Journal of Scheduling **8:** 75–96, 2005.

EVALUATING SOLUTION SETS OF *A POSTERIORI* SOLUTION TECHNIQUES FOR BI-CRITERIA COMBINATORIAL OPTIMIZATION PROBLEMS

JOHN W. FOWLER[1], BOSUN KIM[1], W. MATTHEW CARLYLE[2], ESMA SENTURK GEL[1], AND SHWU-MIN HORNG[3]

[1]*Industrial Engineering, Arizona State University, Tempe AZ 85287-5906, U.S.A.*
[2]*Naval Postgraduate School*
[3]*National Taipei University of Technology*

ABSTRACT

The quality of an approximate solution for combinatorial optimization problems with a single objective can be evaluated relatively easily. However, this becomes more difficult when there are multiple objectives. One potential approach to solving multiple criteria combinatorial optimization problems when at least one of the single objective problems is NP-complete, is to use an *a posteriori* method that approximates the efficient frontier. A common difficulty in this type of approach, however, is evaluating the quality of approximate solutions, since sets of multiple solutions should be evaluated and compared. This necessitates the use of a comparison measure that is robust and accurate. Furthermore, a robust measure plays an important role in metaheuristic optimization for "tuning" various parameters for evolutionary algorithms, simulated annealing, etc., which are frequently employed for multiple criteria combinatorial optimization problems. In this paper, the performance of a new measure, which we call Integrated Convex Preference (ICP) is compared to that of other measures appearing in the literature through numerical experiments—specifically, we use two *a posteriori* solution techniques based on genetic algorithms for a bi-criteria parallel machine scheduling problem and evaluate their performance (in terms of solution quality) using different measures. Experimental results show that the ICP measure evaluates the solution quality of approximations robustly (i.e., similar to visual comparison results) while other alternative measures can misjudge the solution quality. We note that the ICP measure can be applied to other non-scheduling multiple objective combinatorial optimization problems, as well.

KEY WORDS: multiple criteria combinatorial optimization, comparison measures, parallel machine scheduling

1. INTRODUCTION

Multiple criteria combinatorial optimization problems have received relatively little attention compared to single criterion combinatorial optimization problems in the literature. However, multiple objective optimization has become more important in today's competitive environment where continuous improvement along all fronts is essential to business success. Manufacturing operations must constantly choose a good alternative considering conflicting criteria such as maximizing throughput and minimizing cycle time, or minimizing WIP (work in process) inventory and maximizing on time delivery. As such, job scheduling is an area that requires optimization of conflicting performance criteria for most manufacturers. A scheduling solution that minimizes makespan will

Correspondence to: John W. Fowler. E-mail: john.fowler@asu.edu

not necessarily minimize the total weighted tardiness. Using methods to solve these types of problems expeditiously can be a key to maintaining a company's profitability or market share.

Many single criterion scheduling problems are NP-hard due to their inherent combinatorial nature and complicated problem structures (Pinedo, 1995). When multiple criteria are considered, these problems become even more difficult to solve optimally. In fact, for conflicting objectives, it is hard to even say what "optimal" means. As Fry, Armstrong, and Lewis (1989) pointed out, general combinatorial optimization techniques such as branch and bound and dynamic programming methods have limitations when applied to practical problems (e.g., number of jobs is more than 20 or 30). Thus, an approximate solution approach is a proper way to attack these real sized multiple criteria scheduling problems.

Until the early 90's, multiple criteria scheduling research was almost exclusively focused on single machine problems (Fry, Armstrong, and Lewis, 1989; De, Ghosh, and Wells, 1992; Lee and Vairaktarakis, 1993; Nelson, Sarin, and Daniels, 1986). From the middle of the 90's, research has been extended to more complicated multiple criteria scheduling problems such as parallel machine scheduling problems with sequence dependent setups (Cochran, Horng, and Fowler, 2003; Serifoglu and Ulusoy, 1999; Tuzikov, Makhaniok, and Manner, 1998) and flow shop scheduling problems with sequence dependent setups (Marett and Wright, 1996; Murata, Ishibuchi, and Tanaka, 1996). In these cases, each single objective problem is NP-hard (Pinedo, 1995). Hence the multiple criteria problems are clearly NP-hard. Many approximate algorithms such as genetic algorithms, simulated annealing, tabu search, and filtered beam search are introduced in the literature (Cunha, Oliveira, and Covas, 1997; Fonseca and Fleming, 1993; Loughlin and Ranjithan, 1997; Louis and Rawlins, 1993; Schaffer, 1985) to solve multiple criteria combinatorial optimization problems, including scheduling problems. The majority of these algorithms take an *a posteriori* approach. *A posteriori* approaches attempt to generate exact or approximate efficient solutions. The decision maker then selects the most preferred solution from the efficient solutions. This kind of solution approach is well suited to complicated scheduling problems. Hence research to develop more effective and robust algorithms for multiple criteria scheduling problems is needed.

In this case, however, evaluating the solution quality of competing algorithms or algorithms with different parameter sets, is not easy, since the "solution", which is an approximation of the efficient frontier is a set of near Pareto-optimal solutions (also called non-dominated solutions, efficient solutions) in the objective space. When the set of all true non-dominated solutions (i.e., efficient frontier) can be generated by competing algorithms, their performance can be compared by the computational effort needed to solve the same problem instance as is done in the single objective case. However, computational effort is not sufficient to compare competing algorithms when the true set of Pareto-optimal solutions cannot be obtained in a reasonable amount of time. This is the case for most multiple criteria scheduling problems. To compare the performance of heuristic algorithms, one approach is to run competing algorithms for the same amount of computational effort (CPU time or number of evaluations) and then compare the quality of solutions. Another approach is to run each algorithm to its own stopping criteria and then compare both solution quality and computational effort (Schaffer, 1985). In both cases, robust and efficient methods to compare the quality of sets of near Pareto-optimal solutions are required. We note that equating the CPU time of different algorithms can be impacted by different coding methods and data structures; thus this may not be a perfect comparison.

However, there seems to be no generally accepted measure(s) in the multiple criteria optimization literature, as pointed out in Carlyle et al. (2003). This is primarily due to the difficulties in comparing the various geometric features of sets of near Pareto-optimal solutions. These difficulties include the fact that a non-identical number of non-dominated solutions are generated by heuristics and the fact that heuristics generate non-identical tail points (extreme solutions for each objective). Often, only visual comparison of alternative sets of Pareto-optimal solutions is employed in bi-criteria optimization problems. However, visual comparison is not efficient, since typically many experiments are required to verify the effectiveness and robustness of heuristic algorithms. Hence, a new measure called Integrated Convex Preference (ICP), was proposed in Carlyle et al. (2003) to evaluate the quality of sets of Pareto-optimal solutions efficiently. Carlyle et al. (2003) provides a more detailed description of the theoretical development and properties of ICP. In this paper, the robustness and efficiency of the ICP measure for comparing *a posteriori* solution techniques are examined through extensive experiments using a bi-criteria parallel machine scheduling problem. These results are followed by a discussion on the properties of ICP based on the analysis of the experimental results.

The ICP has some similarities with Data Envelopment Analysis (DEA). Both methods assume weighted sum objective functions (outputs), do not determine a specific weight for each objective (output) *a priori*, and consider all envelopment points as efficient. Thus both methods require an efficient way to find all envelopment points from a given set of non-dominated solutions (Decision Making Units). However DEA and ICP have different purposes (i.e., DEA measures the relative efficiency of decision making units (DMUs) in the presence of multiple inputs and outputs and sets targets for inefficient DMUs), and thus need different methods or algorithms. For example, a method to find the optimal weight region for each envelopment point in weight space and integration over the optimal weight region are essential for ICP to evaluate the solution set quality (the decision maker's expected value for the set). However, these calculations are not needed in DEA.

