7: 2500	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 3300 M2: 1900 M3: 2900 M4: 2500 M5: 3700 M6: 3100 M7: 800 M8: 3800 M9: 800
KMN 11 (9×7×6)	B1: 2500 B2: 1200 B3: 2700 B4: 700 B5: 2600 B6: 2400 B7: 1100	1, 3, 4, 7, 6, 5, 9, 2, 8	M1: 4200 M2: 1300 M3: 200 M4: 2000 M5: 2700 M6: 3000 M7: 5100 M8: 4200 M9: 2900	B1: 2800 B2: 1500 B3: 3000 B4: 1000 B5: 2900 B6: 2700 B7: 1400	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 4500 M2: 1600 M3: 500 M4: 2300 M5: 3000 M6: 3300 M7: 5400 M8: 4500 M9: 3200
KMN 12 (9×7×7)	B1: 400 B2: 2200 B3: 1300 B4: 2200 B5: 2500 B6: 4000 B7: 3600	1, 9, 5, 8, 2, 4, 3, 7, 6	M1: 4600 M2: 4600 M3: 4400 M4: 2500 M5: 2300 M6: 1000 M7: 3600 M8: 6800 M9: 3000	B1: 900 B2: 2700 B3: 1800 B4: 2700 B5: 3000 B6: 4500 B7: 4100	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 5100 M2: 5100 M3: 4900 M4: 3000 M5: 2800 M6: 1500 M7: 4100 M8: 7300 M9: 3500
KMN 13 (9×8×5)	B1: 1100 B2: 2700 B3: 2500 B4: 2900 B5: 3400 B6: 1400 B7: 2900 B8: 3600	8, 2, 4, 3, 7, 6, 5, 1, 9	M1: 3100 M2: 4600 M3: 4800 M4: 1000 M5: 3000 M6: 3500 M7: 3000 M8: 4400 M9: 400	B1: 1000 B2: 2600 B3: 2400 B4: 2800 B5: 3300 B6: 1300 B7: 2800 B8: 3500	5, 1, 6, 7, 9, 8, 2, 4, 3	M1: 3000 M2: 4500 M3: 4700 M4: 900 M5: 2900 M6: 3400 M7: 2900 M8: 4300 M9: 300
KMN 14 (9×8×6)	B1: 3700 B2: 3500 B3: 4300 B4: 3000 B5: 3000 B6: 1900 B7: 2300 B8: 2800	9, 1, 7, 6, 8, 3, 2, 4, 5	M1: 6400 M2: 3300 M3: 2300 M4: 3900 M5: 5700 M6: 2100 M7: 3900 M8: 5300 M9: 1300	B1: 3800 B2: 3600 B3: 4400 B4: 3100 B5: 3100 B6: 2000 B7: 2400 B8: 2900	1, 4, 2, 7, 6, 5, 3, 8, 9	M1: 6500 M2: 3400 M3: 2400 M4: 4000 M5: 5800 M6: 2200 M7: 4000 M8: 5400 M9: 1400
KMN 15 (9×8×7)	B1: 2200 B2: 3200 B3: 4400 B4: 2600 B5: 1700	2, 3, 9, 6, 8, 5, 1, 4, 7	M1: 5300 M2: 700 M3: 2300 M4: 2500 M5: 3800	B1: 2500 B2: 3500 B3: 4700 B4: 2900 B5: 2000	9, 6, 8, 5, 1, 4, 7, 3, 2	M1: 5600 M2: 1000 M3: 2600 M4: 2800 M5: 4100

Table 17 (continued)

