ELSEVIER

European Journal of Operational Research 95 (1996) 356-373

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

Genetic algorithms for the two-stage bicriteria flowshop problem

Venkata Ranga Neppalli [a], Chuen-Lung Chen [a], Jatinder N.D. Gupta [b,*]

[a] *Department of Industrial Engineering, Mississippi State University, Mississippi State, MS 39762, USA*

[b] *Department of Management, Ball State University, Muncie, IN 47306, USA*

Received September 1994; revised August 1995

Abstract

This paper considers the two-stage bicriteria flow shop scheduling problem with the objective of minimizing the total flow time subject to obtaining the optimal makespan. In view of the NP-hard nature of the problem, two Genetic Algorithms (GA) based approaches are proposed to solve the problem. The effectiveness of the proposed GA based approaches is demonstrated by comparing their performance with the only known heuristic for the problem. The computational experiments show that the proposed GA based approaches are effective in solving the problem and recommend that the proposed GA based approaches are useful for solving the multi-machine, multi-criteria scheduling problems.

Keywords: Flow shop; Bicriteria scheduling; Genetic algorithms; Comparative evaluations

1. Introduction

Scheduling problems in general are concerned with finding the sequence in which a set of jobs are processed by one or more facilities to minimize a desired performance measure(s) or objective(s). A variety of scheduling problems considering various performance measures such as average flow time, maximum completion time, maximum tardiness, etc., have been addressed in the literature (see, e.g., Tzafestas and Triantafyllakis [38]). Most of the scheduling research has been confined to optimizing a single criterion. However, scheduling decisions quite often involve consideration of more than one criterion and hence require dominant schedules (Ruiz-Diaz and French [32]). Furthermore, combining various objectives may significantly reduce the scheduling cost. Despite their importance, not much attention has focussed on multi-criteria scheduling research, more specifically in the domain of multi-machine scheduling.

The bicriteria scheduling problems, which deal with two criteria C_1 and C_2, are generally divided into two classes. In the first class, the problem involves minimizing one criterion (C_2) subject to the constraint that the criterion C_1 has to be optimized. Criteria C_1 and C_2 are defined as primary criterion and secondary criterion, respectively. In the second class of problems, both the criteria (C_1 and C_2) are considered equally important and the problem involves finding efficient (non-dominated) schedules.

In the first class of problems, the pioneering work can be attributed to Smith [37] who considered the single-machine problem with maximum tardiness as the primary criterion and mean flow time as the secondary criterion. Extensions of Smith's work with the consideration of different primary and secondary

* Corresponding author.

0377-2217/96/$15.00 Copyright © 1996 Elsevier Science B.V. All rights reserved

SSDI 0377-2217(95)00275-8

criterion have been studied (Bianco and Ricciardelli [4], Chand and Schneeberger [7], Chen and Bulfin [8], Emmons [16], Heck and Roberts [21], Miyazaki [27] and Shanthikumar [36]).

In the second class of problems, Van Wassenhove and Gelders [40] extended the problem solved by Smith [37], and developed an algorithm which provides all efficient solutions for the criteria of mean flow time and maximum tardiness. Several extensions of the single-machine scheduling problem with consideration of different criteria have been reported (Bagchi [2], Barnes and Vanston [3], Cenna and Tabucanon [6], Chen and Bulfin [8], Nelson et al. [28], Sen and Gupta [33], Van Wassenhove and Baker [39], and Vickson [42]). A detailed survey of different approaches developed for the single machine bicriteria and multi-criteria scheduling problems can be found in Dileepan and Sen [15] and Ruiz-Diaz and French [32].

As mentioned earlier, not much attention focussed on the multi-machine, multi-criteria scheduling problems. Chen and Bulfin [9] presented a thorough study on the complexity analysis of the multi-machine, multi-criteria problem. Daniels [12] considered the flow shop problem with maximum completion time and maximum tardiness as two equally important criteria. A constructive algorithm was developed for the problem to generate efficient schedules. Rajendran [31] developed a branch and bound approach for the two-stage flow shop problem with makespan as the primary criterion and total flow time as the secondary criterion. He also presented two heuristics for the same problem which iteratively improve the schedule obtained from Johnson's rule.

Ruiz-Diaz and French [32] state that the main reasons for the little attention focussed on multi-criteria scheduling research are: (i) extreme complexity of the problems and (ii) the lack of availability of general approaches. Genetic Algorithms (GA), which are developed based on the mechanism of evolution, demonstrated their potential for solving hard intractable optimization problems. Also, genetic algorithms are general in terms of applicability and are proven to be efficient and adaptive even in case of complex constrained optimization problems. Therefore, application of genetic algorithms may well be suitable to handle the complexity of the multi-criteria scheduling problems.

This paper develops two genetic algorithms based approaches for the two-machine bicriteria flowshop problem with the objective of minimizing the total flowtime subject to the condition that the makespan of the schedule is minimum. The rest of the paper is organized as follows. Section 2 describes the two-stage bicriteria flowshop problem. The basic steps of the genetic algorithms are reviewed in Section 3 where various components of genetic algorithms are briefly discussed. The procedure and effectiveness of the proposed GA based approaches are discussed in Section 4 where results of computational experiments are also presented. Finally, Section 5 concludes the paper with some fruitful directions for future research.

2. The two-machine bicriteria flowshop problem

Following the standard three field notation for scheduling problems, we will represent the two-stage bicriteria flowshop problem as $F2 \mid \mid \Sigma C_i / C_{\max}$ problem where the term $\Sigma C_i / C_{\max}$ implies that the objective is to minimize total flow time subject to the condition that the makespan (also called *maximum completion time*) is minimum. Let the processing times of job j at two machines be represented by a_j and b_j, where $j = 1, 2, \ldots, n$. It is assumed that each job j is first processed by machine 1 and then by machine 2. Before discussing the complexity of the above problem, it is useful to note that the two-machine flowshop problem with makespan objective ($F2 \mid \mid C_{\max}$) can be solved optimally using Johnson's algorithm (1954).

While the $F2 \mid \mid C_{\max}$ problem is polynomially solvable, the $F2 \mid \mid \Sigma C_i$ problem is known to be NP-hard in the strong sense. Using this result, Chen and Bulfin [9] have shown that the $F2 \mid \mid \Sigma C_i / C_{\max}$ problem is NP-hard in the strong sense. In view of this, Rajendran [31] proposed a branch and bound algorithm as well as two heuristic algorithms to solve the above problem. The branch and bound algorithm proposed by Rajendran is computationally expensive and hence cannot be used for solving large-size problems. However, his heuristic algorithms can be used to initialize the GA based approaches. The steps of Rajendran's heuristic 1 used in our investigations are as follows:

Algorithm Rajendran:

Step 1. Let $\sigma=\{\sigma(1), \sigma(2),\ldots, \sigma(n)\}$ be the schedule obtained by using Johnson's rule. Let the job k be the value of u that yields $C(\sigma)$ given by the equation above. Let F represent its total flowtime found by summing the completion time of each job at machine 2. Enter Step 2.

Step 2(a). For each sequence position r not equal to $k-1$, k, or n, calculate the following:
$D_{\sigma(r)}=a_{\sigma(r)}+b_{\sigma(r)}-a_{\sigma(r+1)}-b_{\sigma(r+1)}$
$D'_{\sigma(r)}=2a_{\sigma(r)}+b_{\sigma(r)}-2a_{\sigma(r+1)}-b_{\sigma(r+1)}$.

Step 2(b). Set
$D_{\sigma(k-1)}=D_{\sigma(k)}=D_{\sigma(n)}=-1$.

Step 2(c). Rank order the jobs in descending order of $D(j)$ values above, breaking the ties using the $D'(j)$ values. Let the list thus obtained be $\pi=\{\pi(1), \pi(2),\ldots, \pi(n)\}$. Set $j=1$ and enter Step 3.

Step 3. Find q such that $\sigma(q)=\pi(j)$. If

$$\max\left\{\sum_{i=1}^{q-1} a_{\sigma(i)}+a_{\sigma(j+1)}-\sum_{i=1}^{q-1} b_{\sigma(i)};\ \sum_{i=1}^{q+1} a_{\sigma(i)}-a_{\sigma(j+1)}-\sum_{i=1}^{q-1} b_{\sigma(i)}\right\}$$
$$>\left\{C(\sigma)-\sum b_i\right\},$$

go to Step 5, otherwise enter Step 4.