In the next section, the literature concerning solution techniques for multiple criteria scheduling problems and the measures used to compare sets of Pareto-optimal solutions are reviewed. Then, a summary of the ICP measure is presented, which is followed by the experimental scheme used to compare alternative heuristics. The results are then discussed in the experimental results section and ICP properties are discussed. Finally, conclusions and future research topics are provided.

2. LITERATURE REVIEW

In this section, two topics are reviewed. First, solution approaches for multiple objective optimization problems are reviewed, with particular emphasis on methods for multiple objective scheduling problems. This is followed by the measures appearing in the literature to compare the solution quality of sets of Pareto-optimal solutions.

2.1. Solution approaches for multiple objective optimization problems

The majority of solution approaches for multiple criteria scheduling problems that have appeared in the literature can be divided into two categories (Fry, Armstrong, and Lewis, 1989). The first category is '*a priori*' solution approaches, which assume that the decision maker's preference information, such as a priority, weight, or goal (target) for each criterion, can be obtained before the solution procedure starts. A good example of an objective priority method is found in Lee and Vairaktarakis (1993), which dealt with bi-criteria single machine scheduling problems. The authors assumed that the priority of each objective is given and that the second objective is optimized subject to the constraint that the first criterion meets its minimum value. The paper provides the proof of NP completeness or provides polynomial time algorithms for almost all pairwise combinations of performance criteria considered in the literature. Applications of an objective weighting method can be found in Serifoglu and Ulusoy (1999) and Marett and Wright (1996). Serifoglu and Ulusoy (1999) suggested a genetic algorithm to solve the problem of $Q_m\ |s_{jk}|\ w_E \Sigma\ E_j + w_T \Sigma\ T_j$. They consider two types of machine groups, identical and proportional. In Marett and Wright (1996), compare a tabu search method to a simulated annealing method to solve the problem of F_3 |prmu, s_{jk}| (1) $C_{\max}$, (2) total setup cost, (3) total holding time (job waiting), (4) total late time (machine idle), for 30 jobs. They assume that the weight of each objective is given, and the objective function is linear.

The second category is *a posteriori* solution approaches where the decision maker's preference is not considered in advance of the solution methods. The decision maker eventually selects the single best solution. Nelson, Sarin, and Daniels (1986) suggested an optimal algorithm (non-polynomial time) for a single machine with the following pairs of performance measures; (1) mean flow time and number of tardy jobs; (2) number of tardy jobs and maximal tardiness; and (3) mean flow time and maximal tardiness. De, Ghosh, and Wells (1992) suggested an approximate algorithm (filtered beam search) to generate a set of efficient extreme solutions for a single machine problem with mean and variance of completion times criteria. They assumed that the scalar objective function is a convex combination of objectives and the weight of each objective is unknown. Due to the convex combination of objectives assumption, only the efficient extreme points in the objective space are obtained. Tuzikov, Makhaniok, and Manner (1998) suggested an optimal polynomial time algorithm to generate a set of non-dominated solutions for $Q_m\ |p_j = p, r_j|$ (1) $\Sigma\varphi_j\ (c_j(S))$, (2) $\max\{\psi_j\ (c_j(S))\}$, where $\varphi_j\ (c_j(S))$ and $\psi_j\ (c_j(S))$ are regular (non decreasing) cost functions for the completion time of job j. The problem is to schedule jobs with identical processing times on a uniform processor.

Genetic algorithms are often applied to multiple criteria optimization problems to generate a set of near Pareto-optimal solutions in a reasonable amount of computational effort. Schaffer (1985) proposed the Vector Evaluated Genetic Algorithm (VEGA) method to find a set of near Pareto-optimal solutions for general multiple objective problems. In this method, a population is divided into disjointed sub-populations and each sub-population is optimized with respect to one of the objectives. This method, by the nature of its 'disjointing approach (vector optimization)', tends to form the extreme solutions of the approximate efficient frontier since its search is unidirectional. The lack of a combined search in the Pareto-optimal solution set will naturally restrict the decision-maker's choices. Murata, Ishibuchi, and Tanaka (1996) proposed the Multi-Objective Genetic Algorithm (MOGA). MOGA selects individuals for a crossover operation, based on a weighted sum of linear objective functions with variable weights, which are not constant but are randomly specified for each generation. With these variable weights, MOGA searches in various directions. The method generally produces more diverse Pareto-optimal solutions, enabling the decision maker a broader choice of solutions. They applied MOGA to solve a multiple criteria flow shop scheduling problem, F_{10} |prmu| (1) $C_{\max}$, (2) $\sum w_j T_j$, (3) $\sum w_j C_j$, for 20 jobs and compared the solution quality of a set of near Pareto-optimal solutions generated by MOGA to that generated by VEGA. They showed that MOGA generates a better approximate efficient frontier than VEGA by visual comparison.

Cochran, Horng, and Fowler (2003), proposed the hybridized Multi-Population Genetic Algorithm (MPGA) to solve multiple criteria parallel machine scheduling problems with sequence dependent setups, $P_5\ |s_{jk}, r_j|$ (1) $C_{\max}$, (2) $\sum w_j T_j$, (3) $\sum w_j C_j$, for 100 jobs. In their study, a genetic algorithm is hybridized with dispatching rules. The GA is used to assign jobs to machines, and dispatching rules such as setup avoidance and apparent tardiness cost with setups rules (Pinedo, 1995), are used to schedule the individual machines. The method consists of two stages. In the first stage, multiple objectives are combined as the multiplication of the relative objective functions. In the second stage, the solutions of the first stage are rearranged and divided into several sub-populations, which are the initial populations of the second stage. Each sub-population evolves separately (similar to the VEGA approach). They also sought to find the best time to change between the two stages (called the turning point). MPGA outperformed MOGA when comparing them by the number of Pareto-optimal solutions and the number of combined Pareto-optimal solution measures, which will be defined in the next section (Cochran, Horng, and Fowler, 2003).

2.2. Comparison methods used in the literature

The methods to compare the solution quality of approximate algorithms used in the literature belong to four main groups. The first group involves the visual comparison of the sets of non-dominated solutions using graphical displays of the solution points in the objective space (Murata, Ishibuchi, and Tanaka, 1996; Cieniawski, Eheart, and Ranjithan, 1995). Although this method can compare solution sets of various shapes effectively, it can only be used for bi-criteria optimization problems and it is especially inefficient when a large number of numerical experiments (or replications) need to be performed. Also, such a method clearly cannot be embedded into an algorithm that automatically selects the best heuristic from among many alternative procedures or the best set of heuristic parameters from among a set of possible choices.

The second group of measures focuses on geometric features of the solution sets plotted in the objective space. Measures in this group include (1) length and area measures (De, Ghosh, and Wells, 1992) and (2) distance measures (Czyzak and Jaszkiewicz, 1998; Viana and Sousa, 2000). The length and area measures proposed by De, Ghosh, and Wells (1992) can only handle problems with two objectives and are not applicable when the solution set contains only one or two points. Although the distance measures evaluate the approximate solution set on the characteristics of diversity (coverage), uniformity and closeness defined in Carlyle et al. (2003) to the true set of Pareto-optimal solutions, they are not applicable when the true set of solutions is not available, which is often the case in scheduling problems.

The third group of comparison techniques uses the cardinality of the set of Pareto-optimal solutions. The measures in this group are (1) the number of Pareto-optimal solutions and (2) the number of combined Pareto-optimal solutions (Cochran, Horng, and Fowler, 2003; Schaffer, 1985; Hyun, Kim, and Kim, 1998). To obtain the number of combined Pareto-optimal solutions, all non-dominated solutions are compared together with respect to the Pareto-optimal criterion. If any solution is dominated, then it is discarded. After that the number of non-dominated solutions found by each algorithm is counted. While the number of generated solutions is important, it certainly cannot determine the solution quality quantitatively.