Instance (M×J/B×O)	GA			SA		
	BWT (min)	MASEQ	MWT (min)	BWT (min)	MASEQ	MWT (min)
	B6: 4300 B7: 2000 B8: 3400		M6: 5300 M7: 6200 M8: 5400 M9: 2500	B6: 4600 B7: 2300 B8: 3700		M6: 5600 M7: 6500 M8: 5700 M9: 2800
KMN 16 (9×8×8)	B1: 400 B2: 3900 B3: 4700 B4: 3000 B5: 4000 B6: 3800 B7: 5400 B8: 2700	5, 8, 7, 9, 6, 4, 1, 2, 3	M1: 4700 M2: 3700 M3: 3700 M4: 4600 M5: 2900 M6: 3500 M7: 5700 M8: 6400 M9: 2200	B1: 600 B2: 4100 B3: 4900 B4: 3200 B5: 4200 B6: 4000 B7: 5600 B8: 2900	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 4900 M2: 3900 M3: 3900 M4: 4800 M5: 3100 M6: 3700 M7: 5900 M8: 6600 M9: 2400
KMN 17 (9×9×5)	B1: 1300 B2: 1900 B3: 3000 B4: 2400 B5: 3900 B6: 3300 B7: 2900 B8: 3700 B9: 2600	4, 1, 5, 2, 9, 3, 8, 6, 7	M1: 3900 M2: 500 M3: 4000 M4: 2300 M5: 4900 M6: 3300 M7: 1600 M8: 1500 M9: 3000	B1: 1300 B2: 1900 B3: 3000 B4: 2400 B5: 3900 B6: 3300 B7: 2900 B8: 3700 B9: 2600	4, 1, 5, 2, 6, 9, 3, 8, 7	M1: 3900 M2: 500 M3: 4000 M4: 2300 M5: 4900 M6: 3300 M7: 1600 M8: 1500 M9: 3000
KMN 18 (9×9×6)	B1: 0000 B2: 2800 B3: 2000 B4: 2800 B5: 1800 B6: 3200 B7: 2200 B8: 3000 B9: 1500	6, 7, 8, 9, 1, 2, 3, 4, 5	M1: 3000 M2: 1800 M3: 2000 M4: 1700 M5: 2000 M6: 2500 M7: 3300 M8: 1400 M9: 1600	B1: 200 B2: 3000 B3: 2200 B4: 3000 B5: 2000 B6: 3400 B7: 2400 B8: 3200 B9: 1700	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 3200 M2: 2000 M3: 2200 M4: 1900 M5: 2200 M6: 2700 M7: 3500 M8: 1600 M9: 1800
KMN 19 (9×9×7)	B1: 4400 B2: 3600 B3: 3300 B4: 3500 B5: 3800 B6: 1600 B7: 2200 B8: 4000 B9: 4600	7, 9, 6, 8, 5, 1, 2, 4, 3	M1: 5000 M2: 1800 M3: 2700 M4: 3300 M5: 2000 M6: 3100 M7: 2800 M8: 4300 M9: 6000	B1: 4600 B2: 3800 B3: 3500 B4: 3700 B5: 4000 B6: 1800 B7: 2400 B8: 4200 B9: 4800	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 5200 M2: 2000 M3: 2900 M4: 3500 M5: 2200 M6: 3300 M7: 3000 M8: 4500 M9: 6200
KMN 20 (9×9×8)	B1: 0 B2: 4800 B3: 3600 B4: 4200 B5: 2900 B6: 4900 B7: 2800 B8: 3700 B9: 5000	1, 2, 3, 5, 4, 6, 9, 8, 7	M1: 3600 M2: 1800 M3: 4800 M4: 4500 M5: 1900 M6: 3300 M7: 3200 M8: 4900 M9: 3900	B1: 500 B2: 5300 B3: 4100 B4: 4700 B5: 3400 B6: 5400 B7: 3300 B8: 4200 B9: 5500	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 4100 M2: 2300 M3: 5300 M4: 5000 M5: 2400 M6: 3800 M7: 3700 M8: 5400 M9: 4400
KMN 21 (9×9×9)	B1: 1800 B2: 5000 B3: 4400 B4: 5000 B5: 3700 B6: 5800 B7: 2700 B8: 6500 B9: 3200	7, 6, 9, 8, 5, 2, 3, 1, 4	M1: 5400 M2: 4700 M3: 3100 M4: 3300 M5: 1300 M6: 7000 M7: 6400 M8: 3700 M9: 3200	B1: 1800 B2: 5000 B3: 4400 B4: 5000 B5: 3700 B6: 5800 B7: 2700 B8: 6500 B9: 3200	1 2 3 4 5 6 7 8 9	M1: 5400 M2: 4700 M3: 3100 M4: 3300 M5: 1300 M6: 7000 M7: 6400 M8: 3700 M9: 3200

7 Data set details of processing time of parts and processing sequence of machines

The inter slot between machines, i.e., the gap between machine measures in units are given in Tables 10, 11, and 12. The loading/unloading distance matrix specifies distance from machines to load/unload station are shown in Tables 13, 14, and 15, unit material handling cost per unit, i.e., the carrying cost of parts between machines is unit cost. With the collected information from various literature, it is applied that Tables 1, 2, and 3 dataset details are considered as integrated data for both layout design and FMS scheduling where scheduling is constraint for layout design. Tables 4, 5, 6, 7, 8, and 9 dataset details are concern to FMS scheduling which act as important parameters for optimum allocation of jobs with predefined processing time and routing for generating minimum makespan. Tables 10, 11, 12, 13, 14, and 15 dataset details are used for loop layout design such as interslot distance tables shows that the predefined clearance between slots over which machines are assigned by means of permutation rule. Further load and unload matrix are calculated based on the clearance between machines and direction of parts. Also, the reason for integrating the loop layout design with FMS scheduling is the data set details of Tables 7, 8, and 9 are used for calculating the frequency of trips between machines as one of the key input parameter for loop layout design which is not mentioned in dataset details because it is developed in simulation code. Though the input data from Tables 1, 2, and 3 as well as Tables 10 11, 12, 13, 14, and 15 is entered manually in IDE tool in which simulation code is executed but data of frequency of trip between machines is calculated by code itself and taken as additional input for loop layout design. The necessary pseudo code for calculation of frequency of trips is shown below.

Pseudo code for frequency of trips permutation for loop layout design

```
BEGIN void Genetic::FromTochatObjfunction(int RR[MAX_MC][MAX_BAT])
  {
  for (i=1; i≤no_machines; i++){
  for (j=1; j≤no_machines; j++)
  { FromTochat[i][j]=0;
  }
  }
  for (i=1; i≤no_batchs; i++) {
  for (j=1; j<no_operations; j++)
  {
  if (RR[i][j] !=RR[i][j+1])
  FromTochat[RR[i][j]][RR[i][j+1]]+=(Bsizes[i]);
  }
  } for (i=1; i≤no_machines; i++){
  for (j=1; j≤no_machines; j++){
  if (FromTochat[i][j]<FromTochat[j][i])
  FromTochat[i][j]=FromTochat[j][i];
  else
  FromTochat[j][i]=FromTochat[i][j];
  } END
```

8 Data set details of inter-slot distance between machines

Tables 10, 11, and 12 shows details of inter-slot distance between machines of FSM for six, seven and nine machines.

9 Data set details of load, unload matrices for loop layout

Tables 13, 14, and 15 shows details of load and unload matrices for loop layout with six, seven and nine machines respectively.

10 Results and discussions

In the present work, the optimal solution for loop layout with integrated scheduling uses non-traditional optimization techniques such GA and SA. So far, only non-traditional methods are used for solving such kind of problems. In traditional methods to calculate minimum total transportation cost, it is necessary to check (n!), for example :(6!)(1.393140695 * 10^{17}) sequences in order to find the optimal sequences. The major advantage of using non-traditional algorithms is that even though the number of possible sequences is very high, an optimal solution can be obtained within a fraction of seconds while compiling on a standard PC. These algorithms are verified through computer simulation for various physical life problems area found to be very operative.