Step 4. Interchange jobs in the q-th and $(q+1)$st sequence positions. Find the total flow time, F', of the schedule thus obtained. If $F'\geqslant F$, proceed to Step 5, otherwise, let the new schedule obtained be σ, $F=F'$ and return to Step 2.

Step 5. If $j<n$, set $j=j+1$ and return to Step 3; otherwise stop. The current schedule and its total flowtime constitute an approximate solution to the problem.

In this paper, one of the objectives is to explore the possibility of devising a genetic algorithm based solution procedure for a broad range of multi-machine multi-criteria scheduling problems. Hence we considered the application of the genetic algorithms to solve the $F2\,||\,\sum C_i/C_{\max}$ problem in order to gain an insight into the application of GA based approaches to the solution of multi-criteria scheduling problems.

3. Genetic algorithms

Genetic algorithms are probabilistic search techniques (Holland [22]), developed based on the mechanism of evolution (i.e., natural selection). The probabilistic search procedure in GA, combined with reproduction and recombination, mimics the process of evolution. In GA the solution space is generally represented by a population of structures, where each structure, in general, is a possible solution to the problem. From the concept of genetics that *better parents produce better offspring*, new structures (offspring) are generated by applying simple genetic operators such as crossover, mutation, and inversion, to the potential (parent) structures selected from the existing population. The members with higher fitness values in the current population will have higher probability of being selected as parents, which is similar to Darwin's concept of *survival of the fittest*. In every generation, the existing population of parent solutions is replaced with the newly generated population of offspring solutions. This procedure is repeated to perform an adaptive search for an optimal or near optimal solution.

Since their introduction, genetic algorithms have been applied to a wide variety of combinatorial optimization problems (see, e.g. Goldberg [18]) including the well known Travelling Salesman Problem (Goldberg and Lingle [17]) and the Scheduling Problem (Biegal and Davern [5]; Chen et al. [10]; Neppalli et al. [29]; Vempati et al. [41]).

3.1. Basic procedure of genetic algorithms

The following outline is the basic procedure of GA, where the notation POPSIZE is the size of the population, P_c is the probability of crossover, P_m is the probability of mutation, MAX_GENER is the maximum number of generations, $S(t)$ is the population in the t-th generation; $s_i(t)$ is the i-th member in $S(t)$; $f(s_i(t))$ is the fitness value of $s_i(t)$, and SUMFIT(t) is the sum of $f(s_i(t))$ for all $s_i(t)\in S(t)$.

Step 1. Generate the initial population $S(t)$ of size equal to POPSIZE, where $t=0$.

Step 2. Calculate the fitness value, $f(s_i(t))$, for each member of $S(t)$, and the selection probability for each $s_i(t)$, where the selection probability is defined as

$$P(s_i(t)) = \{f(s_i(t))/\text{SUMFIT}(t)\}.$$

Step 3. Select a pair of members (parents) that will be used for reproduction via the selection probability. Apply genetic operators (cross over and mutation) to the parents based on the probability of cross over (P_c) and probability of mutation (P_m) to generate two new offspring.

Step 4. Insert the generated offspring into a new population, $S(t+1)$, for generation $t+1$. If the size of the new population is equal to POPSIZE, then go to Step 5, else go to Step 3.

Step 5. Check for termination. If current generation, $t+1$, is equal to MAX_GENER, or the population is converged, then stop, else go to Step 2.

According to the above outline, several components have to be determined before implementing GA. These components include initial population, selection probability, genetic operators, termination criteria, and parameters. In general these components affect the performance of GA and are problem dependent. Therefore, analyzing these components is vital for enhancing the performance of GA for the candidate problem. In the remaining subsections, we analyze the above components of GA in order to evolve an optimized approach for solving the candidate problem.

3.2. Initial population

The performance of GA is influenced by the seeding of initial population (Grefenstette [20]). In this research, we analyzed the effect of initial population by comparing three different procedures: (i) a complete random initial population (referred as RIP), (ii) 50% of the population generated by mutating Johnson's sequence (which is the optimal sequence for makespan criterion) and the other 50% of the population generated randomly (referred to as JIP), and (iii) 50% of the population generated by mutating Rajendran's heuristic solution (which is the only heuristic solution for the present bicriteria problem) and 50% of the population generated randomly (referred to as HIP). It should be noted that by using RIP, the feasibility of the final solution (regarding obtaining the optimal makespan) is not guarenteed. In this application we used a method called *Elitist Strategy* which saves the best solution (the solutions with minimum makespan and least total flow time) in every generation. Hence, the feasibility of the final solution (regarding obtaining the optimal makespan) in both JIP and HIP procedures is guarenteed since the initial population of GA is initialized with a feasible (Johnson's or Rajendran's) solution. The reason for considering RIP in the comparative analysis is to investigate the effectiveness of this procedure compared to JIP and HIP since in JIP and HIP there exists a possibility for over-domination of heuristic solutions in the initial generations. So, we implemented RIP in order to justify heuristic initialization. Whenever the solution obtained by RIP is infeasible (without the optimal makespan) we used Johnson's sequence as the final solution for comparative analysis. It should be noted that in very few cases among the total test problems, infeasible solutions were obtained using RIP even though its performance did not dominate the performance of JIP or HIP (a detailed analysis is presented in the next section).

From the comparative analysis it was found that the performance of HIP dominates the performance of JIP and RIP in both the GA based approaches. Hence, we extended the analysis to investigate the effect of different fraction of heuristic solutions in the initial population. We considered five different procedures for generating the initial population with different fractions of heuristic solutions. In the first procedure, Rajendran's heuristic solution is included in the complete random initial population. In the other four procedures different fractions (25%, 50%, 75% and 100%) of the initial population are generated by mutating Rajendran's heuristic solution with the rest of the fraction of initial population generated randomly.

3.3. Selection probability

According to the basic concepts of GA, the selection probability of a member in the population should reflect the member's effectiveness (performance measure) in solving the problem. The method used in the above (basic) procedure of GA is a general method for calculating the selection probability. However, an appropriate scaling of the fitness of the members of the population may control the bias towards exploration and exploitation of the genetic search (Goldberg [18]). In this research, we consider three general scaling mechanisms, Linear, Sigma Truncation, and Power Law.

In the *linear scaling procedure*, the raw fitness is scaled by using the equation

$$f' = a^{*} f + b,$$

where f' is the scaled fitness value, f is the raw fitness value and a and b are constants. The values of a and b are determined in such a way that the average of the scaled fitness is equal to the average of the raw fitness. This scaling procedure increases the fitness values of the members whose fitness values are larger than the average fitness value of the population and decreases the fitness of the members whose fitness values are smaller than the average fitness value of the population. Linear scaling mechanism is quite efficient, however, in some cases the procedure yields negative fitness values. Therefore, in order to avoid the occurrence of negative fitness values, using the similar concept, we increased or decreased the raw fitness of a member in proportion to its deviation from the average fitness of the population. If the raw fitness of a member is less than the average fitness of the population, then its scaled fitness will be reduced by a fraction proportional to its deviation from average fitness of the population. Similarly, if the raw fitness of a member is greater than the average fitness, then its scaled fitness is increased proportionately.

In the *sigma truncation scaling procedure*, along with the average fitness of the population, standard deviation of the fitness values is also considered to eliminate certain bad members of the population. The fitness of each member is scaled as follows:

$$f' = f - (\alpha - C\sigma),$$

where f' is the scaled fitness value, f is the raw fitness value, α is the average fitness, C is the truncation factor, and σ is the standard deviation of the population. In the above equation, the negative scaled fitness is set to zero. In this scaling mechanism the enhancement of the fitness value above and below average fitness is dependent on the standard deviation of the population.

In the *power factor (law) scaling procedure* the raw fitness value is scaled using the equation

$$f' = f^{\mathrm{kfac}},$$

where f' is the scaled fitness value, f is the raw fitness value, and kfac is the power factor. The value of kfac is chosen between 1 and 3. It is evident that the scaling procedure enhances the raw fitness by having the power factor kfac > 1.