The fourth group uses the value (utility) function of the decision maker to obtain a scalar value of a set of Pareto-optimal solutions. Daniels (1992) suggested two measures, maximum and average approximation error (ε) of the discrete approximation from true efficient solutions under the assumption of a linear weighted sum utility (value) function. These measures and the ICP measure suggested in Carlyle et al. (2003) are similar in the sense that both methods utilize an assumption on the decision maker's value (unknown) function in evaluating the solution quality of multiple objective heuristics. However, Daniels' methods evaluate the solution quality of heuristics based on the known discrete true efficient frontier. Hence, they are similar to geometric measures in the sense that they are not applicable when the true set of solutions is not available, whereas ICP is designed for such cases. Hansen and Jaszkiewicz (1998) suggested three types of measures – probability difference, expected value difference, and relative ratio. The probability difference ($R1$) measure is obtained from the cumulative probability that Set-1 gives a better solution than Set-2, and vice versa. The expected value difference ($R2$) measure is similar to the difference between ICP values. The relative ratio ($R3$) measure is the same as the average approximation error (ε) in Daniels (1992). They assumed that $p(u)$ is the probability that utility function, u, is held by the decision maker. The authors also note that these three methods can be applied whether true efficient solutions are known or not. However they did not provide exact calculation methods for the three measures due to the difficulty of the high dimensional integration of smooth or non-smooth functions. Instead, the approximate method of sampling a set of utility functions according to its distribution is suggested in Hansen and Jaszkiewicz (1998). In comparison, ICP is an exact method to obtain a scalar value of a set of near Pareto-optimal solutions when a convex combination of objective functions is assumed.

3. INTEGRATED CONVEX PREFERENCE (ICP)

The basic concept of ICP begins with the fact that the most preferred single solution among feasible solutions would be eventually selected by the decision maker(s), regardless of the employed solution approach: *a priori*, interactive, or *a posteriori.* To select the most preferred solution among the feasible solutions, the decision maker applies his/her value function. Hence, we seek to use this value function approach in comparing the quality of sets of near Pareto-optimal solutions. ICP is, in short, the expected value or utility of a set of Pareto-optimal solutions for an assumed value function. The exact extraction of the value function of the decision maker is difficult and this is a research area itself. Hence, the most frequently used value function, a convex combination (weighted sum) of objective functions, is assumed in this paper to obtain ICP for a set of Pareto-optimal solutions although other types of value functions can be incorporated.

3.1. Integrated convex preference (ICP)

Consider a bi-criteria minimization problem. When the decision maker's preference can be represented as a convex combination of linear objective functions with varying weights, then an optimal solution x^* is selected by $\min_{j\in J}\{wf_{1j} + (1-w)f_{2j}\}$, where J is a set of non-dominated solutions, f_{ij} is the ith objective value of the jth non-dominated solution, and w is in the interval (0, 1). In Figure 1, for example, five efficient extreme points (lower-envelope points) are candidates for an optimal solution among the nine Pareto-optimal points. If the decision maker's preference (i.e., weight) for objective 1, w, equals 0, then the non-dominated point p_1 is the optimal solution since the decision maker only considers objective 2. If w equals 0.5 (both objectives are equally important), then p_2 will be the optimal solution. If w equals 1.0 then p_3 is the optimal solution. As shown in Figure 1, the weight interval for which an efficient extreme point p_2 is optimal can be obtained by a polar cone. This polar cone can be generated by the orthogonal vectors of the faces, which contain p_2. Then, ICP is obtained by Eq. (1).

$$ICP = \int_0^1 \min_{j\in J}\{wf_{1j} + (1-w)f_{2j}\}\, dw. \tag{1}$$

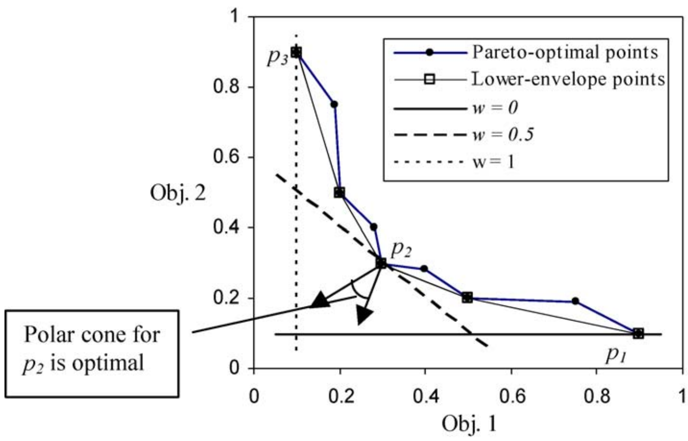

Figure 1. Optimal weight range for each efficient extreme solution in the objective space

The first step to obtain ICP is to find all efficient extreme points from a given set of Pareto-optimal solutions. This is the well-known convex hull problem (i.e., find all extreme points among a set of given points). A pseudo code for a convex hull algorithm in bi-criteria problems is provided below.

- Sort the (f_1, f_2) points in increasing order of f_2 value;
  /* Finding an adjacent extreme point sequentially */
- Let starting point $(f_1{}^0, f_2{}^0)$ be the first point in the sort list;
- Do until the last point in the sort list is selected as an adjacent point.
- Calculate the slope between the starting point $(f_1{}^0, f_2{}^0)$ and all the remaining points;
- Assign an adjacent point, which has the minimum slope to be the adjacent extreme point;
- Delete the non-extreme points between the starting point and the adjacent extreme point
- Assign adjacent extreme point to be the new start point $(f_1{}^0, f_2{}^0)$;

The second step is to calculate the weight intervals within which each extreme point is an optimal solution for a convex combination of the objective functions. There are several methods to obtain these optimal weight intervals (details can be found in Carlyle et al. (2003)). One of the efficient ways for the bi-criteria case is provided here, which can also be applied to more than two objective cases. Assume n efficient extreme solutions $f_{\bullet i}$, $i = 1, 2, \ldots, n$ are obtained from Step 1. Then a system of linear inequalities can be generated for each efficient extreme solution as Eq. (2).

$$\{wf_{1i} + (1-w) f_{2i}\} - \{wf_{1k} + (1-w) f_{2k}\} \leq 0, \quad i \neq k, \quad k \in K, \tag{2}$$

where K is the set of adjacent efficient extreme points of $f_{\bullet i}$.

The system of inequalities above is derived from the fact that if a solution $f_{\bullet i}$ is an optimal solution for a convex combination of objectives, then there should be a weight w for which $wf_{1i} + (1-w)f_{2i}$ is less than or equal to that of all other solutions. In a bi-criteria optimization problem, every efficient extreme point has two adjacent extreme points except for the two tail points. Thus, two linear inequalities can be generated from these two adjacent extreme points. The two linear inequalities give the lower and upper bounds on the optimal weight interval. The two tail points have one adjacent extreme point, which gives a bound on the optimal weight interval. And the other bound is 0 or 1 since w is assumed to belong to (0, 1). Then the optimal solution f^* in the objective space can be decomposed as a function of w as in Eq. (3).

$$f^*(w) = \begin{pmatrix} (wf_{11} + (1-w)f_{21}), & 0 \le w \le w_1 \\ (wf_{12} + (1-w)f_{22}), & w_1 \le w \le w_2 \\ \cdots\cdots & \\ (wf_{1n} + (1-w)f_{2n}), & w_{n-1} \le w \le 1 \end{pmatrix}, \tag{3}$$

where w_i means the breakpoint for the range of w for which extreme solution i is optimal.

