Int J Adv Manuf Technol (2006) 29: 184–196
DOI 10.1007/s00170-004-2492-x

ORIGINAL ARTICLE

R.K. Suresh · K.M. Mohanasundaram

Pareto archived simulated annealing for job shop scheduling with multiple objectives

Received: 24 November 2004 / Accepted: 24 November 2004 / Published online: 27 July 2005

Abstract In this paper, the job shop scheduling problem is studied with the objectives of minimizing the makespan and the mean flow time of jobs. The simultaneous consideration of these objectives is the multi-objective optimization problem under study. A metaheuristic procedure based on the simulated annealing algorithm called Pareto archived simulated annealing (PASA) is proposed to discover non-dominated solution sets for the job shop scheduling problems. The seed solution is generated randomly. A new perturbation mechanism called segment-random insertion (SRI) scheme is used to generate a set of neighbourhood solutions to the current solution. The PASA searches for the non-dominated set of solutions based on the Pareto dominance or through the implementation of a simple probability function. The performance of the proposed algorithm is evaluated by solving benchmark job shop scheduling problem instances provided by the OR-library. The results obtained are evaluated in terms of the number of non-dominated schedules generated by the algorithm and the proximity of the obtained non-dominated front to the Pareto front.

Keywords Job shop scheduling · Multi-objective optimization · Simulated annealing

Notations

n	Number of jobs
m	Number of machines
q	Number of objectives
C_i	The completion time of job i
p_{ij}	The processing time of operation j of job i
mk	Makespan of the schedule $= \max\{C_i, i = 1, 2, 3, \dots, n\}$
mft	Mean flow time of the schedule $= \frac{1}{n}\sum_{i=1}^{n} C_i$
S_s	The seed or current solution
O_{ij}	jth operation of ith job
$\{S_{s'}\}$	Neighbourhood solution set generated by the perturbing mechanism
N_{nd}	Number of non-dominated solutions present in the neighbourhood $\{S_{s'}\}$
$S_{s'}$	The candidate solution selected from the neighbourhood of S_s
mk_e	The best makespan value obtained during the search
mft_e	The best total flow time value obtained during the search
w_i	The non-negative weight for the ith objective, such that $\sum_{i=1}^{q} w_i = 1.0$
Z_i	The weighted sum of the scaled objectives for the ith neighourhood solution, $1 \le i \le N_{\mathrm{nd}}$

R.K. Suresh (✉)
Department of Production Engineering,
Amrita Institute of Technology,
Amrita Vishwa Vidyapeetham, Coimbatore – 641 105, India
E-mail: rk_suresh@ettimadai.amrita.edu

K.M. Mohanasundaram
Department of Mechanical Engineering,
PSG College of Technology,
Coimbatore – 641 105, India
E-mail: kmmsundaram@yahoo.com

1 Introduction

The job shop scheduling problem (JSP) with a single objective is a widely researched problem in the area of production scheduling. In a job shop, several jobs require scheduling, each with different processing times on different machines. Many applications of JSPs in industry have been discussed in the literature. Operations research practitioners, production management experts, management scientists, mathematicians and computer scientists have discussed the scheduling theory [1–7].

The solution procedure for solving the JSP differs as the objective of the scheduling differs. Most of the research concerning the job shop scheduling problem have focused on developing scheduling algorithms for a single objective measure [8]. A detailed overview of the objectives of job shop is given in [5, 9, 10]. Much work has been done to solve JSPs by using single objective metaheuristic procedures like simulated annealing algorithm, genetic algorithm and tabu search algorithm [11]. These algorithms are generic optimization algorithms, i.e. they are intended for use on a wide range of optimization problems.

The real-world scheduling problems are multi-objective in nature. In such cases, several objectives are considered simultaneously when a schedule is generated. Simultaneous consideration of several objectives during scheduling totally modifies the scheduling approach. A scheduler who improves the schedule with respect to one objective may want to know how the schedule performs with respect to the other objectives. Thus the goal is to generate a feasible schedule that minimizes several objectives. This schedule is called a *Pareto optimal solution*. A single feasible schedule that minimizes several objectives may not exist. In other words, individual optimal solutions of each objective are usually different. Under such situations, the scheduler may be interested in having a schedule with weighted combination of several scheduling objectives as the performance measure. It is possible that the weights of various objectives are known before scheduling. This approach [12–14] permits computing of a unique strict Pareto optimal solution. It is also possible that the decision maker wants to choose a Pareto optimal solution according to the priorities existing at the time of decision making. In that case, a family of best trade-off schedules called the Pareto optimal set is to be found. The set of Pareto solutions is called the *Pareto front*. Therefore solving a multi-objective scheduling problem is a Pareto optimization problem.

Generating the Pareto optimal set for the scheduling problem can be computationally expensive and is often infeasible, because of the complexity of the scheduling problem [15]. Moreover, when metaheuristics are used, there is no guarantee that the Pareto set for a given multi-objective optimization problem like multi-objective scheduling can be generated. However, a set of non-dominated solutions can be generated close to the Pareto optimal set [15–17].

2 Literature survey

Researchers in the field of multi-objective optimization have developed several multi-objective optimization algorithms. Suresh and Sahu [18] proposed a SA algorithm based multi-objective optimization method for solving COPs. Extensions of single objective GAs were proposed in different forms for multi-objective optimization by Schaffer [19], Fonseca and Fleming [20], Srinivas and Deb [21], Deb et al. [22] and Chang et al. [23]. The vector evaluated GA (VEGA) proposed by Schaffer was criticized for not generating a compromise solution by favouring the extreme solutions. Fonseca and Fleming [20] used the Pareto dominance relationship in their multi-objective genetic algorithm (MOGA). The performance of the MOGA depends on the value of the sharing factor. Srinivas and Deb [21] proposed the non-dominated sorting genetic algorithm (NSGA). Absence of elitism and sensitiveness to the value of sharing factor are reported to be the major drawbacks of the NSGA approach. However, Deb et al. [22] proposed the fast and elitist NSGA known as NSGA-II to overcome the above drawbacks. The above GA based approaches have been tested on continuous or very small discrete problems only.

Ishibuchi and Murata [24] proposed the multi-objective genetic local search (MOGLS) algorithm for solving two and three objective flow shop scheduling problems. Bagchi [17] proposed the elitist non-dominated sorting GA (ENGA) for solving multi-objective flow shop scheduling problems. ENGA is an adapted version of NSGA. The better performance of ENGA is due to its elitist strategy. Ishibuchi et al. [15] proposed the modified MOGLS algorithm and compared its performance with the strength Pareto approach of Zitzler and Thiele [25] and NSGA II algorithm by using the results obtained for the randomly generated flow shop scheduling test problems with 20 machines. Chang et al. [23] proposed the GA based gradual priority weighting approach called GPWGA to search the non-dominated solutions.

Of late Varadharajan (2003, personal communication) proposed a multi-objective simulated annealing algorithm (mentioned in this paper as “VR” algorithm) for scheduling in flow shops to minimize makespan and total flow time of jobs and presented non-dominated solution sets for the benchmark flow shop problems of Taillard [26]. The performance of the VR algorithm was shown to be superior to the GA based algorithms such as ENGA, GPWGA, and MOGLS and the *a-posteriori* approach which is based on NEH heuristic [27]. To our knowledge, research on multi-objective job shop scheduling is rather limited.

Scheduling problems are combinatorial optimization problems. In most cases, they are NP hard for even a single criterion optimization and are therefore unlikely to be solvable in polynomial time. This difficulty is due to their combinatorial complexity. NP completeness proofs [28] are available for a number of scheduling problems. JSP is proved to be NP hard [29, 30]. Chen and Bulfin [31] presented a thorough study on the complexity analysis of the multi-machine, multi-objective scheduling problems. It is shown that considering more than one objective does not simplify the scheduling problem. Multi-objective scheduling problems are as complex as the corresponding single objective problems [8].

3 The problem under study

The deterministic job shop scheduling problem considered in this paper consists of a finite set J of n jobs to be processed on a finite set M of m dedicated machines. Each job J_i must be processed on every machine once and consists of a set of m operations $\{O_{i1}, O_{i2}, O_{i3}, \ldots O_{im}\}$, which have to be scheduled in a predetermined order, a requirement called a precedence constraint. The routing of one job is independent of the routing of another job. There are N operations in total, $N = n \times m$. Each operation is to be processed for an uninterrupted processing period of p_{ij}. In the proposed model, no machine breakdowns are assumed to occur, transport times of jobs between machines are ignored and all the jobs are assumed to be available at time zero. The process time p_{ij} is assumed to be known in advance, and necessary setup times are included in the processing times. In the present work, regular measures of performance, namely minimizing the makespan and the mean flow time are considered.