3.4. Genetic operators

There are three basic operators used in GA: crossover, mutation, and inversion. Among these three, crossover and mutation are the commonly used operators. Cleveland and Smith [11], analyzed several crossover operators for scheduling problems, and based on their analysis we used Goldberg's PMX operator as the crossover operator in this application. For a detailed description of the PMX operator refer to [11]. Also, we used a random swap operator (which randomly chooses two positions in a schedule and swaps the jobs in these two positions) as the mutation operator.

3.5. Termination criteria

Generally, in most of the GA applications, the termination criterion is based on the number of generations. However, problems such as premature convergence may be avoided by having the termination criteria based on the convergence of the population. Measures were proposed [20] to evaluate the convergence of the population which can be suitably adapted in the termination criterion. In this approach, by using the termination criteria based on convergence, beneficial aspects are investigated. Specifically the following termination criteria are compared:

Criterion 1: *Number of generations*. If the number of generations exceeds 200, terminate the search.

Criterion 2: *Hypermutation*. If there is no improvement in the best solution so far for the last 25 generations, then increase the probability of mutation (hypermutation), and continue the search with the number of generations, i.e. 200, as the termination criterion. In this termination criterion the purpose of increasing the probability is to diversify the population of GA.

Criterion 3: *Entropy of the population.* The entropy measure proposed in [20] can be used to estimate the convergence of the population. Hence we can use a threshold value of the entropy measure below which the population of GA will be diversified by increasing the mutation rate. The entropy measure of the population is measured using the following equation:

$$\text{Entropy} = \sum_{j=1}^{n} H(i)/n,$$

where

$$H(i) = \sum_{j=1}^{n} \frac{1}{\text{Log } n} \left\{ \tfrac{1}{2} n_{ij} * \text{POPSIZE} \times \text{Log}\left(\tfrac{1}{2} n_{ij} * \text{POPSIZE} \right) \right\}.$$

Here, n_{ij} is the number of times job i is preceded by job j in the total population. When the population converges, the entropy value approaches zero. A threshold value for the entropy measure is used as the criterion for increasing the mutation rate in order to diversify the population. Hence, if the population entropy is less than the threshold value, the probability of mutation is increased to 0.9. Thus, in every generation the population is subjected to entropy evaluation and the mutation rate is increased if the entropy falls below the threshold value.

3.6. Parameters

Determination of the values of the parameters of GA is a complex process. In most of the applications the parameter values of GA are tuned based on some trial examples. There are two approaches proposed in the literature for the design of parameters of genetic algorithms (DeJong [14] and Grefenstette [19]) along with an extensive study on the effect of different parameters on the performance of GA (Schaffer [35]). In this research, due to the complexity of the specific application of GA, we analyzed the effect of population size, crossover rate, mutation rate and empically determined the population size, probability of crossover and mutation to be 1.00, 0.90, and 0.10, respectively, with several trial runs with a number of combinations of the parameter values. In all the above empharical comparisons we used the *Elitist Strategy* which saves the best solution of the current population to the population of the next generation. Also, the number of generations is fixed at 200 and the performance of the approaches using different combinations of parameter values are evaluated in terms of the CPU time requirements and the quality of solutions.

4. The GA based approaches

In applying the genetic algorithms to bicriteria scheduling problems like the $F2 \| \Sigma C_i / C_{\max}$ problem, we consider two approaches. In the first approach, the concept of Vector Evaluated GA proposed by Schaffer [34] is adapted (which will be referred to as *Vector Evaluated Approach*). In this approach, the concept is based on combining fittest solutions of different criteria to produce efficient solutions. By having sub-populations, one for each criterion (makespan and total flow time), fittest solutions of the two criteria are combined to obtain efficient solutions. In the second approach, a linear combination of the two criteria is considered (which will be referred to as *Weighted Criteria Approach*). The weighted sum of makespan and total flow time is used as the fitness for each structure in the GA population. It is evident that both the above approaches are general in terms of their applicability. With reasonable modifications, both the approaches can easily deal with other multi-criteria scheduling problems.

4.1. Vector evaluated approach

Genetic algorithms search with a population of solutions rather than a single solution, which enables them to accomplish structured recombination. Using the advantage of the recombination procedure, by combining solutions of different criteria, Schaffer [34] proposed an approach for the class of multi-criteria optimization problems. In his procedure, the traditional genetic algorithms are extended by selection of a sub-population for each criterion, and solutions that are better for different criteria are selected and combined using genetic operators to evolve efficient solutions.

In solving the $F2 \mid \mid \Sigma C_i / C_{max}$ problem with the Vector Evaluated approach, the fitness value of a solution in the population is a vector representing both makespan and total flow time. A sub-population (with half the size of the population) for each criterion is generated by selecting the best solutions for the criterion from the current population. Solutions with good fitness values in each sub-population are selected and recombined in each generation to evolve the solutions which minimize the total flow time subject to the constraint that makespan must be optimized.

Based on the above structure, the step in the basic procedure of GA for calculating the fitness value and selection probability for each member in the population, Step 2, has to be modified for the Vector Evaluate Approach as follows:

Step 2. Calculate the fitness value and selection probability for each member of $S(t)$.

Step 2(a). Calculate $u(s_i(t))$ and $v(s_i(t))$ for each member of $S(t)$, where $u(s_i(t))$ and $v(s_i(t))$ are the total flow time and the makespan of the member $s_i(t)$, respectively.

Step 2(b). Generate the sub-population for each of the criteria by filling the sub-population (SPI(t) for total flow time and SP2(t) for makespan) with the best members of the current population $S(t)$, POPSIZE/2, for each criterion. Note that a member with good total flow time and makespan can be assigned to both SP1(t) and SP2(t).

Step 2(c). Find the maximum $u(s_i(t))$, MaxF, in SP1(t) given by

$$\text{MaxF} = \text{MAX}\{u(s_i(t)),\ s_i(t) \in \text{SP1}(t)\}.$$

Calculate the fitness value and selection probability for each member in SP1(t) with fitness value defined as

$$\{\text{MaxF} - u(s_i(t))\}$$

and selection probability

$$p(s_i(t)) = \frac{\text{MaxF} - u(s_i(t))}{\text{SUMF}},$$

where

$$\text{SUMF} = \sum_{i=1}^{\text{POPSIZE}/2} \{\text{MaxF} - u(s_i(t))\}.$$

Step 2(d). Find the maximum $v(s_i(t))$, MaxC, in SP2(t) given by

$$\text{MaxC} = \text{MAX}\{v(s_i(t)),\ s_i(t) \in \text{SP2}(t)\}.$$

Calculate the fitness value and selection probability for each member in SP2(t) with fitness value defined as

$$\{\text{MaxC} - v(s_i(t))\}$$

and selection probability

$$p(s_i(t)) = \frac{\text{MaxC} - v(s_i(t))}{\text{SUMC}},$$

where

$$\text{SUMC} = \sum_{i=1}^{\text{POPSIZE}/2} \{\text{MaxC} - v(s_i(t))\}.$$

Note that in performing the third step, select a pair of parents via the selection probability, all the members in SP1(t) and SP2(t) are considered together. Furthermore, to guarantee the feasibility of the final solution, a method called the *Elitist Strategy* is used, which preserves the best member (i.e., the member with optimal makespan and lowest total flow time) in every generation. (See Fig. 1.)