One anxiety provoked by the investigators in any research is to compare their approaches with those of other researchers. If the standard usual test problems are open, the performances of different algorithms can be compared on closely the same set of test problems. For this reason, we chose 21 benchmark problems from Kumanan et al. [18] (KMN) as the test problems for this study. These benchmark problems are categorized into two groups, i.e., constant batch size (CBS) problems and variable batch size (VBS) problems. Kumanan has produced a set of problems with seven and nine machines with two and four jobs. There are 2 instances for (nxm=6×6) problem combination and 5 instances for (nxm=7×7)) and 14 instances for (nxm=9×9). Totally, there are 42 problem instances.

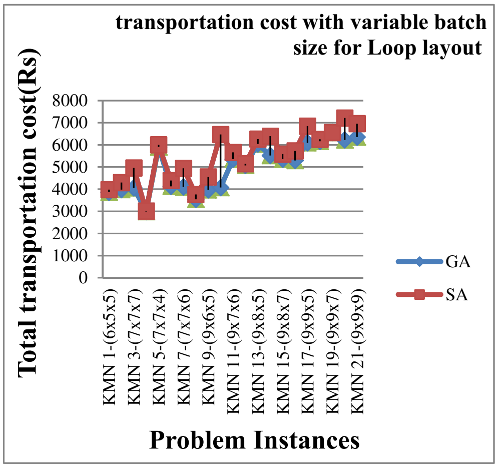

Fig. 6 Comparison of proposed algorithm for transportation cost with variable batch size for loop layout

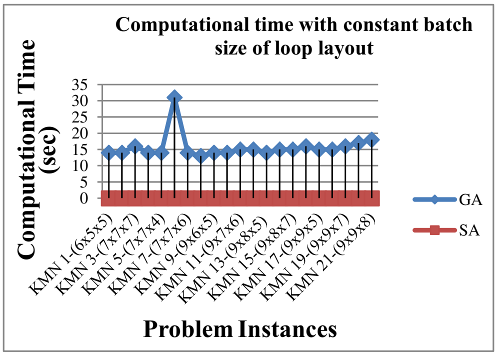

Fig. 8 Comparison of proposed algorithm for computational time with constant batch size of loop layout

Table 16 shows the results of test problems for CBS from KMN1 to KMN21 and is understood that the test problems are solved through the proposed algorithm, and the results are compared and found that performance of GA and SA for calculating total transportation cost (TTC) and make span (MAKSP) is varying as per the problem size. By relative analysis, we observed that solutions are optimized for GA and found that GA can afford the best solution when compared with SA to all test problems. Furthermore, the computational time of GA fluctuates as the problem size varies but the computational time of SA is zero for all problems. Comparison of make span and total transportation cost for CBS by the proposed evolutionary algorithms for different problem sizes is depicted in Figs. 4 and 5. The plot shown in Figs. 4 and 5 is styled for instance KMN 1–KMN 21. It is observed that there are moderate variations in results of TTC and MAKSP against problem instances shown in the plot for GA and SA. It is found that TTC and MAKSP are low at small size problems and reaches to high value as problem size increases. Furthermore, GA curve fluctuates at lower values than SA curve.

Table 17 shows the results of test problems for CBS from KMN 1 to KMN 21 and is figured out that the test problems are solved through the proposed algorithm, and the results are compared and found that performance of GA and SA for calculating batch waiting time (BWT) and machine waiting time (MWT) obtained for corresponding problem instances is varying as per the problem size and based on MAKSP value. By relative analysis, we observed that GA shows minimum waiting times when compared with SA to all test problems (Figs. 6, 7, 8, and 9). Comparison of BWT and MWT for CBS by the proposed evolutionary algorithms is depicted in Figs. 10 and 11. The plot shown in Figs. 10 and 11 is styled for instance which has seven batches/jobs. It is observed that BWT and MWT for constant batch size are less for GA when compared with SA.

Table 18 shows the results of test problems for VBS from KMN1 to KMN21 and is understood that the test problems are solved through the proposed algorithm, and the results are compared and found that performance of GA and SA for calculating TTC and MAKSP is varying as per the problem size. By relative analysis, we observed that solutions are optimized for GA and found that

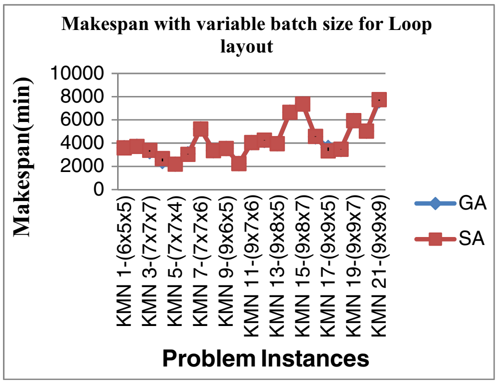

Fig. 7 Comparison of proposed algorithm for makespan with variable batch size for loop layout

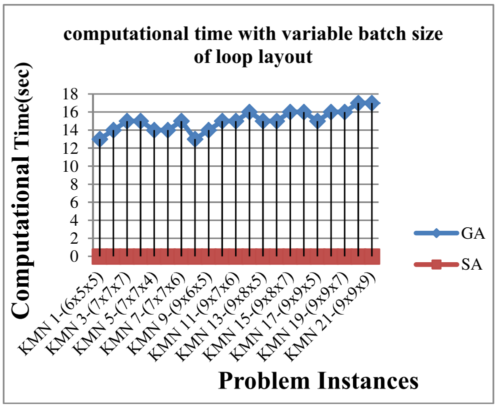

Fig. 9 Comparison of proposed algorithm for computational time with variable batch size of loop layout