Benchmark job shop scheduling problem instances covering small, medium and large size problems, provided by OR-library (http://mscmga.ms.ic.ac.uk/info.html) under various classes have been solved by using the proposed PASA. The various benchmark JSP instances under study are: three instances (ft06, ft10, and ft20) from Fisher and Thompson [1], forty instances (la01–la40) proposed by Lawrence [32], five instances (abz5–abz9) due to Adams et al. [33], ten instances (orb01–orb10) proposed by Applegate and Cook [34], twenty instances (swv01-swv20) proposed by Storer et al. [35] and four instances (yn01–yn04) proposed by Yamada and Nakano [36].

4 The Pareto archived simulated annealing algorithm (PASA)

General purpose optimization methods such as SA, GA and Tabu search (TS) methods have been proposed for multi-objective optimization. The proposed method of solving JSP is based on SA algorithm. Reviews of the theory and application of SA can be found in Kirkpatrick et al. [37] and Aarts and Lenstra [38]. SA has been applied to solve single objective job shop scheduling problems [39–42]. Suresh and Sahu [18] and Czyzak and Jaskiewicz [43] have proposed a SA algorithm based multi-objective optimization approach. The approach proposed by Suresh and Sahu [18] is an *apriori* approach. Czyzak and Jaskiewicz [43] proposed Pareto simulated annealing (PSA) which is a kind of parallel search with a set of solutions using a SA algorithm based extension of simulated annealing. The primary contribution of the present research is the development of a single point local search metaheuristic to solve job shop scheduling problems with multiple objectives.

The characteristic features of the proposed PASA are:

1. A single point local search heuristic is used.
2. A set of neighbourhood solutions are considered to identify a candidate solution.
3. Pareto dominance is used as the criterion for accepting the candidate solution.
4. An archive is created and maintained to preserve the updated set of non-dominated solutions.
5. Re-annealing strategy is used to realize various search directions.

4.1 Pareto search and archiving

A SA algorithm is as such not capable of returning the Pareto optimal or non-dominated solution set from a single run. To preserve the non-dominated solutions obtained during the search process, an archive is maintained for storage. The Pareto search and archiving procedures of the proposed algorithm are explained below.

PASA proceeds its search with a randomly generated solution S_s in the direction specified by the objective axis. The objective axis is fixed by the weighting coefficients (w_1, w_2) representing the relative importance of the objectives. The new perturbation scheme SRI returns a set of neighbourhood solutions of the seed sequence in each iteration. Every member of the newly generated neighbourhood set is compared with the other members in the set. In the case of two objectives, an ith solution is said to dominate a jth solution, if the following condition is satisfied.

$$[((mk_i \leq mk_j) \text{ AND } (mft_i \leq mft_j)) \\ \text{AND } ((mk_i < mk_j) \text{ OR } (mft_i < mft_j))] \quad (1)$$

Once a solution is identified as a dominated solution, it is removed from the generated neighbourhood set. After all comparisons, non-dominated solutions among the $\{S_{s'}\}$ will be left. Then a solution with the least value of Z is (see Eq. 2) returned as the candidate solution $S_{s'}$. If there is a tie, $S_{s'}$ is chosen randomly.

$$Z_i = w_1\,((mk_i - mk_e) \div mk_e) + w_2\,(mft_i - mft_e) \\ \div mft_e * (mft_e/mk_e)) \quad (2)$$

The candidate solution $S_{s'}$ is then compared with the current solution S_s for non-domination. If the candidate solution dominates the current solution, then $S_{s'}$ becomes the current solution. Otherwise, the dominated candidate solution is accepted with the acceptance probability p_{accept} as given in Eq. 3.

$$p_{\text{accept}} = \exp^{-(\nabla/T)} \quad (3)$$

$$\text{where } \nabla = \left[\frac{w_1 * (mk_{s'} - mk_s)}{mk_s} + \frac{w_2 * (mft_{s'} - mft_s)}{mft_s}\right] \quad (4)$$

Whenever a candidate solution $S_{s'}$ is accepted, it is taken as current solution and is compared with every member of the archive. If an archive member dominates the candidate solution, comparison is terminated. Otherwise, if the candidate solution dominates any archive member, the dominated archive member is removed from the archive. In the later case, the candidate solution is copied into the archive after the comparison is over. Irrespective of whether the candidate solution is added into the archive or not, the search process is continued with the current solution. Thus, the Pareto dominance relationship is used as the acceptance criterion. However, an inferior candidate solution is accepted with the probability computed using Eq. 3. The weighting coefficients represent the relative importance of the objectives.

4.2 Re-start strategy

Sometimes during the search process, SA encounters non-improving iterations continuously. To overcome such drawbacks, a FIFO queue is maintained to store a set of recently accepted candidate solutions. During such continuous non-improving moves, PASA retrieves a seed solution at random from a FIFO queue to encourage a different search trajectory. The maximum number of non-improving moves is taken as 100, and the queue size is limited to five based on the trial runs.

4.3 Parameter settings

The values of initial and final temperature during annealing are fixed as follows [37]. By accepting an inferior candidate solution

$S_{s'}$, which is inferior by 30% (δ) relative to the current solution S_s with the acceptance ratio (x_0) of 0.9, the initial temperature (T_i) is fixed as 285 (see Eq. 5).

$$T_i = \left(\frac{-\delta}{in\left(x_0^{-1}\right)} \right) \tag{5}$$

The temperature is reduced after a predetermined number of iterations (E) at a given temperature, using the relationship ($T_{i+1} = r_c * T_i$). Reduction factor (r_c) is fixed as 0.9. The final temperature, which is based on the value of the initial temperature and reduction factor, is fixed as 5. These parameters are obtained after conducting several trials on different JSP instances.

4.4 Re-annealing

Re-annealing refers to restart of the SA process with the best solution obtained during the previous run as the seed solution. The objective axis during the search process is changed by changing the weighting coefficients to uncover more non-dominated solutions in the solution space. This is done after every single run of PASA to realize various search directions. Direction of search for the present problem with two objectives is specified by (w_1, w_2), such that $w_1 + w_2 = 1.0$. Initially w_1 is taken as 0 and w_2 is taken as 1.0. After carrying out the search for a predetermined number of iterations in a given direction specified by w_1 and w_2, the search process is repeated again with $w_1 = (w_1 + 0.2)$ and $w_2 = (1.0 - w_1)$. This is because when temperature is low the probability that an improving neighbour is chosen is small. Since, no better solutions can be found in the direction when the temperature is already low re-annealing is employed. During re-annealing, the temperature and other parameters are re-set to initial values. Re-annealing is done until w_1 becomes 1.0. The number of solutions evaluated by the search process in a given direction is limited such that the total number of solutions evaluated equals ($n \times m \times 10000$) solutions.

4.5 Solution structure and the perturbation mechanism

A schedule is expressed exactly using a finite length of string representing various operations to be performed in the order specified. Thus, the solution structure consists of a string of $n \times m$ integers. This covers all feasible solutions of a JSP instance. For example, the string (0 1 0 2 1 2 ...) represents the first operation of job 0 is to be processed first followed by the first operation of job 1, the second operation of job 0, etc. The working of SRI scheme is explained through the following numerical illustration. Consider a sequence of operations (0 1 0 2 1 2 1 0 2) of a 3 job 3 machine job shop scheduling problem. Job sequences of the machines m_0, m_1 and m_2 are taken to be {(0 1 2), (0 2 1), (1 0 2)}. It is to be noted that both the job number and the machine number starts from 0. Let the sequence (0 1 0 2 1 2 1 0 2) be a seed sequence S. With known locus P and segment length L, a sub-sequence is selected. Neighbourhood solutions are obtained by random insertion of each element of the selected sub-sequence to the left or right side of the sub-sequence. Perturbed solutions along with corresponding job sequence for various machines are shown in Table 1 and Table 2, respectively. It is seen from the above tables that neighbours are generated from the original solution within a small spread by changing the job sequence of a machine. This is very much desirable for a thorough exploration of the search space.