4.1.1. Computational results for the vector evaluated approach

In this section, we investigate the effect of initial population, scaling mechanism, and termination criterion on the effectiveness of the GA based approach in solving the $F2 \mid \mid \Sigma C_i / C_{max}$ problem. For each of the above components of GA, different procedures were compared. In order to perform a reasonable comparative analysis of different procedures, the following three performance measures were used throughout this application: 1) the Average Rank (referred to as ARank) of the procedure among the test procedures, 2) the number of times the procedure provides the best result among the tested procedures (represented by X), and 3) the Average Relative Deviation (ARD), which is the average of the deviations of the solution quality from the best known

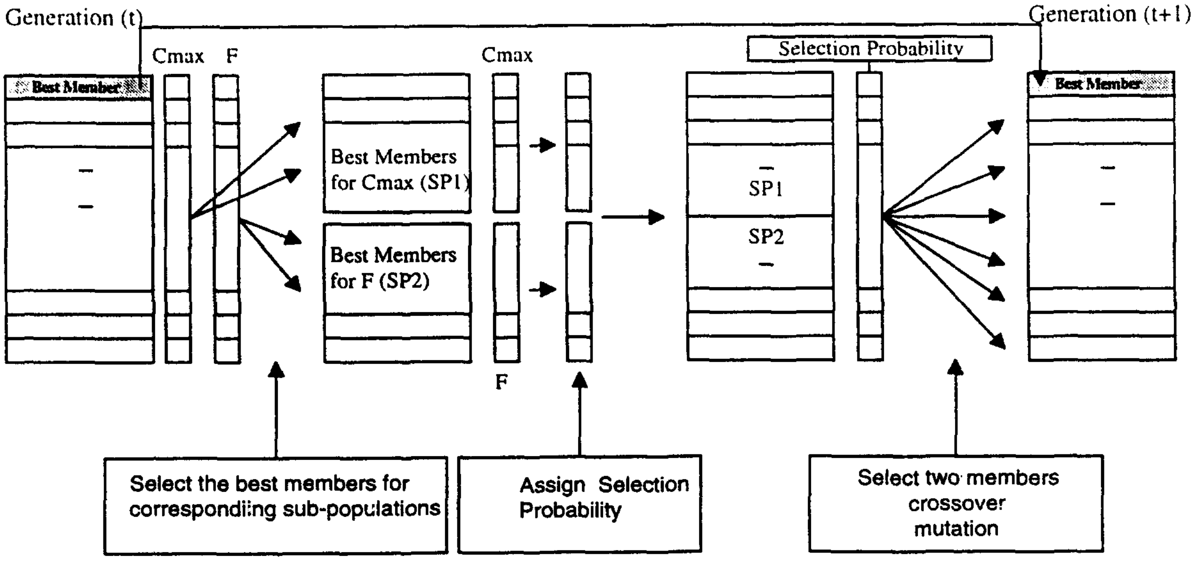

Fig. 1.

solutions for the tested problems. Further details on the above performance measures can be found in [26].

The average rank of approach i is determined by the equation

$$\mathrm{ARank}_i = \sum_{j=1}^{\mathrm{nprobs}} R_{ij},$$

where nprobs is the number of tested problems and R_{ij} is the rank of the approach i for the tested problem j.

The Average Relative Deviation (ARD) of approach i is computed by using the following equation:

$$\mathrm{ARD}_i = \sum_{j=1}^{\mathrm{nprobs}} \frac{F_{ij} - F_j^*}{F_j^*} * 100,$$

where F_{ij} is the flow time provided by approach i for problem j and F_j^* is the best flow time obtained for problem j among the comparing approaches.

The processing times of the two-machine flowshop problems were generated from a uniform distri-

Table 1
Effect of initial population (Vector Evaluated Approach)

n	No. of Problems	RIP			JIP			HIP		
		ARank	X	ARD	ARank	X	ARD	ARank	X	ARD
10	50	1.39	42	0.22	1.40	41	0.17	1.35	43	0.07
15	50	1.84	26	0.98	2.12	18	0.40	1.86	24	0.25
20	50	2.11	18	1.52	2.20	12	0.60	1.64	29	0.10
25	50	2.56	6	3.32	2.00	13	0.55	1.44	33	0.12
30	50	2.55	4	3.14	2.18	6	0.68	1.27	40	0.08
40	50	2.44	4	2.10	2.18	7	0.99	1.31	40	0.21
50	50	2.52	1	2.66	2.25	4	1.17	1.12	45	0.07
60	50	2.56	1	6.80	2.30	4	1.59	1.14	45	0.10
70	50	2.64	2	5.06	2.28	1	1.70	1.08	47	0.11
80	50	2.64	1	4.26	2.30	2	1.75	1.06	47	0.03
Total	500	2.32	105	3.01	2.14	108	0.98	1.32	393	0.12

Table 2
Effect of fraction of heuristic solutions in the initial population (Vector Evaluated Approach)

n	No. of problems	Percentage of the population generated using mutated Rajendran's heuristic solution														
		(Ranom + Raj. Sol.) Pop.			25% Pop.			50% Pop.			75% Pop.			100% Pop.		
		ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP
10	50	1.08	47	0.03	1.02	49	0.01	1.12	46	0.05	1.14	45	0.04	1.10	46	0.05
15	50	1.68	27	0.18	1.76	23	0.19	1.64	29	0.23	1.68	31	0.21	1.52	31	0.12
20	50	2.42	15	0.27	2.28	17	0.23	2.34	18	0.25	2.62	14	0.36	2.90	9	0.41
25	50	3.22	9	0.55	2.38	20	0.30	2.88	11	0.36	2.80	11	0.39	2.74	7	0.32
30	50	2.94	13	0.52	2.68	13	0.28	2.90	10	0.4	2.74	8	0.38	3.06	11	0.44
40	50	2.86	13	0.41	2.86	12	0.40	3.04	11	0.42	3.04	7	0.37	2.78	11	0.43
50	50	3.10	13	0.53	2.72	13	0.28	3.26	6	0.42	2.86	11	0.35	3.00	7	0.38
60	50	3.46	9	0.67	3.08	10	0.53	2.98	6	0.48	2.68	16	0.37	2.66	10	0.43
70	50	3.64	5	0.85	2.58	15	0.44	2.94	10	0.51	2.84	12	0.51	2.92	9	0.50
80	50	3.40	6	0.78	2.56	15	0.38	2.42	16	0.31	3.26	9	0.57	3.28	5	0.56
Total	500	2.78	157	0.48	2.39	187	0.30	2.55	163	0.34	2.57	164	0.36	2.60	146	0.36

bution in the range (1, 30). In the following comparisons the termination criterion is based on the maximum number of generations (MAX_GENER = 200).

The effect of initial population on the Vector Evaluated Approach was analyzed by comparing the three different procedures for generating initial population: RIP, JIP, and HIP. Table 1 presents the comparisons of the three approaches, RIP, JIP and HIP. Column 1 represents the problem size n (number of jobs) and column 2 represents the number of tested problems. From the results in Table 1, it is evident that the seeding of the initial population does affect the performance of the GA based approach. These results also show that HIP is superior to JIP as well as RIP, especially when the number of jobs increases.

As an extension to the conclusions from Table 1, we analyzed the effect of different fractions of heuristic solutions in the initial population. As mentioned earlier we compared five different procedures and the results are presented in Table 2. It is clearly evident that as the fraction of heuristic solutions is increased above 25%, the performance of GA is deteriorating when the average relative performance

Table 3
Effect of scaling mechanisms (Vector Evaluated Approach)

n	No. of problems	No scaling			linear scaling			Sigma truncation			Power law		
		Arank	X	ARD	Arank	X	ARD	Arank	X	ARD	Arank	X	ARD
10	50	1.30	37	0.18	1.16	42	0.04	1.34	37	0.16	1.20	42	0.06
15	50	1.78	21	0.30	1.60	30	0.22	2.42	12	0.74	1.60	28	0.12
20	50	2.14	19	0.36	2.06	16	0.28	2.80	10	1.16	1.70	25	0.18
25	50	2.36	11	0.54	2.04	16	0.28	3.20	5	1.56	1.76	28	0.32
30	50	2.52	13	0.70	2.08	15	0.42	3.34	7	1.68	1.88	18	0.26
40	50	2.88	5	0.78	1.94	17	0.36	3.30	7	1.56	1.78	22	0.22
50	50	2.74	4	0.78	2.02	18	0.42	3.36	6	2.02	1.78	22	0.22
60	50	2.92	2	0.98	1.88	19	0.37	3.40	3	2.01	1.66	27	0.20
70	50	2.96	2	1.10	1.92	19	0.40	3.48	4	2.44	1.64	25	0.15
80	50	3.02	2	1.36	1.84	19	0.38	3.50	4	2.30	1.64	25	0.28
Total	500	2.46	116	0.72	1.85	211	0.32	3.01	95	1.56	1.66	262	0.20

Table 4
Effect of termination (Vector Evaluated Approach)

n	No. of Problems	Termination 1			Termination 2			Termination 3		
		ARank	X	ARD	ARank	X	ARD	ARank	X	ARD
10	50	1.30	41	0.12	1.25	42	0.10	1.05	49	0.02
15	50	1.97	20	0.25	1.85	22	0.20	1.22	45	0.06
20	50	2.17	13	0.28	1.96	18	0.24	1.27	43	0.05
25	50	2.34	11	0.38	2.12	15	0.27	1.42	39	0.02
30	50	2.30	6	0.40	2.20	10	0.40	1.34	40	0.04
40	50	2.06	18	0.20	1.92	22	0.16	1.86	24	0.10
50	50	2.11	17	0.36	1.89	23	0.28	1.78	24	0.12
60	50	2.05	20	0.24	1.95	24	0.22	1.85	25	0.22
70	50	2.13	18	0.28	2.06	20	0.28	1.81	30	0.08
80	50	2.27	12	0.52	2.25	12	0.50	1.48	38	0.10
Total	500	2.08	176	0.30	1.94	208	0.26	1.55	356	0.09

is considered. The results in Table 2 support the point that including heuristic solutions in large fractions in the initial population may cause the loss of diversity (especially for the makespan criterion) since a sub-population is maintained for each criterion.