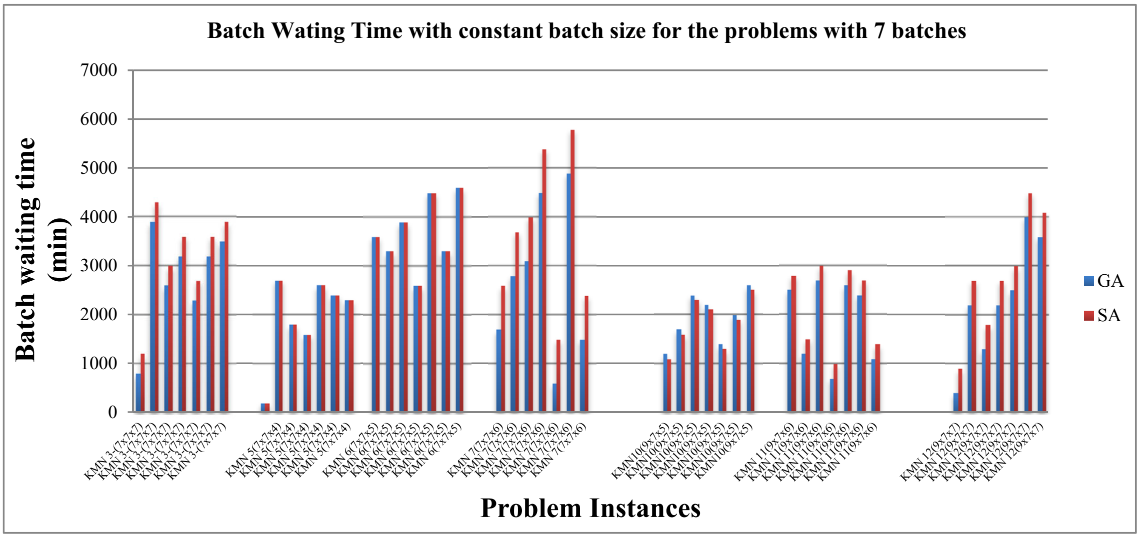

Fig. 10 Comparison of proposed algorithm for batch waiting time with constant batch size for the problems with seven batches

GA affords best solution when compared with SA to all test problems. Furthermore, the computational time of genetic algorithm fluctuates as the problem size varies, but the CPU time of simulated annealing is zero for all problems. Comparison of make span and total transportation cost for VBS by the proposed evolutionary algorithms for different problem sizes is depicted in Figs. 6 and 7. The plot shown in Figs. 6 and 7 is styled for instance KMN 1–KMN 21. It is observed that there are moderate variations in results of TTC and MAKSP against problem instances shown in the plot for GA and SA. It is found that TTC and MAKSP are low at small size problems and reaches to high value as problem size increases and also in Fig. 7. MAKSP variations are almost closer for both GA and SA. Furthermore, GA curve fluctuates at lower values than SA.

Table 19 shows the results of test problems for VBS from KMN 1–KMN 21 and is understood the test problems are solved through the proposed algorithm, and the results are compared and found that performance of GA

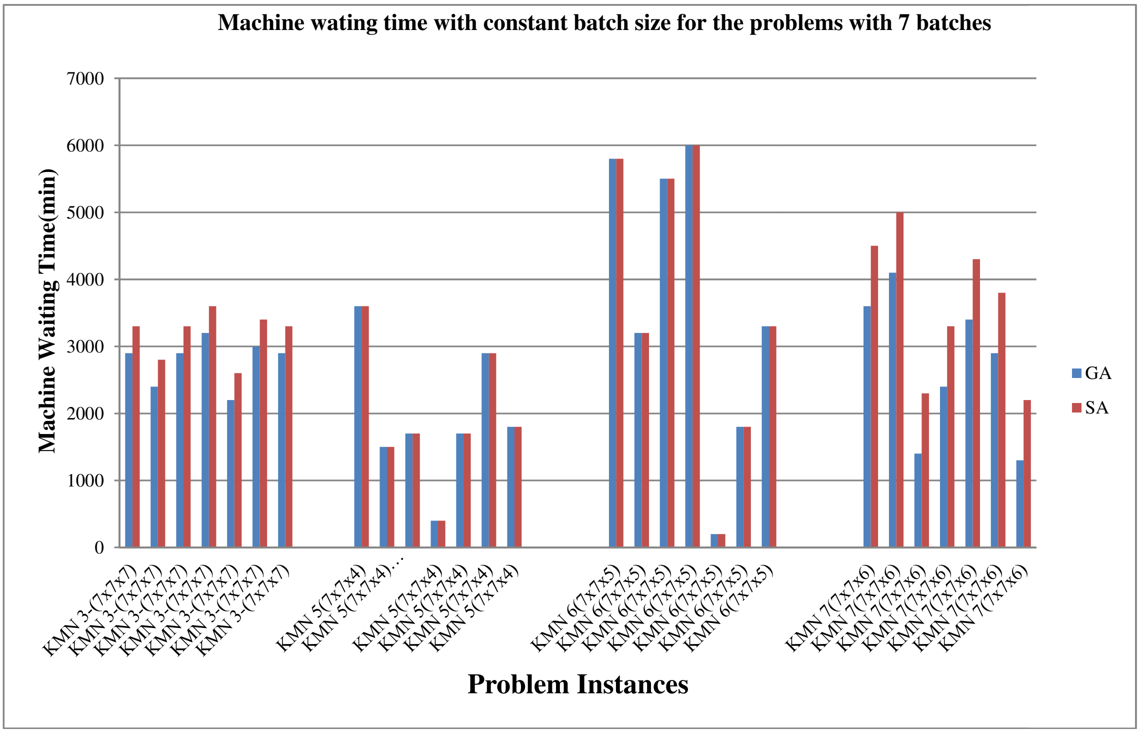

Fig. 11 Comparison of proposed algorithm for machine waiting time with constant batch size for the problems with seven batches