4.6 Proposed PASA algorithm for multi-objective scheduling

The pseudo code of the PASA algorithm for solving the job shop scheduling problem is given below.

Step 1 Assign SA parameters such as final temperature (T_f), rate of cooling (r_c), maximum number of iterations (E) at a given temperature and maximum number of successive non-improving moves (B).
Step 2 Generate a seed solution S_S randomly.
Step 3 Initialize w *to* 0 and the FIFO queue by adding S_S.
Step 4 Initialize SA parameters such as current temperature (T_i), the iteration counter (e), non-improvement counter (r) and the weighting coefficients $w_1 = w$ and $w_2 = (1.0 - w_1)$.
Step 5 Invoke SRI to generate a neighbourhood set $\{S_{S'}\}$.
Step 6 Do non-dominated sorting of the neighbourhood and identify $S_{S'}$.
Step 7 If ($S_{S'}$ dominates S_S)
  copy $S_{S'}$ into S_S and into the archive,
  update archive members,
  assign $r = 0$,
  go to Step 9.
 else
  compute ∇ using Eq. 3.
  if $\left(e^{-(\nabla/T_i)} < U\right)$
   copy $S_{S'}$ into S_S,

Table 1. Perturbed solution set obtained ($P = 4$; $L = 3$)

Jobs selected* for insertion (*bold letter)	Random insert position	Resulting sequence (s)
0 1 0 **2** 1 2 1 0 2	3	$S'_1 = 0\ 1\ 2\ 0\ 1\ 2\ 1\ 0\ 2$
0 1 0 2 **1** 2 1 0 2	1	$S'_2 = 1\ 0\ 1\ 0\ 2\ 2\ 1\ 0\ 2$
0 1 0 2 1 **2** 1 0 2	2	$S'_3 = 0\ 2\ 1\ 0\ 2\ 1\ 1\ 0\ 2$

Table 2. Job sequences of the machines for the perturbed solutions using SRI

Perturbed solutions	Job sequence	
0 1 2 0 1 2 1 0 2	M0	0 1 2
	M1	2 0 1
	M2	1 0 2
1 0 1 0 2 2 1 0 2	M0	1 0 2
	M1	0 2 1
	M2	1 0 2
0 2 1 0 2 1 1 0 2	M0	0 1 2
	M1	2 0 1
	M2	1 0 2

assign $r = 0$,
go to Step 9.
else
go to Step 8.
Step 8 If $(r < B)$
increment r,
go to Step 9.
else
pick a solution S at random from the FIFO queue,
Set $r = 0$.
Step 9 Update FIFO queue and increment iteration counter e.
Step 10 If $(e < E)$
go to Step 5.
else
$T_i = T * r_c$
set $e = 0$.
Step 11 If $(T_i < T_f)$
go to Step 5.
else
set $w = w + 0.2$.
Step 12 If $(w \leq 1.0)$
go to Step 4.
else
output archive members.

5 Quality measures of non-dominated solution set

In order to compare different non-dominated solution sets with one another, some of the quality measures are explained below. Some solutions in one set may be dominated by solutions in the other set. When the number of objectives to be optimized is two or three, graphical plots such as shown in Fig. 1 are useful. Multi-dimensional objective space requires a different approach.

Many common metrics are used in the literature (see Knowles [47] for complete study) for this purpose. Most of the proposed metrics use the true Pareto optimal solution set as the reference set for evaluating the quality of the given non-dominated solution set. Ishibuchi et al. [15] generated the reference set for each of the 20-job test problem with a much longer computational time and larger computer memory. Generating the true Pareto front requires very high computational effort especially for the JSP under study. The required computational effort becomes very high when the problem size is large. In this paper, a relative measure is used for comparison. The net non-dominated front obtained by updating the combined non-dominated front formed by adding non-dominated solutions generated by various algorithms under comparison is used as the reference set. Quality of the non-dominated solution generated by an algorithm is evaluated using *net front contribution ratio* (NFCR). The computational aspect of the measure is outlined below. Let F_1, F_2 and F_3 be the non-dominated fronts obtained by different algorithms. These fronts are then combined to form a combined front. A net front F_n is obtained by updating the combined front. Let n_1, n_2 and n_3 be the number of non-dominated individuals contributed by F_1, $F_2$and F_3 respectively to the net front F_n. The net front contribution ratio of each of the algorithm is computed using Eq. 6.

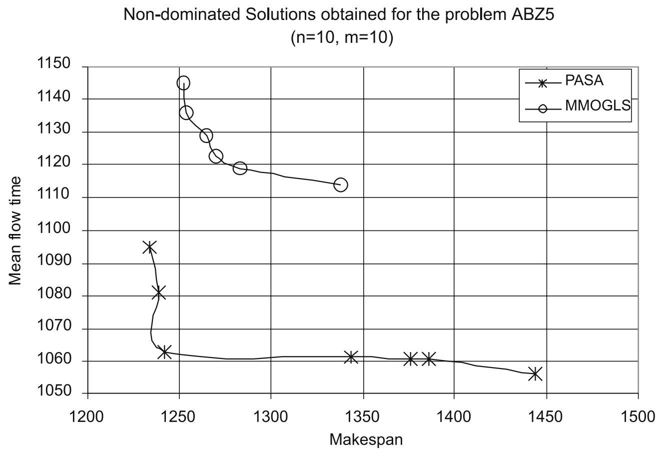

Fig. 1. Graphical comparison of the quality of non-dominated fronts obtained by the PASA and the modified MOGLS algorithm for the problem instance ABZ5

$$NFCR_1 = n_1/n_n,\ NFCR_2 = n_2/n_n \text{ and } NFCR_3 = n_3/n_n \qquad (6)$$

Several researchers reported the best UB for the makespan [11] and mean flow time [44, 45] in their study using single objective

Table 3. Net front contribution ratio by PASA, VR and modified MOGLS algorithms for the 20-job, flow shop scheduling problems of Taillard (1993)

n	m			Problem number 1	2	3	4	5	6	7	8	9	10	Average NFCR
20	5	PASA	NFC1	1.000	0.909	0.417	0.500	0.938	0.750	0.818	0.667	0.706	0.580	0.730
		VR	NFC2	0.000	0.182	0.083	0.278	0.188	0.188	0.182	0.167	0.000	0.260	0.150
		MMOGLS	NFC3	0.000	0.000	0.500	0.278	0.125	0.188	0.091	0.222	0.353	0.110	0.190
20	10	PASA	NFC1	0.400	0.607	0.538	0.556	0.500	0.261	0.588	0.500	0.400	0.260	0.460
		VR	NFC2	0.333	0.143	0.385	0.167	0.571	0.522	0.588	0.500	0.250	0.210	0.370
		MMOGLS	NFC3	0.333	0.357	0.538	0.333	0.071	0.217	0.412	0.056	0.500	0.580	0.340
20	20	PASA	NFC1	0.333	0.458	0.655	0.706	0.704	0.176	0.000	0.667	0.625	0.520	0.480
		VR	NFC2	0.500	0.417	0.276	0.176	0.185	0.176	0.133	0.167	0.125	0.390	0.250
		MMOGLS	NFC3	0.375	0.167	0.276	0.176	0.111	0.706	0.867	0.167	0.281	0.260	0.340

optimization algorithms. In the present work, effectiveness of the proposed algorithm in obtaining the Pareto front is measured by considering the extreme solutions, i.e. the best makespan and the best mean flow time, of the Pareto optimal or near Pareto optimal solution set as the reference. An absolute measure namely the *mean relative percentage increase* of the extreme solutions

Table 4. Relative performance of PASA, compared to the best upper bound reported in the literature for the benchmark job shop scheduling problem instances of Fisher and Thompson (1963), measured in terms of mean relative percentage increase (MRPI) in makespan and mean flow time

Bench mark	*n*	*m*	Makespan (UB)	Best makespan obtained by PASA	% Deviation	MRPI in makespan	Mean flow time (UB)	Best mean flow time obtained by PASA	% Deviation	MRPI in mean flow time
ft06	6	6	55	55	0.000	0.00	44.167	44.170	0.008	0.00
ft10	10	10	930	938	0.860	0.86	750.100	750.500	0.053	0.86
ft20	20	5	1165	1165	0.000	0.00	692.100	**685.550**	0.000	0.00