Based on the conclusions of Tables 1 and 2, we continue analyzing the effect of different scaling procedures by using the procedure HIP to generate the initial population. Table 3 presents the effect of the three scaling mechanisms: linear, sigma truncation, and power law, on the performance of the vector evaluated GA based approach. It is evident that both linear and power law scaling mechanisms dominate no scaling and sigma truncation scaling mechanisms. Power law scaling mechanism slightly dominates linear scaling with its minimum average relative deviation for each problem size. Furthermore, the power law scaling mechanism provided the best result for more than 50% of the total tested problems. Note that the termination criterion used in the comparisons of Table 3 is based on the maximum number of generations.

Table 5
Effect of different sizes (SP1 and SP2) of sub-populations for C_{max} and total flow time criteria (Vector Evaluate Approach)

n	No. of Prob-lems	Different sizes of sub-populations for C_{max} and total flow time criteria														
		SP1 = 20 & SP2 = 80			SP1 = 40 & SP2 = 60			SP1 = 50 & SP2 = 50			SP1 = 60 & SP2 = 40			SP1 = 80 & SP2 = 20		
		ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP
10	50	1.02	49	0.01	1.04	48	0.02	1.06	48	0.02	1.06	47	0.03	1.22	41	0.20
15	50	1.38	36	0.07	1.40	34	0.09	1.44	36	0.06	1.52	33	0.17	1.92	27	0.39
20	50	2.16	23	0.19	2.02	19	0.17	2.06	19	0.17	2.02	25	0.18	2.22	19	0.38
25	50	3.18	7	0.30	2.58	14	0.23	2.66	9	0.36	2.70	13	0.30	2.48	17	0.34
30	50	3.16	8	0.38	3.04	7	0.31	2.72	12	0.45	3.06	4	0.36	2.54	20	0.40
40	50	3.60	3	0.41	3.54	4	0.37	2.74	11	0.31	2.78	7	0.34	2.08	26	0.34
50	50	3.92	1	0.59	3.08	5	0.44	2.82	9	0.37	3.00	6	0.43	2.12	29	0.35
60	50	4.06	3	0.80	3.12	6	0.53	3.22	3	0.49	2.80	4	0.45	1.68	35	0.22
70	50	4.12	0	0.93	3.42	2	0.75	3.00	5	0.56	2.94	3	0.53	1.52	40	0.14
80	50	4.48	0	1.00	3.16	5	0.62	3.02	5	0.54	2.62	5	0.47	1.72	35	0.19
Total	500	3.11	130	0.47	2.64	144	0.35	2.47	157	0.33	2.45	147	0.32	1.95	289	0.29

Following the analysis of scaling mechanisms, we investigate the effect of different termination criteria on the performance of the vector evaluated GA based approach. Table 4 presents the computational results for comparison of termination criteria 1, 2 and 3 under the condition that HIP was used to generate the initial population and power law scaling mechanism was used. The comparisons show that the termination criteria 2 and 3 are useful when the GA prematurely converges, however the performance enhancements will be influenced by the maximum number of generations allowed for genetic search. This can be illustrated by the ARD of termination criterion 1 which ranged from 0.12 to 0.52. By considering all three performance measures, termination criterion 3 clearly is better than both criterion 1 and 2.

In the above analysis we investigated the effect of initial population, different fractions of heuristic solutions in the initial population, scaling mechanisms and termination criteria and we used two sub-populations of equal size for implementing the VEA. Since one of the obejctives of this application is to investigate the applicability of GA based approaches to a broad range of multi-criteria scheduling problems, we extended the above analysis to evaluate the effect of different sub-population sizes for different criteria. In Table 5 we present the computational results of VEA by varying the size of $SP1(t)$ and $SP2(t)$ (the total population size is fixed at 100). In these analyses we implemented VEA with 25% of the initial population generated using Rajendran's heuristic solution, power law scaling and termination criterion based on entropy measure.

The results in Table 5 indicate that having a large sub-population size for the makespan criterion is beneficial. Out of the 500 test problems the approach with (SP1 = 80 & SP2 = 20) provided better results in 289 problems. Since the fraction of the initial population is generated using heuristic solutions with a bias towards having optimal makespan (feasibility of the solution), the above results are justified. However, from Table 5, it is evident that the approach with (SP1 = 40 & SP2 = 60) dominates the approach (SP1 = 80 & SP2 = 20) in terms of average relative performance when the problem size is less than 50, even though the overall performance of the VEA approach with (SP1 = 80 & SP2 = 20) dominates the rest of the procedures.

To evaluate the effectiveness of the above approach, we compared the GA based approach with Rajendran's heuristic, which is the only heuristic for the $F2 \mid \mid \Sigma C_i / C_{\max}$ problem. Based on the recommendation from the above analysis on the effects of initial population, fraction of heuristic solutions in the initial population, scaling mechanisms, termination criteria, and different sizes of sub-populations for the two criteria, we implemented the GA based approach with the following components: 1) 25% of the initial population generated by mutating Rajendran's heuristic solution, 2) the power law scaling mechanism, 3) a termination criterion based on the

Table 6
Performance of GA based (Vector Evaluated) Approach (compared with Rajendran's heuristic)

S. No.	n	No. of Problems	Vector Evaluated Approach			
			Even	Better	Average rel. pref.	CPU (sec)
1	10	50	6	44	2.99	8.59
2	15	50	2	47	2.97	11.04
3	20	50	0	50	4.54	13.48
4	25	50	0	50	4.34	15.52
5	30	50	0	50	4.76	18.84
6	40	50	0	50	4.98	25.58
7	50	50	0	50	5.50	33.90
8	60	50	0	50	6.54	43.58
9	70	50	0	50	8.70	55.22
10	80	50	0	50	9.86	72.18
Total		500	8	491		

entropy measure of the population, and 4) the sizes of the sub-populations (SP1 and SP2) are 80 and 20 respectively. The computational results are presented in Table 6. These results show that the performance of the GA based approach is consistently better than Rajendran's heuristic. The relative improvement provided by the GA based approach ranged from 2.99% to 9.86%, and the performance improved with problem size. Out of the tested 500 problems the VEA provided better results in 491 problems.

4.2. Weighted criteria approach

In solving multi-objective optimization problems, one of the popular approaches is to convert the multi-criteria into a single objective using a weighted sum of the criteria, and then solving the problem as a single criterion problem. The weight assigning aspect of this approach is vital and usually projects the relative importance of each criteria. In applying this approach to the candidate bicriteria problem, a weighted objective function,

$$f=\tfrac{1}{2}n * C_{\max} + \text{TOTFLOW},$$

was defined (where n is the number of jobs and $C_{\max}$ and TOTFLOW represents the makespan and total flow time of a schedule, respectively). By using $\frac{1}{2}n * C_{\max}$ and TOTFLOW in the weighted objective function the procedure attempts to assign equal importance to both the criteria.

Similar to the previous GA based approach, Step 2 in the basic procedure of GA has to be modified for the current approach as follows. (See Fig. 2.)

Step 2″. Calculate the fitness value and selection probability for each member of $S(t)$.