Table 18 Comparison of arithmetical results of the proposed evolutionary algorithms (for VBS with number of iterations=100)

Instan-M/c×J/B×Oper	GA			SA		
	MAKSP (min)	TTC (Rs)	CPU (s)	MAKSP (min)	TTC (Rs)	CPU (s)
KMN 1-(6×5×5)	3600	3860	13	3600	3980	0
KMN 2-(6×6×6)	3720	3980	14	3720	4300	0
KMN 3-(7×7×7)	3275	4060	15	3390	4960	0
KMN 4-(7×6×6)	2430	2990	15	2670	3010	0
KMN 5-(7×7×4)	2190	5900	14	2190	6000	0
KMN 6-(7×7×5)	3080	4130	14	3040	4390	0
KMN 7-(7×7×6)	5220	4100	15	5220	4940	0
KMN 8-(9×5×5)	3360	3540	13	3360	3760	0
KMN 9-(9×6×5)	3540	3970	14	3540	4550	0
KMN 10-(9×7×5)	2250	4070	15	2250	6470	0
KMN 11-(9×7×6)	4050	5350	15	4050	5660	0
KMN 12-(9×7×7)	4230	5070	16	4270	5150	0
KMN 13-(9×8×5)	3960	6040	15	3960	6260	0
KMN 14-(9x8×6)	6660	5520	15	6660	6390	0
KMN 15-(9×8×7)	7380	5350	16	7380	5580	0
KMN 16-(9×8×8)	4550	5290	16	4585	5730	0
KMN 17-(9×9×5)	3625	6090	15	3335	6850	0
KMN 18-(9×9×6)	3460	6160	16	3480	6250	0
KMN 19-(9×9×7)	5940	6560	16	5940	6560	0
KMN 20-(9×9×8)	5040	6220	17	5040	7210	0
KMN 21-(9×9×9)	7650	6350	17	7750	6960	0

and SA for calculating BWT and MWT obtained for corresponding problem instances is varying as per the problem size and based on MAKSP value (i.e., if make span is the same for both algorithm, then waiting times will also be the same, vice-versa). By relative analysis, we observed that GA shows minimum waiting times when compared with SA to all test problems. Furthermore, the required machine sequences (MASEQ) are depicted in the same table.

Comparison of computational time (seconds) by the proposed evolutionary algorithm for CBS and for VBS are shown in Figs. 8 and 9. It is observed that GA requires a fraction of a second for computing given input to optimum solution, but SA gives the results with zero time because it is a single solution method which is well known as trajectory-based heuristic, whereas GA is population-based heuristic which has many solutions (chromosomes) in a mating pool which can be reproduced as new offspring's (new solutions). Actually, GA is an effective algorithm in searching local optima when compared with SA. Furthermore, a necessary simulation code is generated and the code is run by the IDE tool in which C++ compiler used as plug in. This tool has eclipse-based features which afford the competency to figure, correct, steer, and sort out the tasks that use C++ as a programming language using Intel core i3-380 M processor. Furthermore, it is more convenient to user to print and display the results and errors if any in execution.

11 Conclusion

This paper conveys the modeling of loop layout design with integrated scheduling in which the frequency of trips between machines, the clearance between the machines with loading and unloading distance from loading/unloading station to all machines, and unit material handling cost (MHD) are estimated differently. The problem is framed as the quadratic assignment problems (QAP) formulation of facility layout problem. This is owing to the point that in the QAP models, the distance between the positions of slots is identified well in advance but it is order-dependent for the instances considered in this paper.

From the results, we conclude that loop layout is optimized using GA and is better than SA with constant MHD cost and frequency of trips between machines. The parameter like

Table 19 Comparison of arithmetical results of the proposed evolutionary algorithms for VBS