Finally, ICP for a given set of Pareto-optimal solutions can be obtained by Eq. (4).

$$\begin{aligned} ICP &= \int_0^1 f^*(w)\,dw \\ &= \int_0^{w_1} (wf_{11} + (1-w)f_{21})\,dw + \int_{w_1}^{w_2} (wf_{12} + (1-w)f_{22})\,dw \\ &\quad + \cdots + \int_{w_{n-1}}^{1} (wf_{1n} + (1-w)f_{2n})\,dw \end{aligned} \tag{4}$$

3.2. *Integrated convex preference with triangular weight function (ICP_T)*

The decision maker(s) typically does not know his or her weight value for each objective precisely, but is able to specify some relations between weights. In comparing sets of Pareto-optimal soluitions, one practical assumption is that the decision maker wants to give more weight to solutions that are well compromised (good for both criteria: elbow solutions) than to solutions that are only good for one objective (tail solutions). This consideration can be modeled as a weight density function such as a triangular weight function. The ICP described earlier can be considered to have a uniform weight function. As shown in Figure 2, a set of Pareto-optimal solutions which has better solutions in the elbow area and worse solutions in the tail areas, may be preferred by a

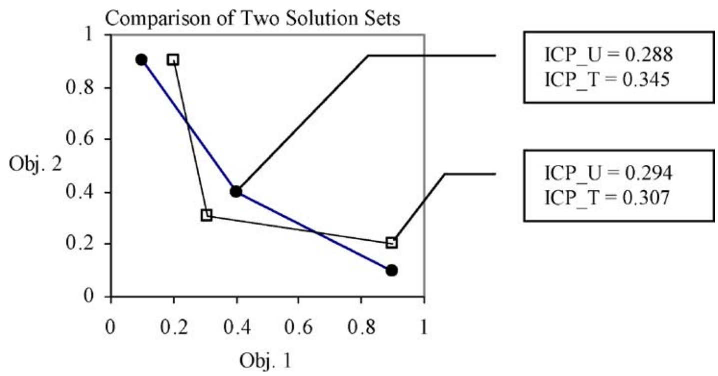

Figure 2. Comparison of two sets of Pareto-optimal solutions by ICP with a uniform weight function and a triangular weight function

decision maker over a set of Pareto-optimal solutions which has worse solutions in the elbow area and better solutions in both tail areas. In Figure 2, the circle shaped solutions have a lower ICP (i.e. are preferred) when compared to ICP with a uniform weight function. On the other hand, the rectangular shaped set of Pareto-optimal solutions has a lower ICP when compared to ICP with a triangular weight function. A detailed calculation procedure of ICP _T can be found in Carlyle et al. (2003).

3.3. Integrated convex preference with scaled objective values (ICP_*_S)

As stated before, the ICP uses a blended value function (unknown) to represent the preference of the decision maker. When objectives are incommensurable like number of tardy jobs and total completion time, it is hard to interpret a blended objective value. Also, when the difference between the ranges of each objective value is so large that one objective value overwhelms the other objective value, proper scaling is clearly needed. In our study, the range of total weighted tardiness is much larger than that of makespan. As Schenkerman (1990) suggested, minimum and maximum values in sets of Pareto-optimal solutions are used in scaling each objective value as shown in (5). The same scaling method is employed in De, Ghosh, and Wells (1992) to compare sets of Pareto-optimal solutions with area and length measures:

$$(g_{1i}, g_{2i}) = \left[\frac{\{f_{1i} - \min_{i\in I}(f_{1i})\}}{\{\max_{i\in I}(f_{1i}) - \min_{i\in I}(f_{1i})\}}\right], \quad \left[\frac{(f_{2i} - \min_{i\in I}(f_{2i}))}{(\max_{i\in I}(f_{2i}) - \min_{i\in I}(f_{2i}))}\right], \tag{5}$$

where (f_{1i}, f_{2i}) are the non-scaled objective values of non-dominated solution i, (g_{1i}, g_{2i}) are the scaled objective values of non-dominated solution i, and $\min_{i\in I}(f_{ji})$ $(\max_{i\in I}(f_{ji}))$ = minimum (maximum) $f_{\bullet i}$ among all $f_{\bullet i}$'s in competitive sets of Pareto-optimal solutions

4. EXPERIMENTAL SCHEME

To test the ICP measure, a multiple criteria parallel machine scheduling problem studied by Cochran, Horng, and Fowler (2003) is used. Extensive experimental results for the solution quality of two competing *a posteriori* solution techniques are reported. A summary of the experimental scheme is presented for the convenience of readers, even though details about the experiments can be found in Cochran, Horng, and Fowler (2003).

4.1. Test problem description

A parallel machine scheduling problem with sequence dependent setups is considered. A setup is required if the next job on the same machine has a different family. 100 jobs with 4 different families are scheduled on five identical machines. Two objectives are optimized simultaneously. The first objective is the makespan, defined as max $\{C_1, C_2, \ldots, C_n\}$, where C_j is the completion time of job j. The second objective is the total weighted tardiness (TWT), defined as TWT = $\sum_{j=1}^{n} w_j T_j$, where $T_j = \max\{0, C_j - d_j\}$ and d_j is the due date of job j. Thus, the problem can be represented as $P_5|s_{jk}, r_j|C_{\max}, \sum w_j T_j$, with 100 jobs. As shown in Table 1, four factors are used to generate the 100 jobs. A total of 36 problem instance sets can be generated using the four factors. Ten problem instances are generated randomly in each set, resulting in 360 test problem instances. All problem instances are solved 10 times due to the inherent randomness of the genetic

Table 1. Four factors and levels to generate 36 ($2^2 \times 3^2$) problem instance sets

Factors	Levels	Description
Range of weights	1 (Narrow)	$U(1, 10)$
	2 (Wide)	$U(1, 20)$
Range of due dates	1 (Narrow)	Ready time + $U(-1, 2) \times$ total process time.
	2 (Wide)	Ready time + $U(-2, 4) \times$ total process time.
Ratio ($\bar{p}/\bar{s}$)	1 (High)	$50/10$, $p = 50 + U(-9, 9)$, $s = U(6, 14)$.
	2 (Moderate)	$30/30$, $p = 30 + U(-9, 9)$, $s = U(18, 42)$.
	3 (Low)	$10/50$, $p = 10 + U(-9, 9)$, $s = U(30, 70)$.
WIP status	1 (High)	All jobs are ready at time 0
	2 (Moderate)	50% of jobs are ready at time 0 and the others are ready at time $U(0, 720)$
	3 (Low)	All jobs are ready at time $U(0, 720)$

Notes: U(a, b): Discrete random number generated from uniform distribution.
$\bar{p}$: Average process time.
$\bar{s}$: Average setup times.
Average setup times of job j: $(1/4) \times (\Sigma s_{ik})$, where s_{ik} is the setup time from a job of family i to a job of family k, i is the family of job j, and $k = 1, 2, 3, 4$.
Due dates are determined after ready times, processing times and setup times have been generated.

algorithms. Two genetic algorithms, MOGA (Murata, Ishibuchi, and Tanaka, 1996) and MPGA (Cochran, Horng, and Fowler, 2003), described in the literature review section are tested for all problem instances.

4.2. *Parameter settings for each algorithm*

In Cochran, Horng, and Fowler (2003), preliminary experiments were performed to find the best parameter settings for both genetic algorithms, since the performance of each genetic algorithm is dependent on the parameter settings used. We use the same parameter settings as in Cochran, Horng, and Fowler (2003), listed as follows:

- Crossover probability: 0.6
- Mutation probability: 0.01
- Population size: 20
- Elitism: three elite solutions are selected from the tentative set of non-dominated solutions
- Stopping criteria: 5000 generations

For MPGA, the turning criterion is set at the 2000th generation. After the turning criterion has been reached, the population is divided into three sub-populations, one for each of the two objectives and one for the combined objective function.