Table 5. Relative performance of PASA, compared to the best upper bound reported in the literature for the benchmark job shop scheduling problem instances of Lawrence (1984), measured in terms of mean relative percentage increase (MRPI) in makespan and mean flow time

Bench mark	*n*	*m*	Makespan (UB)	Best makespan obtained by PASA	% Deviation	MRPI in makespan	Mean flow time (UB)	Best mean flow time obtained by PASA	% Deviation	MRPI in mean flow time
la01	10	5	666	666	0.000		483.200	483.200	0.000	
la02	10	5	655	655	0.000		445.900	446.800	0.202	
la03	10	5	597	597	0.000	0.00	415.100	417.500	0.578	0.16
la04	10	5	590	590	0.000		425.900	425.900	0.000	
la05	10	5	593	593	0.000		407.200	407.200	0.000	
la06	15	5	926	926	0.000		581.733	582.400	0.115	
la07	15	5	890	890	0.000		550.067	542.800	0.000	
la08	15	5	863	863	0.000	0.00	529.933	533.870	0.743	0.17
la09	15	5	951	951	0.000		615.667	609.870	0.000	
la10	15	5	958	958	0.000		600.800	588.070	0.000	
la11	20	5	1222	1222	0.000		729.500	723.800	0.000	
la12	20	5	1039	1039	0.000		606.700	605.250	0.000	
la13	20	5	1150	1150	0.000	0.00	680.850	691.650	1.586	0.51
la14	20	5	1292	1292	0.000		748.200	755.400	0.962	
la15	20	5	1207	1207	0.000		731.050	728.100	0.000	
la16	10	10	945	945	0.000		739.300	743.500	0.568	
la17	10	10	784	784	0.000		653.700	656.400	0.413	
la18	10	10	848	848	0.000	0.11	705.200	708.000	0.397	0.44
la19	10	10	842	842	0.000		726.000	722.700	0.000	
la20	10	10	902	907	0.554		746.400	752.500	0.817	
la21	15	10	1046	1055	0.860		838.000	840.530	0.302	
la22	15	10	927	927	0.000		791.600	812.800	2.678	
la23	15	10	1032	1032	0.000	0.61	845.267	867.000	2.571	2.22
la24	15	10	935	945	1.070		807.000	814.000	0.867	
la25	15	10	977	988	1.126		785.733	822.600	4.692	
la26	20	10	1218	1218	0.000		986.600	999.700	1.328	
la27	20	10	1235	1264	2.348		1016.200	1110.350	9.265	
la28	20	10	1216	1225	0.740	1.36	975.650	1026.500	5.212	4.26
la29	20	10	1152	1195	3.733		914.350	962.550	5.272	
la30	20	10	1355	1355	0.000		1009.850	1012.300	0.243	
la31	30	10	1784	1784	0.000		1229.633	1301.370	5.834	
la32	30	10	1850	1850	0.000		1340.433	1396.670	4.195	
la33	30	10	1719	1719	0.000	0.00	1204.867	1265.970	5.071	3.77
la34	30	10	1721	1721	0.000		1291.433	1289.530	0.000	
la35	30	10	1888	1888	0.000		1268.000	1315.500	3.746	
la36	15	15	1268	1282	1.104		1122.867	1144.730	1.947	
la37	15	15	1397	1422	1.790		1186.267	1199.130	1.084	
la38	15	15	1196	1208	1.003	0.00	1048.467	1048.870	0.038	0.00
la39	15	15	1233	1256	1.865		1055.600	1084.130	2.703	
la40	15	15	1222	1241	1.555		1064.800	1076.070	1.058	

Table 6. Relative performance of PASA, compared to the best upper bound reported in the literature for the benchmark job shop scheduling problem instances of Adams et al., (1988), measured in terms of mean relative percentage increase (MRPI) in makespan and mean flow time

Bench mark	*n*	*m*	Makespan (UB)	Best makespan obtained by PASA	% Deviation	MRPI in makespan	Mean flow time (UB)	Best mean flow time obtained by PASA	% Deviation	MRPI in mean flow time
abz5	10	10	1234	1234	0.000	0.00	1056.300	1056.300	0.000	0.00
abz6	10	10	943	943	0.000		780.800	780.800	0.000	
abz7	20	15	656	682	3.963		589.850	591.150	0.220	
abz8	20	15	646	700	8.359	6.68	594.950	601.950	1.177	1.37
abz9	20	15	662	713	7.704		595.950	612.200	2.727	

Table 7. Relative performance of PASA, compared to the best upper bound reported in the literature for the benchmark job shop scheduling problem instances of Applegate and Cook (1991), measured in terms of mean relative percentage increase (MRPI) in makespan and mean flow time

Bench mark	*n*	*m*	Makespan (UB)	Best makespan obtained by PASA	% Deviation	MRPI in makespan	Mean flow time (UB)	Best mean flow time obtained by PASA	% Deviation	MRPI in mean flow time
orb01	10	10	1059	1059	0.000		810.600	**802.300**	0.000	
orb02	10	10	888	889	0.113		734.500	738.800	0.585	
orb03	10	10	1005	1022	1.692		812.800	814.100	0.160	
orb04	10	10	1005	1024	1.891		794.900	817.100	2.793	
orb05	10	10	887	889	0.225	0.50	697.800	699.800	0.287	0.46
orb06	10	10	1010	1013	0.297		815.500	816.800	0.159	
orb07	10	10	397	397	0.000		331.600	333.300	0.513	
orb08	10	10	899	906	0.779		695.400	695.800	0.058	
orb09	10	10	934	934	0.000		745.000	**736.100**	0.000	
orb10	10	10	944	944	0.000		789.600	**786.200**	0.000	

Table 8. Relative performance of PASA, compared to the best upper bound reported in the literature for the benchmark job shop scheduling problem instances of Storer et al., (1993), measured in terms of mean relative percentage increase (MRPI) in makespan and mean flow time

Bench mark	*n*	*m*	Makespan (UB)	Best makespan obtained by PASA	% Deviation	MRPI in makespan	Mean flow time (UB)	Best mean flow time obtained by PASA	% Deviation	MRPI in mean flow time
swv01	20	10	1407	1473	4.691		983.700	1008.300	2.501	
swv02	20	10	1475	1479	0.271		1053.300	1031.850	0.000	
swv03	20	10	1398	1474	5.436	3.75	1045.050	1077.400	3.096	3.16
swv04	20	10	1450	1510	4.138		1054.450	1104.150	4.713	
swv05	20	10	1424	1484	4.213		1037.950	1094.800	5.477	
swv06	20	15	1591	1806	13.514		1417.300	1430.100	0.903	
swv07	20	15	1447	1736	19.972		1317.250	1354.650	2.839	
swv08	20	15	1641	1914	16.636	15.55	1438.100	1483.350	3.147	2.04
swv09	20	15	1605	1798	12.025		1400.550	1429.550	2.071	
swv10	20	15	1632	1887	15.625		1438.150	1455.700	1.220	
swv11	50	10	2983	3233	8.381		2011.800	2075.040	3.143	
swv12	50	10	2972	3276	10.229		2029.500	2086.980	2.832	
swv13	50	10	3104	3295	6.153		2091.360	2133.680	2.024	
swv14	50	10	2968	3126	5.323		1948.460	2011.540	3.237	
swv15	50	10	2885	3146	9.047	3.91	1949.460	2005.760	2.888	2.12
swv16	50	10	2924	2924	0.000		1904.060	**1857.640**	0.000	
swv17	50	10	2794	2794	0.000		1805.400	1876.040	3.913	
swv18	50	10	2852	2852	0.000		1824.060	**1808.120**	0.000	
swv19	50	10	2843	2843	0.000		1866.580	1925.480	3.156	
swv20	50	10	2823	2823	0.000		1813.560	**1789.220**	0.000	

Table 9. Relative performance of PASA, compared to the best upper bound reported in the literature for the benchmark job shop scheduling problem instances of Yamada and Nakano (1992), measured in terms of mean relative percentage increase (MRPI) in makespan and mean flow time

Bench mark	*n*	*m*	Makespan (UB)	Best makespan obtained by PASA	% Deviation	MRPI in makespan	Mean flow time (UB)	Best mean flow time obtained by PASA	% Deviation	MRPI in mean flow time
yn1	20	20	846	920	8.747		827.150	840.950	1.668	
yn2	20	20	870	956	9.885	10.64	867.400	881.450	1.620	2.34
yn3	20	20	840	948	12.857		827.050	851.750	2.987	
yn4	20	20	920	1022	11.087		894.400	922.150	3.103	

of the obtained non-dominated front with respect to the best UB reported in the literature is used as the second quality measure.