Step 2(a). Calculate $u(s_i(t))$ and $v(s_i(t))$ for each member of $S(t)$, where $u(s_i(t))$ and $v(s_i(t))$ are the total flow time (TOTFLOW) and the makespan ($C_{\max}$) of the member $s_i(t)$, respectively.Let

$$f(s_i(t)) = n * \nu(s_i(t)) + u(s_i(t)).$$

Step 2(b). Find the maximum $f(s_i(t))$, MAXFIT, in $S(t)$ given by

$$\text{MAXFIT} = \text{MAX}\{u(s_i(t)),\ s_i(t) \in \text{SP1}(t)\}.$$

Calculate the fitness value and selection probability for each member in $S(t)$ with fitness value defined as

$$\{\text{MAXFIT} - f(s_i(t))\}$$

and the selection probability

$$p(s_i(t)) = \frac{\text{MAXFIT} - f(s_i(t))}{\text{SUMFIT}}$$

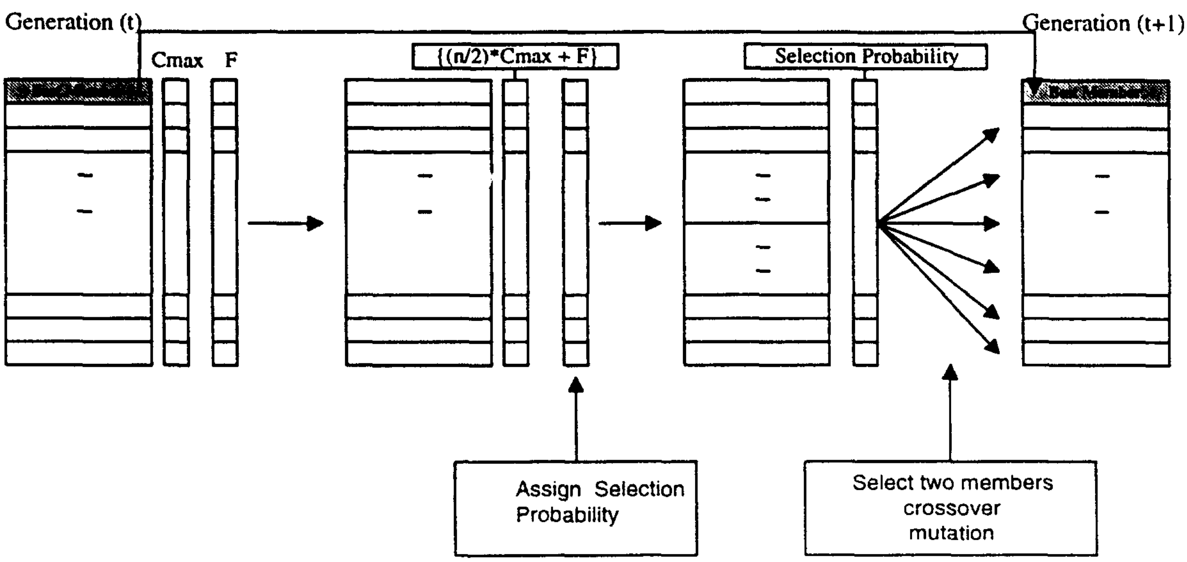

Fig. 2.

Table 7
Effect of initial population (Weighted Criteria Approach)

n	No. of problems	RIP			JIP			HIP		
		ARank	X	ARD	ARank	X	ARD	ARank	X	ARD
10	50	1.22	45	0.22	1.20	44	0.14	1.11	48	0.03
15	50	2.12	19	0.82	2.00	22	0.30	1.65	31	0.15
20	50	2.30	8	2.89	2.28	9	0.74	1.41	35	0.12
25	50	2.38	1	2.20	2.42	8	1.12	1.20	43	0.16
30	50	2.48	2	2.20	2.36	3	1.33	1.16	45	0.15
40	50	2.45	5	3.78	2.40	2	2.17	1.14	43	0.08
50	50	2.40	3	3.14	2.55	0	2.56	1.06	47	0.02
60	50	2.54	1	4.62	2.44	0	3.32	1.02	49	0.01
70	50	2.42	2	3.34	2.42	4	3.29	1.16	45	0.12
80	50	2.54	2	4.52	2.40	0	3.30	1.16	48	0.01
Total	500	2.28	88	2.78	2.24	92	1.81	1.20	434	0.09

where

$$\text{SUMFIT} = \sum_{i=1}^{\text{POPSIZE}} \{\text{MAXFIT} - f(s_i(t))\}.$$

We now investigate the effects of the factors on the performance of the Weighted Criteria GA based approach and determine the conditions which will provide good performance of the approach on the candidate bicriteria problem. The performance of the GA based approach is then evaluated by comparing the results obtained with Rajendran's heuristic.

4.2.1. Computational results for the Weighted Criteria Approach

Similar to the analysis on the previous approach, the three methods for generating the initial population, RIP, JIP, and HIP, were used to estimate the effect of initial population on the performance of the proposed Weighted Criteria Approach. The computational results are presented in Table 7. Similar to the results in the Vector Evaluated Approach, the results in Table 5 indicate that the initial population generated using heuristic solutions does improve the performance of current GA based approach, and HIP

Table 8
Effect of fraction of heuristic solutions in the initial population (Weighted Criteria Approach)

n	No. of problems	Percentage of the population generated using the mutated Rajendran's heuristic solution														
		RIP with Raj. sol.			25% Pop.			50% Pop.			75% Pop.			100% Pop.		
		ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP
10	50	1.10	45	0.05	1.10	46	0.07	1.18	42	0.08	1.16	43	0.10	1.20	43	0.08
15	50	1.66	27	0.20	1.72	25	0.20	1.94	21	0.33	1.66	29	0.20	1.68	28	0.21
20	50	2.16	20	0.24	2.16	24	0.24	2.14	20	0.20	2.06	22	0.19	2.48	20	0.27
25	50	2.68	13	0.21	2.88	8	0.38	2.44	16	0.20	2.58	15	0.29	2.56	16	0.23
30	50	3.02	11	0.33	2.76	9	0.28	2.70	12	0.31	2.66	13	0.28	2.86	13	0.34
40	50	2.76	12	0.24	2.86	10	0.27	3.08	7	0.44	2.86	16	0.43	2.82	11	0.34
50	50	3.02	9	0.32	3.26	9	0.39	3.10	10	0.42	2.58	12	0.30	2.94	10	0.35
60	50	3.12	12	0.49	3.10	8	0.46	2.88	10	0.46	2.80	13	0.36	2.82	8	0.43
70	50	2.62	15	0.32	3.00	9	0.39	3.14	7	0.44	3.08	10	0.43	2.98	10	0.36
80	50	2.94	11	0.53	2.48	16	0.36	3.14	9	0.50	3.42	5	0.53	2.96	9	0.43
Total	500	2.51	175	0.29	2.54	164	0.30	2.57	154	0.34	2.49	178	0.31	2.53	168	0.30

Table 9
Effect of scaling mechanisms (Weighted Criteria Approach)

n	No. of problems	No scaling			Linear scaling			Sigma truncation			Power law		
		ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP
10	50	1.06	47	0.03	1.18	44	0.15	1.18	42	0.08	1.10	45	0.03
15	50	1.44	34	0.08	1.52	29	0.16	2.72	5	0.86	1.48	31	0.12
20	50	1.74	22	0.18	1.72	25	0.22	3.32	2	1.50	1.86	20	0.26
25	50	1.98	18	0.44	1.68	25	0.14	3.66	0	2.12	1.94	15	0.30
30	50	1.98	16	0.48	1.80	21	0.26	3.82	0	2.98	2.06	15	0.36
40	50	2.02	16	0.30	1.72	28	0.30	3.86	0	2.65	2.22	7	0.34
50	50	2.38	5	0.52	1.68	28	0.20	3.98	0	3.26	1.96	17	0.35
60	50	2.26	9	0.60	1.48	33	0.15	3.94	0	3.78	2.20	8	0.62
70	50	2.30	6	0.80	1.44	35	0.18	4.00	0	0.48	2.26	9	0.80
80	50	2.48	6	0.85	1.36	37	0.10	4.00	0	5.45	2.16	7	0.70
Total	500	1.92	179	0.42	1.56	305	0.18	3.45	49	2.75	1.92	175	0.38

clearly dominates both RIP and JIP. Note that similar to Table 1, in the analysis of Table 7 the approach is implemented with no scaling mechanism and termination criteria based on the maximum number of generations.