Instance (M×J/B×O)	GA			SA		
	BWT (min)	MASEQ	MWT (min)	BWT (min)	MASEQ	MWT (min)
KMN 1 (6×5×5)	B1: 1550 B2: 1600 B3: 1250 B4: 2340 B5: 3100	1, 3, 4, 5, 2, 6	M1: 2160 M2: 2370 M3: 1020 M4: 1340 M5: 2910 M6: 2390	B1: 1550 B2: 1600 B3: 1250 B4: 2340 B5: 3100	1, 3, 4, 5, 2, 6	M1: 2160 M2: 2370 M3: 1020 M4: 1340 M5: 2910 M6: 2390
KMN 2 (6×6×6)	B1: 1470 B2: 1600 B3: 1000 B4: 2070 B5: 3180 B6: 2395	6, 5, 2, 1, 3, 4	M1: 1530 M2: 1535 M3: 1660 M4: 1550 M5: 1885 M6: 2555	B1: 1470 B2: 1600 B3: 1000 B4: 2070 B5: 3180 B6: 2395	4, 3, 5, 2, 1, 6	M1: 1530 M2: 1535 M3: 1660 M4: 1550 M5: 1885 M6: 2555
KMN 3 (7×7×7)	B1: 225 B2: 2075 B3: 695 B4: 2165 B5: 2815 B6: 2350 B7: 215	7, 5, 2, 1, 4, 3, 6	M1: 1550 M2: 1500 M3: 1495 M4: 1685 M5: 1195 M6: 1605 M7: 1510	B1: 340 B2: 2190 B3: 810 B4: 2280 B5: 2930 B6: 2465 B7: 330	7, 6, 3, 4, 5, 2, 1	M1: 1665 M2: 1615 M3: 1610 M4: 1800 M5: 1310 M6: 1720 M7: 1625
KMN 4 (7×6×6)	B1: 180 B2: 230 B3: 270 B4: 1380 B5: 1920 B6: 1530	4, 7, 5, 1, 3, 6, 2	M1: 805 M2: 1730 M3: 1305 M4: 1375 M5: 940 M6: 835 M7: 950	B1: 420 B2: 470 B3: 510 B4: 1620 B5: 2160 B6: 1770	4, 7, 6, 2, 3, 5, 1	M1: 1045 M2: 1970 M3: 1545 M4: 1615 M5: 1180 M6: 1075 M7: 1190
KMN 5 (7×7×4)	B1: 140 B2: 1550 B3: 690 B4: 1380 B5: 2020 B6: 1715 B7: 390	4, 6, 7, 1, 3, 2, 5	M1: 1875 M2: 975 M3: 960 M4: 210 M5: 1040 M6: 1790 M7: 1035	B1: 140 B2: 1550 B3: 690 B4: 1380 B5: 2020 B6: 1715 B7: 390	4, 3, 1, 7, 6, 2, 5	M1: 1875 M2: 975 M3: 960 M4: 210 M5: 1040 M6: 1790 M7: 1035
KMN 6 (7×7×5)	B1: 1680 B2: 1950 B3: 1470 B4: 2160 B5: 3330 B6: 2580 B7: 1020	5, 2, 6, 3, 4, 1, 7	M1: 3065 M2: 1780 M3: 2900 M4: 2780 M5: 880 M6: 1020 M7: 1765	B1: 1090 B2: 1360 B3: 880 B4: 1570 B5: 2740 B6: 1990 B7: 430	6, 2, 3, 1, 7, 4, 5	M1: 2475 M2: 1190 M3: 2310 M4: 2190 M5: 290 M6: 430 M7: 1175
KMN 7 (7×7×6)	B1: 2420 B2: 3420 B3: 2700 B4: 4380 B5: 4550 B6: 4620 B7:3550	3, 4, 2, 6, 1, 5, 7	M1: 3580 M2: 4065 M3: 2950 M4: 2760 M5: 3745 M6: 2490 M7: 2500	B1: 2420 B2: 3420 B3: 2700 B4: 4380 B5: 4550 B6: 4620 B7: 3550	4, 5, 7, 6, 1, 2, 3	M1: 3580 M2: 4065 M3: 2950 M4: 2760 M5: 3745 M6: 2490 M7: 2500
KMN 8 (9×5×5)	B1: 1310 B2: 2120 B3: 1400 B4: 1710 B5: 2860	7, 3, 5, 8, 4, 2, 9, 1, 6	M1: 2240 M2: 2640 M3: 2430 M4: 2450 M5: 2410 M6: 2260 M7: 1790 M8: 2910 M9: 2310	B1: 1310 B2: 2120 B3: 1400 B4: 1710 B5: 2860	8, 9, 1, 6, 5, 4, 2, 3, 7	M1: 2240 M2: 2640 M3: 2430 M4: 2450 M5: 2410 M6: 2260 M7: 1790 M8: 2910 M9: 2310
KMN 9 (9×6×5)	B1: 1740 B2: 1900 B3: 1600 B4: 1980 B5: 3130	5, 3, 9, 7, 1, 4, 2, 6, 8	M1: 2440 M2: 2865 M3: 2985 M4: 1835 M5: 1860	B1: 1740 B2: 1900 B3: 1600 B4: 1980 B5: 3130	2, 6, 1, 8, 3, 4, 7, 9, 5	M1: 2440 M2: 2865 M3: 2985 M4: 1835 M5: 1860

Table 19 (continued)