6 Computational study

The program coding is done using 'C' language and executed on AMD Athlon XP 2000 processor. Since, SA algorithm involves sampling of random numbers all experiments have been conducted twice using the uniform random numbers $u_1, u_2, u_3, u_4 \ldots$ in the first run and $(1.0-u_1), (1.0-u_2), (1.0-u_3), (1.0-u_4)\ldots$ in the second run, so that these two sets of uniform random numbers are negatively correlated. The non-dominated solutions obtained from the two runs are combined to form the combined front. The combined front is updated to yield the net non-dominated front. Updating refers to deletion of dominated solutions within the combined front.

The proposed PASA algorithm is compared with the existing algorithms namely VR algorithm and modified MOGLS algorithm. First, the non-dominated solutions generated by the PASA algorithm for a set of benchmark flow shop scheduling problems of Taillard [26] are compared with the results presented by Varadharajan (2003, personal communication) and the results obtained using the modified MOGLS algorithm. The results of comparison in terms of the average NFCR is presented in Table 3. It indicates that the modified MOGLS and PASA perform better than VR algorithm. Therefore, the performance of PASA for solving JSPs is compared with the modified MOGLS algorithm. Eighty-two benchmark JSP instances provided by OR-library (www.mscmga.ms.ic.ac.uk) under various classes are solved using the PASA and the modified MOGLS algorithms and the results are compared. The average NFCR is found to be 1.0 for all JSPs except the instance FT06. The superior performance of the PASA can be attributed to its acceptance mechanism.

The extreme solutions obtained by the PASA are presented and the results are then compared with the corresponding upper bound reported in the literature. Different authors for different problems reported the best UB for the makespan (see http://www.uni-weimar.de/∼henning3). The best UB for the mean flow time is computed from the total flow time re-

Table 10. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Fisher and Thompson (1963)

	FT06 ($n=6$, $m=6$)		FT10 ($n=10$, $m=10$)		FT20 ($n=20$, $m=5$)	
	mk	*mft*	*mk*	*mft*	*mk*	*mft*
1	55	50.170	938	789.000	1165	778.150
2	57	49.500	939	786.900	1167	776.400
3	58	46.670	944	785.600	1173	747.850
4	60	45.000	945	785.300	1175	722.500
5	64	44.170	950	784.000	1176	722.250
6			951	783.800	1178	720.650
7			958	782.200	1180	708.600
8			964	778.600	1182	705.750
9			965	775.600	1190	705.200
10			971	767.700	1200	704.900
11			972	766.800	1202	704.200
12			975	766.300	1204	703.450
13			1010	759.700	1207	700.300
14			1024	758.200	1210	696.500
15			1035	756.300	1227	696.300
16			1048	755.700	1233	696.300
17			1056	754.000	1234	694.650
18			1068	753.000	1275	694.200
19			1112	750.500	1278	692.000
20					1281	691.450
21					1284	690.250
22					1289	685.550

Table 11. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Lawrence (1984)

	LA01 ($n=10$, $m=5$)		LA02 ($n=10$, $m=5$)		LA03 ($n=10$, $m=5$)		LA04 ($n=10$, $m=5$)		LA05 ($n=10$, $m=5$)	
	mk	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*
1	666	495.300	655	491.000	597	528.600	590	487.100	593	422.700
2	677	494.000	660	489.100	598	526.500	594	484.700	600	416.500
3	678	491.100	663	472.100	603	484.300	598	465.000	605	416.200
4	679	489.300	669	459.500	606	478.500	605	456.300	606	412.100
5	682	489.100	687	458.000	614	477.300	608	455.600	643	409.100
6	684	488.100	694	457.700	618	472.300	610	448.900	648	408.300
7	691	485.800	699	457.600	619	459.800	616	446.200	656	407.200
8	751	484.700	709	457.600	627	459.500	630	445.100		
9	766	483.300	714	453.300	628	450.200	636	440.000		
10	937	483.200	729	448.300	631	445.200	648	434.800		
11			747	447.900	632	444.700	652	432.300		
12			867	446.800	636	442.900	655	426.000		
13					638	440.800	686	425.900		
14					640	427.500				
15					665	425.300				
16					672	422.700				
17					677	421.100				
18					684	418.100				
19					689	417.500				

Table 12. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Lawrence (1984)

	LA06		LA07		LA08		LA09		LA10		LA11		LA12		LA13		LA14		LA15	
	($n = 15$, $m = 5$)		($n = 15$, $m = 5$)		($n = 15$, $m = 5$)		($n = 15$, $m = 5$)		($n = 15$, $m = 5$)		($n = 20$, $m = 5$)		($n = 20$, $m = 5$)		($n = 20$, $m = 5$)		($n = 20$, $m = 5$)		($n = 20$, $m = 5$)	
	mk	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*
1	926	591.200	890	564.930	863	571.800	951	620.07	958	608.4	1222	736.700	1039	620.300	1150	722.100	1292	756.3	1207	750.650
2	961	588.730	891	561.730	865	570.730	966	617.47	965	607.13	1229	736.450	1041	616.900	1159	714.400	1321	755.4	1211	747.900
3	984	585.070	892	561.730	866	569.600	968	617.27	984	604.67	1249	734.150	1063	616.800	1172	711.750			1212	743.050
4	989	584.530	895	559.600	868	568.730	972	612.67	985	598.13	1251	731.050	1070	616.500	1176	705.600			1221	739.850
5	1067	583.270	904	557.270	869	565.800	975	609.87	1042	592.2	1309	730.950	1072	615.900	1203	703.300			1228	738.200
6	1086	582.400	905	555.870	870	564.200			1044	591.47	1310	729.050	1081	610.100	1215	703.000			1251	737.850
7			906	552.600	871	563.270			1050	588.8	1318	723.800	1112	607.450	1219	701.300			1252	734.550
8			968	550.330	878	562.270			1053	588.07			1138	606.200	1230	699.750			1257	733.300
9			1038	546.470	883	552.930							1160	605.250	1247	698.300			1279	732.950
10			1106	542.800	887	549.470									1258	697.400			1283	732.650
11					894	547.330									1266	696.850			1307	730.050
12					900	544.470									1299	696.400			1336	728.100
13					908	543.400									1309	691.650				
14					909	543.200														
15					910	542.730														
16					913	538.730														
17					940	533.870														

Table 13. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Lawrence (1984)

	LA16		LA17		LA108		LA19		LA20		LA21		LA22		LA23		LA24		LA25	
	($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)	
	mk	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*	*mk*	*mft*
1	945	863.500	784	734.200	848	762.200	842	752.700	907	798.000	1055	920.730	927	852.530	1032	877.930	945	876.130	988	900.470
2	946	825.300	785	724.500	853	751.300	849	744.600	911	773.100	1056	908.870	930	846.400	1038	876.070	951	842.000	990	895.330
3	967	823.200	786	722.500	857	748.800	850	731.900	912	772.600	1058	908.000	932	844.930	1045	870.600	952	828.870	992	890.530
4	975	806.300	787	715.000	861	728.500	856	722.700	914	770.000	1060	903.730	946	844.870	1051	870.270	954	827.930	994	879.730
5	979	806.200	790	714.600	871	726.400			915	758.500	1062	897.870	947	839.000	1052	869.930	958	817.930	996	853.800
6	980	801.600	797	703.000	886	715.400			945	757.500	1064	891.270	953	837.800	1053	867.000	964	815.470	1000	847.330
7	981	773.800	798	690.900	954	713.000			946	752.500	1065	890.200	958	837.670			967	814.000	1001	843.070
8	984	763.300	802	689.500	967	711.200					1070	886.800	966	832.200					1003	839.270
9	995	762.500	808	689.200	1007	708.000					1071	885.600	995	831.270					1005	837.070
10	996	755.200	810	679.700							1120	880.870	1020	825.070					1011	834.270
11	998	749.500	831	677.800							1126	880.130	1024	824.130					1023	824.670
12	1004	748.900	835	675.100							1137	878.330	1031	820.200					1029	824.130
13	1024	748.500	843	666.700							1140	877.670	1041	814.330					1030	823.930
14	1028	748.400	865	660.900							1146	870.330	1047	812.800					1050	823.730
15	1043	746.300	1040	659.200							1147	869.200							1053	822.600
16	1047	743.500	1069	656.400							1148	869.130								
17											1152	866.070								
18											1156	863.670								
19											1179	850.930								
20											1192	846.730								
21											1213	841.400								
22											1314	840.530								