4.3. *Measures to compare MPGA with MOGA*

To provide evidence on which measure gives reasonable and robust comparison results for sets of near Pareto-optimal solutions, the measures in the literature and the different types of

ICP measures need to be tested. However, the geometrical comparison methods outlined in the literature review section (De, Ghosh, and Wells, 1992; Czyzak and Jaszkiewicz, 1998; Viana and Sousa, 2000; Daniels, 1992) are not applicable because there exist no efficient algorithms to generate the true set of Pareto-optimal solutions for the scheduling problem considered. Thus, the methods we experiment with are restricted to the following four: (1) visual comparison, (2) the number of Pareto-optimal solutions (# of POS), (3) number of combined Pareto-optimal solutions (# of CPOS), and (4) ICP. In ICP measures, four types of ICP measures—uniform weight function with scaling (ICP_U_S), triangular weight function with scaling (ICP_T_S), uniform weight function without scaling (ICP_U), and triangular weight function without scaling (ICP_T) are considered.

There are three types of comparison methods using the ICP. The first one is to compare the ICP values directly. This method is useful when several sets of Pareto-optimal solutions (several heuristics) need to be compared simultaneously. A set of solutions with the minimum ICP value can then be considered as the best set of solutions among the alternatives. Also, when parameter optimization (tuning) is performed through an experimental design and response surface optimization, ICP values can be used as a response value. The second method is the ICP difference between two sets of Pareto-optimal solutions (two heuristics). When pairwise comparison is needed, ICP difference can be used to determine which set of Pareto-optimal solutions has better solution quality, by the sign of the difference between ICP values. The magnitude of ICP difference represents the quantitative difference between two solution sets. A third method of comparison is the ratio of ICP values (e.g., $\frac{ICP(A)-ICP(R)}{ICP(R)}$ or $\frac{ICP(A)}{ICP(R)}$, where R is a reference set and A is an approximate set). When a reference set of solutions such as a set of true Pareto-optimal solutions is known, the ratio of ICP can provide useful information about the solution quality of heuristics based on the solution quality of the reference set. In this paper, two heuristics are compared and no reference sets are available, hence, ICP difference is used.

5. DISCUSSION OF EXPERIMENTAL RESULTS

Table 2 contains the experimental results of four types of ICP measures and two types of cardinality measures for 36 problem instance sets. The 'Problem instance set' column represents the combination of levels of four factors. For example, '1111' means that level '1' is used to generate 100 jobs for all four factors in Table 1. Values in the four 'ICP_*' columns represent the number of wins of MPGA (over MOGA) out of 100 comparisons (10 randomly generated problem instances * 10 replicates) in using the corresponding ICP measure. In the '# Pareto-optimal' column, the average number of Pareto-optimal solutions generated by MPGA and MOGA are shown respectively. In the '# Combined Pareto' column, the average combined number of Pareto-optimal solutions generated by MPGA and MOGA are shown. In the 'Total' row, values in the four 'ICP' columns indicate the number of wins of MPGA out of 3,600 comparisons and values in the last four columns indicate the sum of the average number of Pareto-optimal and combined solutions for MPGA and MOGA, respectively. Finally, in the 'Ratio' row, values are the ratio of the number of wins of MPGA out of the total number of comparisons.

The first thing to notice in Table 2 is that all ratio values are greater than or equal to 0.5. This means that MPGA outperforms MOGA in overall performance. This result is consistent with the results in Cochran, Horng, and Fowler (2003). However, comparison results (total number of wins) are much different depending on the measure used. When the number of Pareto-optimal solutions is used, MPGA wins in all 36 problem instance sets. When ICP_U, ICP_T, and combined

Table 2. Number of wins (0.5 for a tie) of MPGA out of 100 comparisons in each 36 problem instance set

Problem instance set	ICP_U	ICP_U_S	ICP_T	ICP_T_S	# Pareto-optimal		# Combined Pareto	
					MPGA	MOGA	MPGA	MOGA
1111	62	50	62	47	11.9	9.4	6.6	5.2
1112	67	61	67	57	12.0	9.5	8.2	4.2
1113	81	77	81	75	10.7	8.4	7.9	2.5
1121	55	47	55	43	11.7	8.8	6.0	5.4
1122	65	56.5	65	56.5	5.7	4.6	3.5	1.9
1123	66	62	66	60	7.0	5.4	4.4	2.3
1131	52	46	52	42	12.6	8.6	5.8	5.6
1132	44	42	44	41	1.8	1.2	0.6	0.7
1133	58	44	58	43	2.9	1.9	1.2	1.1
1211	59	53	60	53	12.0	9.7	7.3	4.7
1212	70	62	70	62	14.1	9.5	9.5	4.0
1213	84	69	84	66	10.8	8.9	7.9	3.0
1221	57	45	57	39	11.4	9.2	6.0	5.4
1222	73	60	73	57	6.7	4.8	3.9	2.0
1223	71	55	71	53	7.9	5.3	5.0	2.5
1231	52	49	52	45	12.5	9.5	5.7	6.1
1232	54	49	54	47	2.1	1.3	0.8	0.7
1233	48	38	50	35	4.3	1.9	1.5	1.3
2111	58	53	58	52	12.2	8.9	6.6	4.8
2112	76	71	76	71	13.3	9.4	9.7	3.7
2113	72	62	72	60	11.5	8.8	7.1	3.7
2121	64	54	64	52	11.9	8.7	6.7	4.5
2122	74	62.5	74	61.5	5.7	4.8	3.7	1.4
2123	63	55	63	52	8.1	5.4	4.5	2.7
2131	55	41	55	39	12.6	9.3	6.0	6.0
2132	51	47	51	47	1.6	1.3	0.7	0.7
2133	62	44	63	43	3.3	2.0	1.3	1.1
2211	53	51	54	46	12.4	9.3	6.9	4.9
2212	82	73	82	69	13.8	9.4	10.7	3.2
2213	80	74	80	73	11.9	9.2	8.9	3.0
2221	59	49	59	43	11.7	8.6	6.5	4.7
2222	73	65	73	64	7.5	5.2	5.0	1.8
2223	60	49	60	48	8.5	5.6	4.7	2.9
2231	45	45	45	45	12.9	9.6	6.1	6.0
2232	45	41	46	39	1.9	1.3	0.7	0.8
2233	59.5	39.5	59.5	39.5	4.2	2.0	1.8	1.0
Total	2249.5	1941.5	2255.5	1865.5	323.1	236.7	189.4	115.5
Ratio	0.62	0.54	0.63	0.52	1.00		0.89	

Pareto-optimal solutions are used, MPGA wins 32 times out of 36 problem instance sets. When ICP_U_S is used, however, MPGA only wins 19 times out of 36 comparisons. MPGA and MOGA are tied (each wins 18 times) when ICP_T_S is used.

To verify the effectiveness of the six measures used in evaluating the quality of sets of near Pareto-optimal solutions with various shapes, a detailed investigation for randomly selected problem instances is performed. Problem instance set 2212 is randomly selected among the 36 sets. Two problem instances, the best one for MPGA and MOGA, are selected in that problem instance set to analyze the performance of the measures considered.

At first, visual comparison is performed as a baseline due to the lack of a standard measure(s). Then, the performances of all other numerical measures are compared to the visual comparison results. To avoid the subjectivity involved in visual comparison, the set Pareto dominance relation (see Definition 1 in the next section) is used. If solutions in Set-1 dominate all of the solutions in Set-2 visually (clear cases), then Set-1 is judged to be a winner, and vice versa. If Set-1 and Set-2 cross each other (not-clear cases), then visual comparison will not decide the winner. Visual comparisons of sets of near Pareto-optimal solutions generated by MPGA and MOGA for problem instance set 2212 are shown in Figures 3 and 4. Visual comparison results by graph, four ICP measures, and two cardinality measures for Figures 3 and 4 are provided in Tables 3 and 4, respectively.