Following the conclusions from the analysis on the initial population as done for the previous approach, we extended the analysis for estimating the effect of different fractions of the heuristic solution in the initial solution on the performance of the WCA. We used five different procedures (similar to the procedures used in VEA) and the computational results are presented in Table 8. The computational results indicate that the procedure in which Rajendran's heuristic solution is included in the random initial population seems to perform better than the rest of the procedures. Even though the procedure (when 75% of the initial population is generated using the mutated Rajendran's heuristic solution) is comparable, the overall performance of the earlier procedure is consistent throughout the entire problem sizes. Based on the above analysis we generated the initial population randomly and included Rajendran's heuristic solution.

In view of its best desirable effect of the initial population we used the procedure HIP to generate the initial population and continued the analysis on scaling mechanisms and termination criteria. Table 9 presents the computational results of the three scaling mechanisms. The results in Table 9 show that the linear scaling procedure dominates the rest of the procedures, which is different from the conclusion

Table 10
Effect of termination (Weighted Criteria Approach)

n	No. of Problems	Termination 1			Termination 2			Termination 3		
		ARank	X	ARP	ARank	X	ARP	ArgRank	X	ARP
10	50	1.21	45	0.14	1.13	47	0.06	1.08	49	0.01
15	50	1.75	27	0.18	1.53	32	0.18	1.34	40	0.06
20	50	1.90	21	0.20	1.88	23	0.26	1.70	30	0.08
25	50	2.14	15	0.24	1.99	14	0.18	1.87	26	0.12
30	50	2.10	17	0.26	2.08	14	0.26	1.82	27	0.14
40	50	2.06	16	0.22	1.87	22	0.18	2.02	21	0.22
50	50	2.15	16	0.32	1.89	18	0.26	1.86	25	0.17
60	50	2.11	18	0.34	2.05	19	0.32	1.84	28	0.18
70	50	2.18	18	0.36	1.94	21	0.30	1.82	28	0.17
80	50	2.22	12	0.38	2.17	15	0.36	1.60	34	0.12
Total	500	1.98	205	0.26	1.85	228	0.23	1.71	308	0.15

Table 11
Effect of different weights for C_{max} and total flow time (Weighted Criteria Approach)

n	No. of problems	Different weight factors for C_{max} (C) and total flow time (F) criteria														
		C(40%) & F(60%)			C(40%) & F(60%)			C(50%) & F(50%)			C(60%) & F(40%)			C(80%) & F(20%)		
		ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP	ARank	X	ARP
10	50	1.08	47	0.03	1.08	47	0.05	1.02	49	0.02	1.10	47	0.04	1.06	47	0.03
15	50	1.68	29	0.14	1.52	30	0.11	1.44	34	0.09	1.70	29	0.16	1.58	30	0.24
20	50	3.02	9	0.41	1.98	21	0.22	2.62	13	0.32	2.04	20	0.24	2.26	23	0.31
25	50	3.20	7	0.50	2.44	12	0.34	2.48	15	0.29	2.96	11	0.34	2.84	9	0.51
30	50	3.62	5	0.61	2.78	9	0.34	2.82	13	0.29	2.68	13	0.36	2.68	11	0.34
40	50	3.50	6	0.40	2.80	12	0.38	2.58	15	0.40	2.68	10	0.33	3.06	8	0.54
50	50	3.36	10	0.50	2.88	7	0.38	2.54	12	0.46	2.68	13	0.58	3.12	9	0.50
60	50	3.56	9	0.51	2.68	13	0.34	2.86	12	0.38	2.92	7	0.46	2.90	10	0.44
70	50	3.64	9	0.59	3.00	6	0.43	2.68	10	0.32	2.96	11	0.38	2.64	15	0.3
80	50	3.46	8	0.59	3.24	8	0.44	2.66	11	0.30	2.92	11	0.46	2.58	13	0.31
Total	500	3.01	139	0.43	2.44	165	0.30	2.37	184	0.29	2.46	172	0.33	2.47	175	0.35

reached for the Vector Evaluated Approach.

Table 10 presents the computational results comparing the three termination criteria on the Weighted Criteria Approach, which show that the termination criterion based on entropy measure is superior to the other two termination criteria. Note that the analysis in this table is conducted by using the linear scaling mechanism in all three approaches.

The above analysis was performed in an attempt to provide equal priority (weightage) to both criteria in computing the fitness value of a solution. In the following analysis (similar to the analysis performed for VEA by varying the sub-populations) we estimated the effect of having different weight factors for the two criteria. We modified the objective function to

$$w_1 * \tfrac{1}{2}n * C_{max} + w_2 * F,$$

in order to incorporate the weight factors (w_1 and w_2) for the two criteria. We varied the weights of the two criteria and the results are presented in Table 11. From the results it is evident that no single procedure significantly dominates the rest of the procedures. Based on the results, providing equal priority to each

Table 12
Performance of GA based Weighted Criteria Approach (compared with Rajendran's heuristic)

S. No.	n	No. of problems	Weighted Criteria Approach			
			Even	Better	Average rel. perf.	CPU (sec)
1	10	50	6	43	2.98	8.15
2	15	50	4	48	3.22	10.28
3	20	50	0	50	4.55	12.86
4	25	50	0	50	4.24	15.10
5	30	50	0	50	4.60	17.92
6	40	50	0	50	4.60	23.94
7	50	50	0	50	5.08	32.32
8	60	50	0	50	6.02	43.08
9	70	50	0	50	8.12	55.92
10	80	50	0	50	9.35	77.70
Total		500	10	491		

Table 13
Relative performance of VEA and WCA based algorithms

S. No.	n	No. of problems	VEA better	Even	WCA better	WCA/VEA aver. rel. performance
1	10	50	1	41	8	−0.17
2	15	50	5	26	19	−0.25
3	20	50	26	10	14	0.01
4	25	50	32	4	14	0.12
5	30	50	38	0	12	0.18
6	40	50	39	0	11	0.39
7	50	50	40	0	10	0.45
8	60	50	43	0	7	0.57
9	70	50	47	0	3	0.62
10	80	50	43	0	7	0.56
Total		500	314	81	105	

of the criteria ($w_1 = 1.0$ and $w_2 = 1.0$) seems to be promising.

The effectiveness of the Weighted Criteria Approach is then evaluated by comparing its performance with Rajendran's heuristic. Based on the above analysis, we have implemented the weighted criteria GA approach with: 1) random initial population with Rajendran's heuristic solution, 2) linear scaling mechanism, 3) termination criterion using entropy, and 4) equal priority for each criterion (where the weight factors are $w_1 = 1.0$ and $w_2 = 1.0$). Table 12 presents the comparative results of the GA based heuristic (WCA) with Rajendran's heuristic. The improvement provided by the GA based approach over Rajendran's heuristic ranged from 2.98% to 8.96%. Out of the 500 test problems, the WCA based GA approach provided better results in 490 problems, which clearly shows the effectiveness of the approach.

4.3. Comparison of Vector Evaluated Approach and Weighted Criteria Approach

The above computational results show that both the Vector Evaluated Approach (VEA) and Weighted Criteria Approach (WCA) are effective in solving the two-stage bicriteria flowshop problem. However, one of the two approaches may be better than the other. For this purpose we used the VEA and WCA based GA algorithms to solve the same set of problems. In doing so, we have implemented the VEA and WCA based GA algorithms with parameter values found most favorable in our empirical investigations. Thus, for VEA, we implemented the GA based approach with the following components: 1) 25% of the initial population generated using Rajendran's heuristic solution, 2) power law scaling mechanism, 3) termination criterion based on the entropy measure, and 4) sub-population sizes (SP1 = 80 and SP2 = 20). Similarly, for WCA, we implemented the weighted criteria GA approach with: 1) random initial population with Rajendran's heuristic solution, 2) linear scaling mechanism, 3) termination criteria based on entropy, and 4) weight factors ($w_1 = 1.0$ and $w_2 = 1.0$) for two criteria. Table 13 presents the results, where the columns are similar to those in Table 12.