Instance (M×J/B×O)	GA			SA		
	BWT (min)	MASEQ	MWT (min)	BWT (min)	MASEQ	MWT (min)
	B6: 2365		M6: 1955 M7: 2255 M8: 3090 M9: 2450	B6: 2365		M6: 1955 M7: 2255 M8: 3090 M9: 2450
KMN 10 (9×7×5)	B1: 300 B2: 890 B3: 630 B4: 1380 B5: 1880 B6: 1475 B7: 400	5, 8, 4, 3, 7, 6, 1, 2, 9	M1: 1770 M2: 985 M3: 1265 M4: 660 M5: 1340 M6: 1650 M7: 1065 M8: 1950 M9: 370	B1: 300 B2: 890 B3: 630 B4: 1380 B5: 1880 B6: 1475 B7: 400	3, 4, 2, 9, 6, 5, 7, 8	M1: 1770 M2: 985 M3: 1265 M4: 660 M5: 1340 M6: 1650 M7: 1065 M8: 1950 M9: 370
KMN 11 (9×7×6)	B1: 2200 B2: 2050 B3: 1950 B4: 2400 B5: 3690 B6: 3100 B7: 1600	5, 6, 4, 7, 8, 2, 9, 3, 1	M1: 2560 M2: 2335 M3: 1525 M4: 2770 M5: 2520 M6: 2125 M7: 3550 M8: 3255 M9: 2850	B1: 2200 B2: 2050 B3: 1950 B4: 2400 B5: 3690 B6: 3100 B7: 1600	8, 5, 6, 7, 9, 2, 4, 3, 1	M1: 2560 M2: 2335 M3: 1525 M4: 2770 M5: 2520 M6: 2125 M7: 3550 M8: 3255 M9: 2850
KMN 12 (9×7×7)	B1: 280 B2: 1790 B3: 30 B4: 2400 B5: 3650 B6: 3155 B7: 50	5, 9, 6, 1, 2, 8, 7, 3, 4	M1: 2575 M2: 2710 M3: 1975 M4: 2110 M5: 1630 M6: 1250 M7: 1670 M8: 3390 M9: 2455	B1: 320 B2: 1830 B3: 70 B4: 2440 B5: 3690 B6: 3195 B7: 40	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 2615 M2: 2750 M3: 2015 M4: 2150 M5: 1670 M6: 1290 M7: 1710 M8: 3430 M9: 2495
KMN 13 (9×8×5)	B1: 860 B2: 2120 B3: 1080 B4: 2640 B5: 3570 B6: 2485 B7: 1020 B8: 3405	6, 7, 9, 1, 5, 8, 3, 4, 2	M1: 2525 M2: 3190 M3: 2980 M4: 1265 M5: 1960 M6: 2040 M7: 2515 M8: 2220 M9: 1425	B1: 860 B2: 2120 B3: 1080 B4: 2640 B5: 3570 B6: 2485 B7: 1020 B8: 3405	2, 3, 4, 5, 6, 7, 8, 9, 1	M1: 2525 M2: 3190 M3: 2980 M4: 1265 M5: 1960 M6: 2040 M7: 2515 M8: 2220 M9: 1425
KMN 14 (9×8×6)	B1: 3660 B2: 4180 B3: 3420 B4: 4650 B5: 5990 B6: 4710 B7: 2550 B8: 5625	6, 3, 8, 5, 1, 4, 2, 7, 9	M1: 5615 M2: 4600 M3: 3700 M4: 3880 M5: 4685 M6: 3250 M7: 4245 M8: 5080 M9: 3840	B1: 3660 B2: 4180 B3: 3420 B4: 4650 B5: 5990 B6: 4710 B7: 2550 B8: 5625	3, 2, 4, 8, 7, 9, 1, 5, 6	M1: 5615 M2: 4600 M3: 3700 M4: 3880 M5: 4685 M6: 3250 M7: 4245 M8: 5080 M9: 3840
KMN 15 (9×8×7)	B1: 3380 B2: 4580 B3: 3900 B4: 5100 B5: 6530 B6: 5905 B7:46 0 B8: 6360	9, 3, 7, 1, 5, 8, 4, 2, 6	M1: 5760 M2: 3400 M3: 4245 M4: 3825 M5: 4845 M6: 5160 M7: 5905 M8: 5880 M9: 4115	B1: 3380 B2: 4580 B3: 3900 B4: 5100 B5: 6530 B6: 5905 B7: 4600 B8: 6360	1, 2, 4, 7, 3, 6, 9, 8, 5	M1: 5760 M2: 3400 M3: 4245 M4: 3825 M5: 4845 M6: 5160 M7: 5905 M8: 5880 M9: 4115
KMN 16 (9×8×8)	B1: 30 B2: 2310 B3: 1670	2, 3, 4, 1, 8, 7, 9, 6, 5	M1: 3180 M2: 2395 M3: 2140	B1: 35 B2: 2345 B3: 1705	2, 3, 6, 9, 7, 8, 5, 4, 1	M1: 3215 M2: 2430 M3: 2175

Table 19 (continued)

Instance (M×J/B×O)	GA			SA		
	BWT (min)	MASEQ	MWT (min)	BWT (min)	MASEQ	MWT (min)
	B4: 2600 B5: 4000 B6: 3125 B7: 860 B8: 3530		M4: 2400 M5: 2345 M6: 1940 M7: 2950 M8: 3590 M9: 1705	B4: 2635 B5: 4035 B6: 3160 B7: 895 B8: 3565		M4: 2435 M5: 2380 M6: 1975 M7: 2985 M8: 3625 M9: 1740
KMN 17 (9×9×5)	B1: 975 B2: 1745 B3: 1465 B4: 2365 B5: 3355 B6: 2800 B7: 295 B8: 3190 B9: 825	2, 1, 4, 5, 9, 3, 8, 7, 6	M1: 2665 M2: 345 M3: 2650 M4: 1280 M5: 2810 M6: 2125 M7: 965 M8: 1670 M9: 2505	B: 685 B2: 1455 B3: 1175 B4: 2075 B5: 3065 B6: 2510 B7: 50 B8: 2900 B9: 535	2, 5, 4, 1, 7, 6, 9, 3, 8	M1: 2375 M2: 55 M3: 2360 M4: 990 M5: 2520 M6: 1835 M7: 675 M8: 1380 M9: 2215
KMN 18 (9×9×6)	B1: 460 B2: 2180 B3: 1060 B4: 2500 B5: 3040 B6: 2760 B7: 40 B8: 3010 B9: 310	6, 7, 9, 8, 1, 4, 3, 2, 5	M1: 2075 M2: 1020 M3: 1760 M4: 995 M5: 1850 M6: 1325 M7: 2700 M8: 2190 M9: 1445	B1: 480 B2: 2200 B3: 1080 B4: 2520 B5: 3060 B6: 2780 B7: 60 B8: 3030 B9: 330	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 2095 M2: 1040 M3: 1780 M4: 1015 M5: 1870 M6: 1345 M7: 2720 M8: 2210 M9: 1465
KMN 19 (9×9×7)	B1: 3740 B2: 3860 B3: 2640 B4: 4350 B5: 5440 B6: 4140 B7: 2800 B8: 5220 B9: 3000	8, 6, 7, 4, 2, 3, 5, 1, 9	M1: 4915 M2: 2785 M3: 2665 M4: 3165 M5: 3550 M6: 2850 M7: 3220 M8: 4385 M9: 4855	B1: 3740 B2: 3860 B3: 2640 B4: 4350 B5: 5440 B6: 4140 B7: 2800 B8: 5220 B9: 3000	8, 6, 7, 4, 2, 3, 5, 1, 9	M1: 4915 M2: 2785 M3: 2665 M4: 3165 M5: 3550 M6: 2850 M7: 3220 M8: 4385 M9: 4855
KMN 20 (9×9×8)	B1: 840 B2: 3600 B3: 2160 B4: 3780 B5: 4490 B6: 4165 B7: 3300 B8: 4335 B9: 2660	8, 7, 9, 2, 1, 3, 4, 5, 6	M1: 3065 M2: 1460 M3: 3310 M4: 3000 M5: 2400 M6: 3230 M7: 2490 M8: 3635 M9: 3440	B1: 840 B2: 3600 B3: 2160 B4: 3780 B5: 4490 B6: 4165 B7: 3300 B8: 4335 B9: 2660	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 3065 M2: 1460 M3: 3310 M4: 3000 M5: 2400 M6: 3230 M7: 2490 M8: 3635 M9: 3440
KMN 21 (9×9×9)	B1: 2950 B2: 5170 B3: 3570 B4: 5790 B5: 6900 B6: 6300 B7: 60 B8: 6945	5, 4, 3, 2, 1, 9, 6, 8, 7	M1: 4935 M2: 4395 M3: 3200 M4: 4095 M5: 2905 M6: 5765 M7: 5520 M8: 4810 M9: 4050	B1: 3050 B2: 5270 B3: 3670 B4: 5890 B5: 7000 B6: 6400 B7: 100 B8: 7045 B9: 2150	1, 2, 3, 4, 5, 6, 7, 8, 9	M1: 5035 M2: 4495 M3: 3300 M4: 4195 M5: 3005 M6: 5865 M7: 5620 M8: 4910 M9: 4150