In Tables 3 and 4, the visual comparison columns document the visual comparison results. In 'clear cases', the name of the algorithm that is decided as a winner is represented and a '–' represents the 'not-clear' cases. Numbers in the four 'ICP_*' columns represent the ICP difference between two sets of Pareto-optimal solutions by MOGA and MPGA. A positive value means that MPGA is evaluated as a winner, and vice versa for a negative value. In the cardinality number measure

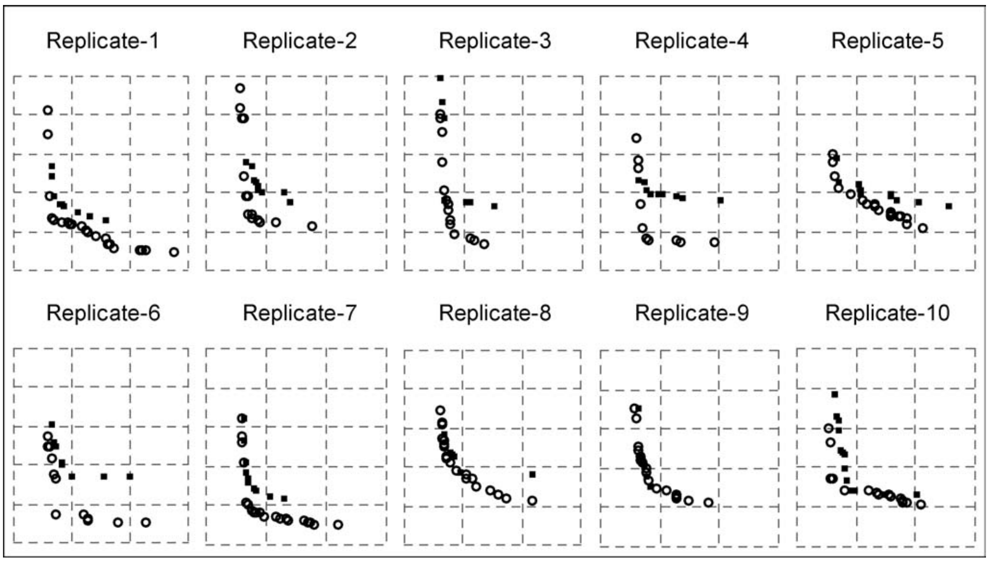

Figure 3. Visual comparison of MPGA with MOGA—best case for MPGA in problem instance set 2212 ('o' represents solutions of MPGA and '–' represents solutions of MOGA. *X*-axis is makespan from 1,000 to 1,300 time units and each grid line is 100 time units. *Y*-axis is total weighted tardiness from 200,000 to 350,000 time units and each grid line is 30,000 time units)

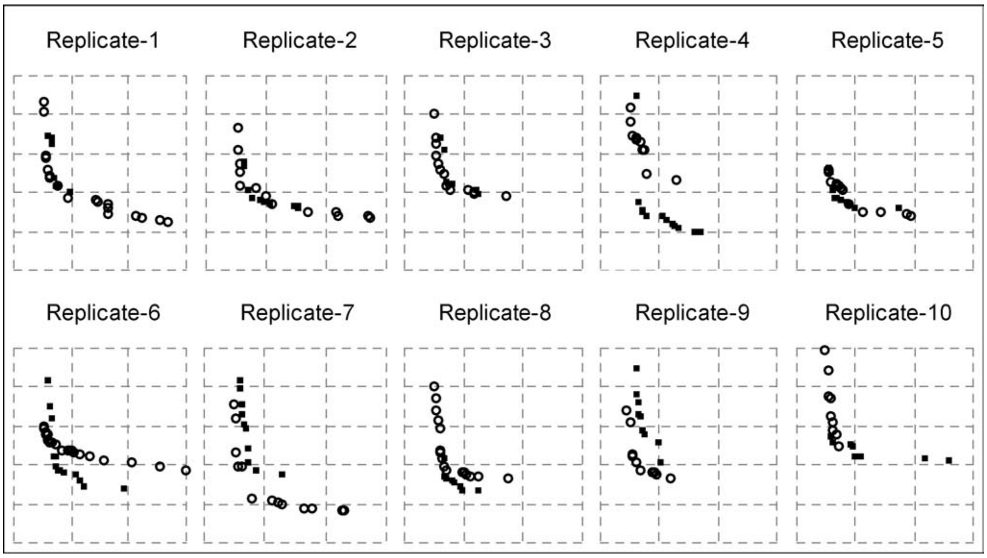

Figure 4. Visual comparison of MPGA with MOGA—best case for MOGA in problem instance set 2212. ('o' represents solutions of MPGA and '–' represents solutions of MOGA. *X*-axis is makespan. *Y*-axis is total weighted tardiness)

columns, the number of Pareto-optimal solutions and the combined number of Pareto-optimal solutions are represented.

By visual comparison of the 10 graphs in Figure 3, we can see that MPGA generates better Pareto-optimal solutions, except in Replication 9. MPGA and MOGA generate similar solutions for the makespan objective, but MPGA outperforms MOGA for the total weighted tardiness objective. In Figure 4, MPGA wins four times in visual comparison. In both figures, the objective value range difference between makespan and total weighted tardiness is significant. For example, the makespan objective values range from 1,000 to 1,300, and the total weighted tardiness objective values range from 200,000 to 350,000 in Figure 3. Hence, the makespan objective is overwhelmed by total weighted tardiness in non-scaled ICP measures (ICP_U and ICP_T). In this case, the scaled ICP can provide more reasonable comparison results.

An analysis of Tables 3 and 4 shows that the number of Pareto-optimal solutions cannot be used alone since it does not reflect the overall solution quality of the set. Consider, for example, Replicate-6 in Table 4. MPGA would be preferred over MOGA if the number of Pareto-optimal solutions is used (21 vs. 13). However, from a visual comparison (see Replicate-6 in Figure 4) MOGA appears to be better than MPGA in this case, even though both Pareto-fronts cross each other.

The number of combined Pareto-optimal solutions provided the same results that visual comparison did in all the clear cases, which shows that this measure works pretty well when the difference in the solution qualities of competitive algorithms is fairly large. However, this measure does not consider the location of Pareto-optimal solutions in the not-clear cases. For example, in Replicate-2 in Figure 4, MPGA has better solutions in both tail areas and MOGA has better solutions in the elbow areas. MPGA would be preferred over MOGA if this measure was used (14 vs. 8), which shows that although this measure is more effective than the number of Pareto-optimal solutions, it still has shortcomings.

Table 3. ICP difference between MPGA and MOGA in best case for MPGA in problem instance set 2212

		Difference between MOGA and MPGA				# Pareto-optimal		# Combined Pareto	
Replicate	Visual comparison	ICP_U	ICP_U_S	ICP_T	ICP_T_S	MPGA	MOGA	MPGA	MOGA
1	MPGA	12,293	0.065	24,660	0.059	22	9	22	1
2	MPGA	9,091	0.086	18,211	0.093	14	10	14	0
3	MPGA	14,317	0.067	28,626	0.058	14	9	12	2
4	MPGA	16,479	0.120	32,949	0.122	12	9	8	1
5	MPGA	8,278	0.031	16,534	0.017	18	10	18	1
6	MPGA	17,477	0.136	34,974	0.143	12	8	12	0
7	MPGA	9,878	0.063	19,812	0.064	22	10	22	2
8	MPGA	10,337	0.026	20,691	0.015	19	7	18	2
9	–	5,756	−0.001	11,579	−0.013	17	6	9	5
10	MPGA	3,709	0.039	7,434	0.037	15	13	14	1

Note: Value = ICP(MOGA)—ICP(MPGA).
'–' in visual comparison column means that it is difficult to judge whether MPGA is winner or not by visual comparison. This occurs when competing sets of Pareto-optimal solutions cross each other.