Table 14. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Lawrence (1984)

	LA26		LA27		LA28		LA29		LA30		LA31		LA32		LA33		LA34		LA35	
	($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 10$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)		($n = 15$, $m = 10$)	
	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft
1	1218	1077.100	1264	1110.750	1225	1077.500	1195	1049.100	1355	1063.900	1784	1301.370	1850	1454.070	1719	1319.470	1721	1388.830	1888	1345.130
2	1220	1073.400	1277	1108.000	1227	1076.100	1196	1041.750	1357	1063.350			1852	1453.370	1721	1311.300	1724	1384.400	1919	1344.030
3	1221	1069.550	1280	1099.850	1236	1069.850	1202	1038.950	1359	1053.550			1855	1453.330	1725	1295.870	1725	1384.030	1922	1341.570
4	1228	1069.550	1281	1097.500	1240	1067.800	1208	1012.550	1360	1052.500			1856	1444.030	1737	1289.730	1729	1380.470	1924	1341.400
5	1232	1068.700	1282	1094.050	1241	1059.300	1210	1011.800	1361	1043.900			1859	1440.800	1749	1289.200	1731	1373.400	1926	1340.600
6	1233	1067.950	1292	1093.450	1247	1059.000	1226	1008.000	1367	1043.300			1863	1423.930	1753	1277.300	1733	1373.230	1942	1338.530
7	1234	1066.250	1293	1072.700	1248	1058.900	1228	1007.250	1369	1028.450			1876	1422.300	1754	1273.670	1735	1372.130	1973	1336.230
8	1239	1065.800	1299	1067.300	1251	1051.400	1278	1001.100	1370	1019.950			1881	1417.470	1761	1271.030	1748	1371.200	1985	1315.500
9	1240	1065.450	1370	1066.650	1252	1048.450	1300	991.350	1403	1019.250			1892	1413.370	1772	1270.200	1749	1370.400		
10	1242	1060.400	1378	1061.350	1260	1041.400	1323	990.950	1421	1019.050			1896	1409.930	1830	1268.000	1750	1364.370		
11	1246	1057.500			1314	1037.850	1330	987.100	1436	1012.300			1908	1399.570	1832	1267.630	1755	1339.930		
12	1248	1049.750			1326	1036.800	1333	974.000					1928	1399.530	1836	1265.970	1764	1336.770		
13	1250	1046.350			1330	1035.050	1375	973.750					1976	1397.930			1796	1320.700		
14	1254	1030.400			1339	1032.100	1379	966.050					1987	1396.670			1830	1320.430		
15	1272	1029.700			1357	1029.950	1406	964.250									1851	1319.400		
16	1314	1025.150			1360	1026.500	1438	963.200									1853	1318.670		
17	1318	1024.250					1468	962.650									1892	1317.200		
18	1325	1011.700					1472	962.550									2023	1310.930		
19	1328	1008.600															2112	1307.600		
20	1361	1006.300															2120	1289.530		
21	1394	999.700																		

Table 15. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Lawrence (1984)

	LA36 ($n = 15$, $m = 15$)		LA37 ($n = 15$, $m = 15$)		LA38 ($n = 15$, $m = 15$)		LA39 ($n = 15$, $m = 15$)		LA40 ($n = 15$, $m = 15$)		ABZ5 ($n = 10$, $m = 10$)		ABZ6 ($n = 10$, $m = 10$)		ABZ7 ($n = 20$, $m = 15$)		ABZ8 ($n = 20$, $m = 15$)		ABZ9 ($n = 20$, $m = 15$)	
	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft
1	1282	1224.270	1422	1272.130	1208	1086.800	1256	1165.600	1241	1110.270	1234	1094.900	943	833.3	682	630.7	700	649.650	713	636.450
2	1283	1202.270	1426	1268.070	1213	1085.530	1257	1165.270	1243	1109.670	1239	1081.100	945	827.2	686	630.45	701	649.550	717	626.850
3	1292	1180.930	1427	1266.600	1214	1082.470	1258	1148.330	1246	1104.330	1242	1062.800	947	822.5	689	621.75	702	649.400	722	625.600
4	1318	1158.870	1439	1261.000	1224	1076.930	1259	1123.330	1249	1104.200	1344	1061.100	950	821.6	693	620.6	706	643.750	725	625.100
5	1321	1157.070	1444	1257.670	1243	1071.670	1264	1118.400	1251	1093.530	1376	1060.900	954	817.6	695	616.8	711	643.150	729	624.050
6	1324	1155.330	1446	1246.870	1246	1061.600	1273	1108.930	1255	1083.600	1386	1060.800	964	811.9	703	612.85	712	637.650	733	623.950
7	1326	1153.670	1447	1240.530	1259	1058.000	1280	1102.200	1303	1076.070	1444	1056.300	971	808	704	611.25	716	634.700	740	622.700
8	1363	1152.130	1453	1239.530	1286	1054.530	1331	1084.130					976	806	707	609.9	717	633.150	743	622.150
9	1370	1151.670	1454	1235.530	1305	1054.130							979	803.4	708	609.15	718	628.600	746	620.900
10	1385	1144.730	1456	1233.730	1306	1052.530							982	803.4	709	608.55	720	627.050	757	615.050
11			1464	1230.270	1307	1049.800							985	787	723	596.25	723	623.750	777	612.200
12			1466	1223.930	1310	1049.000							1001	786.2	725	591.75	725	622.550		
13			1470	1210.070	1389	1048.930							1053	783.6	740	591.15	732	606.050		
14			1483	1209.530	1391	1048.870							1058	780.8			735	605.900		
15			1492	1203.470													760	605.500		
16			1495	1199.130													762	601.950		

Table 16. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Applegate and Cook (1991)

	ORB01 ($n = 10$, $m = 10$)		ORB02 ($n = 10$, $m = 10$)		ORB03 ($n = 10$, $m = 10$)		ORB04 ($n = 10$, $m = 10$)		ORB05 ($n = 10$, $m = 10$)		ORB06 ($n = 10$, $m = 10$)		ORB07 ($n = 10$, $m = 10$)		ORB08 ($n = 10$, $m = 10$)		ORB09 ($n = 10$, $m = 10$)		ORB10 ($n = 10$, $m = 10$)	
	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft
1	1059	824.7	889	789.600	1022	888.900	1024	970.800	889	781.900	1013	839.800	397	351.800	906	825.700	934	799.900	944	803.400
2	1072	806	891	780.800	1023	884.400	1025	887.400	890	768.000	1016	834.500	398	346.800	911	745.400	937	797.400	951	787.700
3	1095	804.7	903	768.500	1026	872.100	1031	873.700	891	744.700	1021	831.600	399	343.400	913	732.700	939	795.800	982	786.200
4	1211	802.3	911	765.800	1029	868.500	1032	866.900	896	738.500	1034	828.800	401	340.400	921	731.400	943	778.400		
5			925	762.300	1036	868.500	1053	866.000	898	734.900	1046	825.100	402	339.800	922	731.000	952	746.300		
6			926	750.800	1038	856.200	1057	860.400	899	731.300	1053	820.600	408	336.600	923	724.000	975	746.100		
7			933	745.500	1041	814.100	1060	841.700	904	726.500	1057	820.500	411	333.300	937	702.900	978	743.200		
8			965	742.600			1065	832.300	905	722.800	1074	816.800			947	701.600	1056	741.800		
9			971	741.700			1066	823.400	907	722.100					988	695.800	1059	741.100		
10			986	741.500			1082	822.500	908	718.700							1078	738.200		
11			1056	740.600			1086	819.300	914	713.400							1082	738.000		
12			1084	738.800			1197	817.100	921	709.400							1086	736.100		
13									932	703.000										
14									942	702.700										
15									953	702.000										
16									955	701.400										
17									976	700.800										
18									978	699.800										

Table 17. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Yamada and Nakano (1992).