From the results in Table 13, it is evident that the Vector Evaluated Approach is slightly better than the Weighted Criteria Approach. Even though the results favor the VEA over the WCA, the difference in performance is not significant, especially when the average relative performances of both the approaches with respect to Rajendran's heuristic is considered. Hence, either of the two approaches can be used to measure the fitness of a solution in the population without affecting the performance of the GA based approach.

5. Conclusions

Genetic algorithms, within their general framework, have proven to be efficient in single-criterion optimization problems. Extending the traditional concepts of GA, this paper has shown the potential of two GA based approaches for solving the two-stage bicriteria flow shop problem. Computational experience demonstrates that the proposed GA based approaches are quite effective in solving two-stage bicriteria flowshop problems. On a more global level, this paper has shown the manner in which various parameters of the GA based approach can be adapted to effectively solve several bicriteria scheduling problems. Even though the paper concentrated on two-machine flowshop problems, the discussion has been broad enough to be applied to any bicriteria scheduling problem.

Several research projects are worthy of further investigation. We believe that the performance of the two approaches can be improved by incorporating problem specific knowledge in terms of perturbation operators. The proposed approaches can be easily extended to other bicriteria problems when the two criteria are not equally important. However, it may not be as easy to extend the approaches for bicriteria problems with equally important criteria. A research project to modify the approaches for such bicriteria problems may provide a good tool for solving goal integer programming problems and multi-criteria optimization problems.

References

[1] Azizoglu, M., Kondakci, S., and Kirca, O., "Bicriteria scheduling problem involving total tardiness and total earliness penalties", *International Journal of Production Economics* 23 (1991) 17–24.

[2] Bagchi, U., "Simultaneous minimization of mean and variation of flow time and waiting time in single machine systems", *Operations Research* 37 (1989) 118–125.

[3] Barnes, J.W., and Vanston, L.K., "Scheduling jobs with linear delay penalties and sequence dependent setup costs", *Operations Research* 29 (1981) 146–160.

[4] Bianco, L., and Ricciardelli, S., "Scheduling of a single machine to minimize total weighted completion time subject to release dates", *Naval Research Logistics Quarterly* 29 (1982) 151–167.

[5] Biegal, J.E., and Davern, J.J., "Genetic algorithms and job shop scheduling", *Computers and Industrial Engineering* 19 (1990).

[6] Cenna, A.A., and Tabucanon, M.T., "Bicriteria scheduling problem in a job shop with parallel processors", *International Journal of Production Economics* 25 (1991) 95–101.

[7] Chand, S., and Schneeberger, H., "A note on the single machine scheduling problem with weighted completion time and maximum allowable tardiness", *Naval Research Logistics Quarterly* 33 (1986) 551–557.

[8] Chen, C.L., and Bulfin, R.L., "Complexity of single machine, multi-criteria scheduling problems", *European Journal of Operational Research* 70 (1993) 115–125.

[9] Chen, C.L., and Bulfin, R.L., "Complexity of multiple machine multiple criteria scheduling problems", in: *Proceedings of the Third IERC*, Atlanta, GA, 1994.

[10] Chen, C.L., Vempati, V.S., and Aljaber, N., "An application of genetic algorithms for flow shop problems", *European Journal of Operational Research* 80 (1995) 389–396.

[11] Cleveland, G.A., and Smith, S.F., "Using genetic algorithms to schedule flow shop release", in: *Proceedings of the Third International Conference on Genetic Algorithms Applications*, 1989, 160–169.

[12] Daniels, R.L., "Multi-objective flow shop scheduling", *Naval Research Logistics Quarterly* 37 (1992) 981–995.

[13] Davis, L., *Handbook on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1987.

[14] DeJong, K.A., "Analysis of the behavior of a class of genetic adaptive systems", Ph.D. Dissertation, University of Michigan, 1975.

[15] Dileepan, P., and Sen, T., "Bicriterion state scheduling research for a single machine", *Omega* 16 (1988) 53–59.

[16] Emmons, H., "One machine sequencing to minimize mean flow time with minimum number of tardy", *Naval Research Logistics Quarterly* 22 (1975) 585–592.

[17] Goldberg, D.E., and Lingle, Jr., R., "Alleles, loci and the Travelling Salesman Problem", in: *Proceedings of International Conference on Genetic Algorithms and their Applications*, Carnegie-Mellon University, Pittsburgh, PA, 1985, 154–159.

[18] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[19] Grefenstette, J.J., "Optimized control of parameters for genetic algorithms", *IEEE Transactions on Systems, Man, and Cybernetics* 16 (1986) 122–128.

[20] Grefenstette, J.J., "Incorporating problem specific knowledge into genetic algorithms", in: L. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, Los Altos, CA, 1987.

[21] Heck, H., and Roberts, S., "A note on the extension of a result on scheduling with secondary criteria", *Naval Research Logistics Quarterly* 19 (1972) 403–405.

[22] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[23] Holland, J.H., "Genetic algorithms", *Scientific American*, July (1992) 117–125.

[24] Johnson, S.M., "Optimal two- and three-stage production schedules with set-up times included", *Naval Research Logistics Quarterly* 1 (1954) 61–68.

[25] Keeney, R.L., and Raiffa, H., *Decision with Multiple Objectives: Preference and Value Tradeoffs*, Wiley, New York, 1976.

[26] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B. (eds.), *The Travelling Salesman Problem*, Wiley, New York, 1985.

[27] Miyazaki, S., "One machine scheduling problem with dual criteria", *Journal of the Operations Research Society of Japan* 24 (1981) 37–51.

[28] Nelson, R.T., Sarin, R.K., and Daniels, R.L., "Scheduling with multiple performance measures: The one machine case", *Management Science* 32 (1986) 464–479.

[29] Neppalli, V.R., Chen, C.L., and Aljaber, N.J., "An effective heuristic for the flow shop problem with weighted tardiness", in: *Proceedings of the Third Industrial Engineering Research Conference*, 1994, 634–638.

[30] Rajeev, S., and Krishnamoorthy, C.S., "Discrete optimization of structures using genetic algorithms", *Journal of Structural Engineering* 15 (1992) 1233–1250.

[31] Rajendran, C., "Two-stage flowshop scheduling problem with bicriteria", *Journal of the Operational Research Society* 43 (1992) 871–884.

[32] Ruiz-Diaz, F., and French, S., "A survey of multi-objective combinatorial scheduling", in: S. French, L.C. Thomas, R. Hartley and D.J. White (eds.), *Multi-objective Decision Making*, Academic Press, New York, 1983, 59–77.

[33] Sen, T., and Gupta, S.K., "A branch-and-bound procedure to solve a bicriterion scheduling problem", *IIE Transactions* 15 (1983) 84–88.

[34] Schaffer, J.D., "Multiple objective optimization with vector evaluated genetic algorithms", in: *Proceedings of International Conference on Genetic Algorithms*, Morgan Kaufmann, New York, 1989, 93–100.

[35] Shaffer, J.D., Caruna, R.A., Eshelman, L.J., and Das, R., "A study of control parameters affecting online performance of genetic algorithms for function optimization", in: J.D. Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithm*, Morgan Kaufmann, New York, 1989, 51–60.

[36] Shanthikumar, J.G., "Scheduling *n* jobs on one machine to minimize the maximum tardiness with minimum number of tardy", *Computers & Operations Research* 10 (1983) 255–266.

[37] Smith, W.E., "Various optimizers for single state production", *Naval Research Logistics Quarterly* 3 (1957) 59–66.

[38] Tzafestas, S., and Triantafyllakis, A., "Deterministic scheduling in computing and manufacturing systems: A survey of models and algorithms", *Mathematics and Computers in Simulation* 35 (1993) 397–434.

[39] Van Wassenhove, L.N., and Baker, K.R., "A bicriterion approach to time/cost trade-offs in sequencing", *European Journal of Operational Research* 11 (1982) 48–54.

[40] Van Wassenhove, L.N., and Gelders, L.F., "Solving a bicriterion scheduling problem", *European Journal of Operational Research* 4 (1980) 42–48.

[41] Vempati, V.S., Chen, C.L., and Bullington, S.F., "An application of genetic algorithms to the Flow Shop Problem", in: *Proceedings of 15th International Conference of Computers and Industrial Engineering*, 1993.

[42] Vickson, R.G., "Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine", *Operations Research* 28 (1980) 1155–1167.