Table 4. ICP difference between MPGA and MOGA in best case for MOGA in problem instance set 2212

		Difference between MOGA and MPGA				# Pareto-optimal		# Combined Pareto	
Replicate	Visual comparison	ICP_U	ICP_U_S	ICP_T	ICP_T_S	MPGA	MOGA	MPGA	MOGA
1	MPGA	11,385	0.031	22,895	0.015	20	7	19	2
2	–	4,231	0.005	8,573	−0.004	16	10	14	8
3	MPGA	1,188	0.009	2,411	0.007	13	7	12	1
4	–	−20,110	−0.145	−40,242	−0.143	10	12	4	11
5	–	2,827	−0.009	5,674	−0.016	14	10	6	9
6	–	−6,946	−0.068	−13,822	−0.070	21	13	7	10
7	MPGA	14,416	0.097	28,902	0.091	13	10	13	0
8	–	−5,339	−0.042	−10,632	−0.040	16	8	8	8
9	MPGA	5,784	0.071	11,588	0.074	10	9	10	0
10	–	−4,838	−0.026	−9,784	−0.021	9	8	2	6

Note: Value = ICP(MOGA)—ICP(MPGA).

In all the clear cases, the four ICP difference measures provided the same results that a visual comparison did (all four measures have the same sign). If all four measures have the same sign, one set of solutions has better solutions in the elbow area and in both tail areas, or one set of solutions has better solutions in the elbow area and significantly better solution in one tail area. And the ICP difference is larger, relatively, than that of the not-clear cases (where all four ICP differences do not have the same signs). Consider, for example, Replicate-3 in Table 4. The ICP_U difference is

1,188, the least difference among 10 replicates, but all 4 signs of ICP difference are positive. Hence it can be interpreted that MPGA generates a better set of Pareto-optimal solutions for both the tail and elbow areas, even though the difference is very small.

However, in not so clear cases, the four ICP measures provide different comparison results (provide different signs) for the same two sets of Pareto-optimal solutions. Consider, for example, Replicate-9 in Figure 3. Even though the two sets of solutions are very close and cross each other, MGPA has better solutions for the total weighted tardiness objective and MOGA has better solutions in the elbow area. Both sets of solutions have similar solutions for the makespan objective. As can be seen in Table 3, the comparison result by ICP_U (ICP_T) is that MPGA generates better solutions than MOGA. On the other hand, the comparison result by ICP_U_S (ICP_T_S) is reversed. This is due to the objective value range difference between the two objectives as indicated earlier. Hence, positive ICP_U and negative ICP_U_S can be interpreted as MPGA has better solutions for the TWT objective and MOGA has better solutions for the makespan objective or in the elbow area. And negative ICP_T_S implies that MPGA has worse solutions in the elbow area. Thus, it appears that MOGA has better solutions in the elbow area (at least). For Replicate-10 in Table 4, all four ICP differences are negative, even though this is one of the not-clear cases due to the crossing of sets of Pareto-optimal solutions. This implies that MOGA generates better Pareto-optimal solutions for the TWT objective and for the elbow area. For Replicate-2 in Table 4, ICP_U, ICP_U_S, and ICP_T are positive, but ICP_T_S is negative. It can thus be interpreted that MPGA has better solutions in almost all the weight ranges, but has worse solutions in the elbow area. And we can see that ICP_T is not sensitive enough to give some useful information in our experiments. The signs of all 20 ICP_T differences are the same as the signs of the ICP_U differences as shown in Tables 3 and 4. This is due to the large difference in objective value range between the two objectives.

The magnitude of ICP difference also provides useful information to interpret the difference of solution qualities of competing sets of solutions. For example, the maximum ICP_U difference between MPGA and MOGA is 17,477 in Table 3, which implies the maximum difference of the solution quality of MPGA and MOGA occurs in Replicate-6 (also the maximum difference of ICP_U_S is 0.136 in Replicate-6). The minimum ICP_U difference is 3,709 in Replicate-10 among clear cases. And the minimum ICP_U_S difference is - 0.001 in Replicate-9. This implies that the difference of the solution quality of MPGA and MOGA is very small in both cases.

To summarize, the ICP measure gave comparison results that were the closest to the method of visual comparison, which is taken as the baseline comparison method in this study. For replications with a clear winner, ICP yielded the same preferences as visual comparison and for the not-clear replications the ICP values for the two sets were close. These interpretations are consistent with visual comparison.

Based on the analyses above, we can interpret the entire experimental results in Table 2 again. MPGA outperformed (i.e., had better values for all four ICP measures) MOGA 18 times out of the 36 problem instance sets. On the other hand, MOGA outperformed MPGA 4 times out of the 36 problem instance sets (1132, 1233, 2231, and 2232). In the remaining 14 problem instance sets (e.g., 1121), neither algorithm generates a dominant set of solutions.

The four MOGA wins occur when the process time/setup time ratio is '3' (low). This implies that the MPGA and MOGA performance depend on the factor levels. To perform the factor analysis, Table 5 is derived from Table 2. As shown in Table 5, MPGA wins across all weight range and due date range factor levels. However, when the level of the process-setup ratio is '3' and WIP ratio factor is '1', it cannot be said that MPGA outperforms MOGA.

Table 5. Number of wins of MPGA by levels of four factors

Factor	Level	ICP_U	Win	ICP_U_S	Win	ICP_T	Win	ICP_T_S	Win
Weight range	1	1118	1	965.5	1	1121	1	921.5	1
	2	1131.5	1	976	1	1134.5	1	944	1
Due date range	1	1125	1	975	1	1126	1	942	1
	2	1124.5	1	966.5	1	1129.5	1	923.5	1
Process- setup ratio	1	844	1	756	1	846	1	731	1
	2	780	1	660	1	780	1	629	1
	3	625.5	1	525.5	0	629.5	1	505.5	0
WIP ratio	1	671	1	583	0	673	1	546	0
	2	774	1	690	1	775	1	672	1
	3	804.5	1	668.5	1	807.5	1	647.5	1

Note: In 'win' columns, '1' means MPGA wins over 900 times out of 1800 comparisons in the weight range factor and the due date range factor rows and MPGA wins over 600 times out of 1200 comparisons in the process-setup time ratio factor and the WIP status factor rows.

For further analysis of the relation between the performance of MPGA and process-setup and WIP ratio factors, Table 6 is derived from Table 2. As shown in Table 6, when the level of process-setup ratio is '3'(low), or when the level of WIP ratio is '1', it is difficult to say which one generates a better Pareto front. MOGA outperformed MPGA when the level of process-setup ratio is '3'(low) and the level of WIP ratio is '2'(moderate). In all other cases, MPGA outperforms MOGA.

Because of the randomness of the genetic algorithms, it is not easy to understand the exact reasons why the performance of the algorithms varies for different problem instances. However, one of the reasons is due to the algorithmic characteristics of MPGA. In MPGA, the population is divided after the specified turning criteria. Then each subpopulation evolves for the improvement of the objective assigned to it. This is one of the reasons that MPGA outperformed MOGA in

Table 6. Number of wins of MPGA by levels for sensitive factors

Process setup time ratio	WIP ratio	ICP_U	Win	ICP_U_S	Win	ICP_T	Win	ICP_T_S	Win
1	1	232	1	207	1	234	1	198	0
1	2	295	1	267	1	295	1	259	1
1	3	317	1	282	1	317	1	274	1
2	1	235	1	195	0	235	1	177	0
2	2	285	1	244	1	285	1	239	1
2	3	260	1	221	1	260	1	213	1
3	1	204	1	181	0	204	1	171	0
3	2	194	0	179	0	195	0	174	0
3	3	227.5	1	165.5	0	230.5	1	160.5	0

Note: In 'win' columns, '1' means MPGA wins over 200 times out of 400 comparisons.

most cases. However, once an objective reaches the optimal or near optimal solution before the termination of the algorithm, then the subpopulation assigned to that objective has little chance to improve the solutions.