	YN01 ($n = 20$, $m = 20$)		YN02 ($n = 20$, $m = 20$)		YN03 ($n = 20$, $m = 20$)		YN04 ($n = 20$, $m = 20$)	
	mk	mft	($n = 20$, $m = 10$)		($n = 20$, $m = 10$)		($n = 20$, $m = 10$)	
1	920	871.550	956	903.100	948	878.250	1022	922.950
2	922	861.800	960	901.800	949	875.100	1135	922.150
3	923	848.700	962	894.750	950	874.450		
4	958	847.600	981	893.350	954	872.250		
5	966	846.500	984	889.350	961	863.450		
6	969	844.050	991	887.850	963	861.700		
7	976	842.400	992	887.700	983	858.150		
8	988	842.350	1001	887.400	993	855.350		
9	1018	840.950	1006	885.400	995	855.300		
10			1009	881.450	1009	854.200		
11					1027	851.750		

ported by Henning [45]. The mean relative percentage increase (MRPI) in makespan and mean flow time yielded by PASA with respect to the upper bound is presented in Tables 4 to 9. It is observed that the extreme solutions of non-dominated fronts generated by PASA are very close to extreme solutions of the corresponding Pareto front. Net nominated fronts obtained for different problem size are presented in Tables 10 to 20.

7 Summary

The problem of job shop scheduling is solved with the objectives of minimizing the makespan and the mean flow time of jobs and presented a multi-objective simulated annealing algorithm called Pareto archived simulated annealing (PASA). The proposed algorithm made use of both Pareto dominance and a simple aggregating function to accept the candidate solution among the neighbourhood set of solutions generated by the segment random insertion (SRI) neighbourhood structure. An archive is created, maintained and updated during successive iterations to preserve non-dominated solutions identified during the search. Two simple quality measures namely, net front contribution ratio and mean relative percent increase are used to

Table 18. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Storer et al., (1993)

	SWV01 (n=20, m=10)		SWV02 (n=20, m=10)		SWV03 (n=20, m=10)		SWV04 (n=20, m=10)		SWV05 (n=20, m=10)	
	mk	mft	($n = 20$, $m = 10$)		($n = 20$, $m = 10$)		($n = 20$, $m = 10$)		($n = 20$, $m = 10$)	
1	1473	1101.350	1479	1124.050	1474	1184.550	1510	1173.750	1484	1139.550
2	1474	1053.150	1489	1120.150	1479	1157.700	1516	1141.300	1503	1135.550
3	1483	1043.850	1490	1115.350	1480	1151.150	1524	1131.600	1513	1134.350
4	1488	1042.650	1491	1063.500	1483	1150.550	1533	1126.150	1520	1131.800
5	1489	1040.600	1494	1052.900	1484	1143.200	1539	1122.550	1521	1131.200
6	1505	1039.800	1497	1052.500	1490	1142.900	1554	1121.800	1535	1124.650
7	1511	1038.250	1548	1046.800	1494	1135.700	1557	1121.350	1536	1119.200
8	1516	1034.250	1559	1042.950	1498	1133.850	1568	1121.300	1539	1118.900
9	1518	1025.200	1579	1035.400	1506	1117.950	1577	1120.400	1553	1118.750
10	1545	1018.950	1583	1031.850	1512	1101.150	1593	1117.350	1562	1117.100
11	1560	1012.600			1539	1100.100	1596	1112.650	1563	1115.200
12	1567	1008.750			1550	1099.750	1599	1104.150	1565	1114.100
13	1599	1008.450			1556	1099.400			1585	1100.300
14	1622	1008.300			1561	1088.900			1595	1097.300
15					1591	1088.550			1599	1096.200
16					1597	1085.000			1601	1095.150
17					1599	1084.900			1711	1094.800
18					1604	1082.550				
19					1608	1080.500				
20					1614	1077.400				

Table 19. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Storer et al., (1993)

	SWV06 ($n = 20$, $m = 15$)		SWV06 ($n = 20$, $m = 15$)		SWV07 ($n = 20$, $m = 15$)		SWV08 ($n = 20$, $m = 15$)		SWV09 ($n = 20$, $m = 15$)		SWV10 ($n = 20$, $m = 15$)		SWV10 ($n = 20$, $m = 15$)	
	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft	mk	mft
1	1806	1584.000	1916	1466.500	1736	1549.500	1914	1573.850	1798	1639.550	1887	1693.950	2027	1504.900
2	1808	1581.650	1927	1466.500	1739	1537.950	1957	1560.650	1806	1597.650	1895	1639.850	2030	1500.750
3	1810	1577.950	1934	1465.700	1740	1512.450	1958	1551.350	1809	1590.600	1899	1613.700	2044	1494.500
4	1812	1523.750	1936	1465.050	1741	1512.100	1971	1551.200	1810	1589.200	1913	1609.400	2083	1492.450
5	1816	1523.650	1942	1460.850	1742	1497.050	1974	1545.200	1813	1560.150	1927	1606.750	2110	1462.200
6	1824	1521.450	1951	1459.650	1746	1474.400	1978	1544.800	1815	1535.650	1933	1593.300	2113	1459.150
7	1826	1521.150	1960	1453.950	1760	1473.100	1979	1538.150	1822	1478.900	1939	1592.750	2116	1455.700
8	1827	1519.250	1996	1453.200	1772	1449.000	1983	1516.600	1828	1476.500	1944	1591.700		
9	1828	1518.200	1998	1452.650	1779	1436.200	1985	1503.500	1837	1475.100	1945	1587.250		
10	1839	1517.100	2052	1451.750	1800	1421.200	1986	1494.350	1866	1456.700	1946	1586.400		
11	1853	1514.700	2059	1438.950	1835	1420.750	1991	1491.500	1867	1448.850	1949	1585.650		
12	1855	1511.750	2069	1430.300	1839	1412.550	2040	1485.000	1886	1448.600	1952	1585.400		
13	1869	1509.800	2108	1430.150	1841	1398.000	2041	1483.350	1906	1440.250	1953	1575.250		
14	1870	1509.700	2133	1430.100	1855	1371.650			1916	1434.100	1969	1537.350		
15	1877	1487.100			1870	1356.500			2029	1433.100	1971	1536.600		
16	1882	1486.850			1892	1354.650			2031	1432.700	1974	1536.100		
17	1887	1485.750							2033	1429.550	1975	1531.400		
18	1907	1485.450									1976	1530.300		
19	1912	1484.600									1994	1512.850		
20	1914	1470.450									2025	1512.850		

compare the quality of non-dominated fronts obtained by different algorithms and the effectiveness of the Pareto search by PASA respectively. It has been found that the performance of PASA is better compared to other algorithms considered in this paper. The superior performance of the PASA can be attributed to its acceptance mechanism used to accept the candidate solution. The non-dominated set of solution generated is quiet useful to any decision maker. From the available set of non-dominated solutions, the decision maker can choose the final solution that satisfy the required objectives based on the conditions existing in the shop floor at the time of decision making. The proposed PASA can handle any number of objectives because the non-dominated sorting and weight vector can be extended to any number of objectives.

Table 20. Net non-dominated front obtained by PASA for the benchmark JSP instances proposed by Storer et al., (1993)