Para parameters, *BWT* batch waiting time, *GA* genetic algorithm, *TTC* total transportation cost, *MWT* machine waiting time, *SA* simulated annealing, *MAKSP* total make span, *CPU* computational time, *MASEQ* machine sequence

transportation cost with machine sequences considering scheduling parameters as constraints such as MAKSP is determined for loop layout by running the C++ code on eclipse (IDE) tool for ten test runs. The performance of the proposed algorithm is tested over a number of problems selected from the literature and comparison is made between GA and SA. The experimental results reveal that the proposed genetic algorithm is effective and efficient for loop layout design. From the graph, it is clear that for loop layout, the total transportation cost is less for lower level problems and reaches to high value as the problem size enhanced. Furthermore, it is concluded that GA provides optimum solutions than SA, but computational time is more than SA.

References

1. Luggen WW (1991) Flexible manufacturing cells and systems. Prentice-Hall, Inc. A division of Simon & Schustes Englewood Cliffs, New Jersey 07632 ISBN 0-13-321738-8
2. Tompkins JA, White JA, Bozer YA, Janchoco JMN (2003) Facilities planning, 3rd edn. Wiley, New York
3. Apple JM (1977) Plant layout and material handling, 3rd edn. The Ronald Press Co, New York
4. Sateesh Kumar RM, Ashokan P, Kumanan S (2008) Design of loop layout. In flexible manufacturing system using non-traditional optimization technique. Int J Adv Manuf Technol 38:594–599
5. Groover MP (1992) Automation, Production systems and computer-integrated manufacturing. Prentice-Hall Inc. ISBN-0-87692-618-9
6. Ruey-Mawchen^{+}, Shin-Tang Lo^{++} (2006) using ant colony system to solve resource constrained project scheduling problems. Int-J. Comp Sci Netw Secu vol. 6 No-11, November 2006
7. Xia W-J, Wu Z-M (2006) A hybrid particle swarm optimization approach for the job-shop scheduling problem. Int J Adv Manuf Technol 29:360–366
8. Tiwari A, Chang P-C (2014) A Pareto block-based estimation and distribution algorithm for multi-objective permutation flow shop scheduling problem. Int J Prod Res 53(3):793–834. doi:10.1080/00207543.2014.933273
9. Muthuswamy S, Vélez-Gallego MC (2012) Minimizing makespan in a two-machine no-wait flow shop with batch processing machines. Int J Adv Manuf Technol 63:281–290
10. Ayough A, Zandieh M (2012) GA and ICA approaches to job rotation scheduling problem: considering employee's boredom. Int J Adv Manuf Technol 60:651–666
11. Filho MG, Barco CF (2012) Using Genetic Algorithms to solve scheduling problems on flexible manufacturing systems (FMS): a literature survey, classification and analysis. Flex Serv Manuf J. doi:10.1007/s10696-012-9143-6
12. Udhayakumar P, Kumanan S (2010) Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm
13. Costa A, Cappadonna FA (2014) Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment. Int J Prod Res 52(9):2696–2728. doi:10.1080/00207543.2014.883469
14. Javadian N, Fattahi P (2012) An immune algorithm for hybrid flow shop scheduling problem with time lags and sequence-dependent setup times. Int J Adv Manuf Technol 63:337–348
15. Ranjbar M, Razavi MN (2012) A hybrid metaheuristic for concurrent layout and scheduling problem in a job shop environment. Int J Adv Manuf Technol 62:1249–1260. doi:10.1007/s00170-011-3859-4
16. Wangta P-A, Pongcharoen P (2010) Designing machine layout using tabu search and simulated annealing. Naresuan Univ J 2010:18(3)
17. Khamseh A, Jolai F (2014) Integrating sequence-dependent group scheduling problem and preventive maintenance in flexible flow shops. Int J Adv Manuf Technol. doi:10.1007/s00170-014-6429-8
18. Karthikeyan S, Asokan P (2014) A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints. Int J Adv Manuf Technol 72:1567–1579. doi:10.1007/s00170-014-5753-3
19. Pooranian Z, Shojafar M (2013) A hybrid metaheuristic algorithm for Job scheduling on computational grids. Informatica 37(2013):157–164
20. Liu H, Abraham A, Grosan C, Li N, (2007) A novel variable neighbourhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. Digital Information Management, 2007. ICDIM '07. 2nd International Conference on Volume: 1 doi:10.1109/ICDIM.2007.4444214, Page(s): 138 - 145
21. Kumanan S, Udhayakumar P (2008) Integrated scheduling of flexible manufacturing system using evolutionary algorithms. Int J Adv Manuf Technol 38:594–599
22. Ulusoy G, Serifoglu FS, Bilge U (1997) A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Comput Ops Res 14(4):335–351