	SWV11		SWV12		SWV13		SWV14		SWV15		SWV16		SWV17		SWV18		SWV19		SWV20	
	($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)		($n = 50$, $m = 10$)	
	mk	mft	($n = 20$, $m = 10$)		($n = 20$, $m = 10$)		($n = 20$, $m = 10$)		($n = 20$, $m = 10$)		mk	mft	mk	mft	mk	mft	mk	mft	mk	mft
1	3233	2200.260	3276	2163.880	3295	2208.860	3126	2140.680	3146	2060.680	2924	1964.560	2794	1915.920	2852	1890.240	2843	2034.100	2823	1888.540
2	3241	2162.760	3280	2140.320	3296	2207.940	3136	2131.940	3169	2060.160	2931	1961.400	2797	1908.500	2854	1889.540	2849	2021.740	2824	1885.300
3	3246	2158.640	3294	2137.480	3315	2202.260	3138	2130.780	3191	2043.060	2935	1961.320	2810	1901.360	2879	1888.420	2855	2010.520	2825	1880.420
4	3249	2154.100	3299	2129.040	3326	2197.960	3143	2125.360	3202	2015.200	2939	1961.000	2819	1899.200	3342	1844.440	2879	2010.320	2827	1875.040
5	3256	2153.320	3303	2119.700	3328	2197.140	3146	2110.460	3205	2009.220	2942	1960.460	2827	1897.560	3344	1839.540	2885	2005.880	2828	1870.660
6	3260	2138.160	3341	2102.940	3329	2187.100	3147	2091.380	3206	2006.380	2944	1956.480	2834	1887.340	3345	1837.620	2892	2004.300	2829	1861.120
7	3266	2123.060	3422	2099.120	3335	2179.640	3157	2071.680	3212	2005.760	2946	1952.960	2853	1879.060	3346	1823.080	2894	2003.240	2872	1861.040
8	3268	2117.760	3424	2091.500	3354	2179.220	3163	2070.600			3026	1950.920	2858	1876.040	3353	1808.120	2896	2002.640	2878	1853.140
9	3303	2093.920	3437	2086.980	3355	2171.700	3170	2062.800			3056	1949.500					2902	1997.000	2909	1852.860
10	3305	2093.540			3357	2165.980	3176	2057.340			3106	1949.080					2913	1996.720	3084	1849.960
11	3316	2079.620			3362	2165.120	3178	2057.180			3119	1948.100					2960	1990.420	3111	1829.340
12	3334	2078.500			3372	2138.240	3196	2020.040			3452	1918.660					2993	1987.980	3117	1813.440
13	3341	2076.280			3382	2136.300	3261	2020.020			3453	1891.260					3719	1964.340	3143	1789.220
14	3373	2075.040			3384	2133.680	3281	2013.220			3455	1890.680					3730	1960.120		
15							3283	2011.540			3475	1889.640					3733	1957.780		
16											3478	1887.760					3741	1955.640		
17											3504	1883.580					3784	1950.420		
18											3517	1857.640					3787	1925.480		

References

1. Fisher H, Thompson G (1963) Probabilistic learning combinations of local job shop scheduling rules. In: Muth JF, Thompson GL (eds) Industrial Scheduling, Prentice-Hall, New York, pp 225–251
2. Convay RW, Maxwell WL, Miller LW (1967) Theory of Scheduling. Addition-Wesley, Boston
3. Baker KR (1974) Introduction to sequencing and scheduling. Wiley, New York
4. Coffman EG (1976) Computes and job shop scheduling theory. Wiley, New York
5. Pinedo M (1995) Scheduling-theory, algorithms and systems. Prentice-Hall, Englewood Cliffs, NJ
6. Pinedo M, Chao X (1999) Operations scheduling; with applications in manufacturing and services Limited-Computer Science Series. McGraw-Hill, Singapore
7. Sule DR (2000)Industrial scheduling. PWS, Boston
8. T'kindt V, Billaut J–C (2001) Multicriteria scheduling problems: A survey. RAIRO Oper Res 35:143–163
9. Mellor P (1966) A review of job shop scheduling. Oper Res Q 17:161–171
10. French S (1982) Sequencing and acheduling - an introduction to the mathematics of the job shop. Wiley, New York
11. Jain AS, Meeran S (1998) Job shop scheduling using neural networks. Int J Prod Res 36(5):1249–1272
12. Nagar A, Haddock J, Heragu SS (1995) Multiple and bi-criteria scheduling, a literature review. Eur J Oper Res 81:88–104
13. Rajendran C (1995) Heuristics for scheduling in flow shop with multiple objectives. Eur J Oper Res 82:540-555
14. Sridhar J, Rajendran C (1996) Scheduling in flow shop and cellular manufacturing systems with multiple objectives - A genetic algorithmic approach. Prod Plan Control 7(3):374–382
15. Ishibuchi H, Yoshida T, Murata T (2003) Balance between genetic search and local search in mementic algorithms for multiobjective permutation flowshop scheduling. IEEE Trans Evol Comput 7(1):204–223
16. Daniels RL, Chambers RJ (1990) Multi-objective flow shop scheduling. Naval Res Logist Q 37:981-995
17. Bagchi TP (1999) Multiobjective scheduling by genetic algorithms. Kluwer, Boston, MA
18. Suresh G, SahuS (1993) Multiobjective facility layout using simulated annealing. Int J Prod Econ 32:239-254
19. Schaffer JD (1985) Multiobjecive optimization with vector evaluated genetic algorithms. Proc First ICGA, pp 93–100
20. Fonseca CM, Fleming PJ (1991) Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. Proc Fifth International Conference on Genetic Algorithms, pp 416–423
21. Srinivas N, Deb K (1995) Multiobjective function optimisation using nondominated sorting genetic algorithms. IEEE J Evol Comput 2(2):221-248
22. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) Fast and elitist multiobjective genetic algorithm: NSGA II . IEEE J Evol Comput 6(1):182–197
23. Chang PC, Hsieh JC, Lin SG (2002) The development of gradual priority weighting approach for the multi-objective flow shop scheduling problem. Int J Prod Econ 79:171–183
24. Ishibuchi H, Murata T (1998) A multi-objective genetic local search algorithm and its application to flow shop scheduling. IEEE Trans Syst, Man Cybernetics-Part C: Appl Rev 28:392–403
25. Zitzler E, Thiele L (1999) Multi-objective evolutionary algorithm: A comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3:251–257
26. Taillard E (1993) Benchmarks for basic scheduling problems. Eur J Oper Res 64:278–285
27. Framinan JM, Leisten R, Ruiz-Usano R (2002) Efficient heuristics for flow shop sequencing with the objectives of makespan and flow time minimization. Eur J Oper Res 14:559-569
28. Garey MR, Johnson DS (1979) Computers and intractability, a guide to the theory of NP-completeness. Freeman, New York
29. Garey MR, Johnson DS, Sethi R (1976) The complexity of flow shop and job shop scheduling. Math Oper Res 1:117–129
30. Lawler EL, Lenstra JK, Rinnooy Kan AHG (1982) Recent developments in deterministic sequencing and scheduling: A survey. Reidel, Dordrecht pp 35–74
31. Chen P, Bulfin R (1994) Complexity of multiple machine multicriteria scheduling problems. Proc third IERC, Atlanta, GA
32. Lawrence S (1984) Supplement to, resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques. Technical Report, GSIA, Carnegie Mellon Univ
33. Adams J, Balas E, Zawack D (1998) The shifting bottleneck procedure for job shop scheduling. Manage Sci 34:391–401
34. Applegate D, Cook W (1991) A computational study of the job shop scheduling problem. ORSA J Comput 3(1):149–156
35. Storer RH, Wu SD, Vaccari R (1992) New search spaces for sequencing instances with application to job shop instances. Manage Sci 38:1495–1509
36. Yamada T, Nakano R (1992) A genetic algorithm applicable to large-scale job-shop instances. In: Manner R, Manderick B (eds) Parallel instance solving from nature 2, North-Holland, Amsterdam pp 281–290
37. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
38. Aarts EHL, Lenstra JK (eds) (1997) Local search and combinatorial optimization. Wiley, New York
39. Matsuo H, Suh CJ, Sullivan RS (1988) A controlled search simulated annealing method for the general job shop scheduling problem. Working Paper, Graduate School of Business, The University of Texas at Austin, TX
40. Van Laarhoven PJM, Aarts EHL, Lenstra JK (1992) Job shop scheduling by simulated annealing. Oper Res 40(1):113–125

41. Yamada T, Nakano R (1996) Job shop scheduling by simulated annealing combined with deterministic local search. In: Osman IH, Kelly JP (eds) Proc Metaheuristics International Conference, Hilton Breckenridge, Colorado, pp 344–349
42. Kolonko M (1999) Some new results on simulated annealing applied to the job shop scheduling problem. Eur J Oper Res 113(1):123–136
43. Czyzak P, Jaszkiewicz A (1998) Pareto simulated annealing – A metaheuristic technique for multi-objective combinatorial optimizations. J Multicriteria Decis 7(1):34–47
44. Calabrese J, Henley R, Udayabhanu V (2001) Benchmarking of dispatching rules for the job shop flow time problem. Proc First International Conference on Logistics and Supply Chain Management, PSG College of Technology, India 1:205–209
45. Henning A (2002) Practical job shop scheduling problems Dissertaition, Friedrich-Schiller-University Jena, Jena, Germany (in German)
46. Rajendran C, Chaudhuri D (1992) An eficient heuristic approach to the scheduling of jobs in a flow shop. Eur J Oper Res 61:318–325
47. Knowles JD (2002) Local search and hybrid evolutionary algorithms for Pareto optimization. PhD Thesis, Department of Computer Science, University of Reading, UK