As shown in Table 1, when the level of process time and setup time ratio is 3(low), the process times of 100 jobs are randomly generated from $U(1, 19)$ and the setup times are from $U(30, 70)$. In this case, the makespan objective is much more dependent on the setup times than process times. As stated before, there are four families and five identical machines. Thus a near optimal schedule for the makespan objective can be obtained easily by assigning the jobs with the same family to the same machine. Once a job sequence that satisfies this approximately is determined by the genetic algorithm (crossover or mutation operation), then not much room remains for improving the solutions. It is more likely that the makespan objective reaches a (near) optimal solution within a relatively few generations in level 3 than level 1 or level 2 of the process time and setup time ratio. In the similar way, when the level of WIP status is 1(high), the release time (r_j) of all jobs are zero. In this case, the makespan objective without release times is easier to solve than the one with non zero release times.

6. ICP PROPERTIES

Several useful ICP properties can be derived by the following definitions and the experimental results.

Definition 1 (Set Pareto dominance relation). Assume sets of Pareto-optimal solutions (Pareto-fronts) A and B are not empty and $A \cup B = C$. The set of Pareto-optimal (non-dominated) solutions from set C is D. If $D \equiv A$ (or $D \equiv B$), then $A(B)$ dominates $B(A)$ in a set Pareto dominance relation. Thus, if the number of combined Pareto-optimal solutions from one solution set is 0, then the solution set is dominated by the other solution set according to the set Pareto dominance relation. In this case, all four ICP difference values have the same sign as can be seen in Property 1. If $D \neq A$ (or $D \neq B$) and $D \supset A$ (or $D \supset B$), then the two sets have a 'Non Set Pareto dominance relation'. When two sets belong to this relation, these sets can be classified according to the following Definition 2.

Definition 2 (Cross relation). Assume there are two sets of Pareto optimal solutions A and B which are not empty and include a finite number of solutions. Let the efficient frontier of a set be a set of lines drawn between two adjacent points in that set and a line from each tail point directed to each objective. If the efficient frontiers of the two sets cross each other, then the two sets have a 'cross relation'. If the efficient frontiers of the two sets do not cross each other, then the two sets have a 'non cross relation'.

According to Definitions 1 and 2 above, the sets of Pareto optimal solutions can be categorized into three cases; (1) set Pareto dominance relation, (2) non set Pareto dominance relation with non cross relation, and (3) non set Pareto dominance with cross relation. These three cases are illustrated in Figure 5. When two sets have the set Pareto dominance relation, then the two sets also have a non-cross relation as can be seen in Figure 5(a). When two sets have the non set Pareto dominance relation, the two sets can have a cross or non-cross relation as can be seen in Figure 5(b) and (c) respectively.

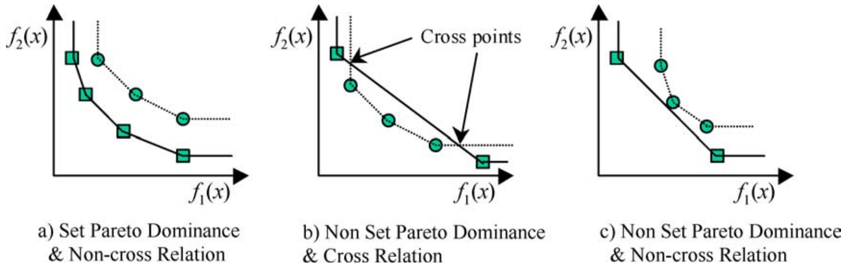

Figure 5. ‘Set Pareto Dominance’ and ‘Cross’ Relation of two efficient frontiers

The following properties show that these cases can be estimated and hence, evaluated as such by numerical ICP measures.

Property 1. In minimization problems, if a set of Pareto-optimal solutions B is dominated by a set of Pareto-optimal solutions A in a set Pareto dominance relation, then ICP(A) is always less than or equal to ICP(B) regardless of scaling of the objective value or the types of weight density functions used.

Proof. Recall that ICP is proportional to the sum of $f^*(w)$ for every weight w (0,1). If set A dominates set B in a set Pareto dominance relation, then $f^*(w)$ of set A is always less than or equal to $f^*(w)$ of set B for every weight. Hence the sum of $f^*(w)$ of set A is always less than or equal to the sum of $f^*(w)$ of set B for all weights. Let the step length of w go to 0 ($\Delta w \to 0$), i.e. the number of values of w goes to infinity. Then the sum of $f^*(w)$ for all weights within (0, 1) converges to the integration of $f^*(w)$ over (0, 1). Therefore Property 1 holds. We note that the reverse clearly does not always hold. ■

Corollary 1. In minimization problems, if two sets A and B have a non set Pareto dominance with non cross relation, then ICP(A) is always less than or equal to ICP(B) regardless of scaling the objective values or the types of weight density function used.

Proof. From the proof of Property 1, it can be easily proved. ■

Corollary 1 may be a weak point of the ICP measures, since well-compromised non-extreme solutions may actually be preferable to good solutions for only one objective. However, this seems not to have occurred in practical cases as can be seen in Figures 3 and 4. From Property 1 and Corollary 1 above, if two sets have either a set Pareto dominance relation or non set Pareto dominance with non cross relation, then all four ICP differences have the same sign. In our experiments, this can be detected in Replication 1 through 8 and 10 in Figure 3 (Table 3), and Replications 1, 3, 7, and 9 in Figure 4 (Table 4).

By visual comparison, the cross relation and location (elbow or tail) of two sets can be easily detected. Using several ICP measures, the cross relation and location can be detected numerically as can be seen in the Propertys 2 and 3. This will be helpful to interpret the solution quality of sets of Pareto optimal solutions in a computerized framework.

Property 2. If the ICP_U and ICP_U_S differences have different signs, then the two Pareto fronts cross each other. But the reverse does not always hold.

Proof. If set A is better in ICP_U but set B is better in ICP_U_S then, from the proof of Property 1, there is at least one point (in the scaled objective space) in set B at which $f^*(w)$ is the lowest for a certain weight w. In the same way, there is at least one point (in the objective space) in set A at which $f^*(w)$ is the lowest for a certain weight w. Thus, if ICP_U and ICP_U_S have different signs, then two sets have a cross relation. ■

In our experiments, Property 2 can be detected in Replicate-9 in Table 3, Replicate-5 in Table 4 (see corresponding graphs from Figures 3 and 4).

Property 3. If the signs of ICP_U (ICP_U_S) and ICP_T (ICP_U_T) differences are different, then the competing Pareto fronts cross each other. More precisely the crossing occurs in the elbow area rather than the tail area. But the reverse does not always hold.

Proof. Property 3 can be easily proved from the definition of ICP_U (ICP_U_S) and ICP_T (ICP_U_T). In our experiments, Property 3 can be seen in Replicate-2 in Table 4 and Figure 4. ■

7. CONCLUSIONS AND FUTURE RESEARCH

An *a posteriori* solution approach is one of the practical ways to attack multiple criteria combinatorial optimization problems. Approximate solution techniques will generally be more appropriate to solve these problems rather than exact methods due to the complexity inherent in these problems. In developing and applying such heuristics to solve practical problems, robust and efficient measures play an important role in (1) evaluating the quality of sets of Pareto-optimal solutions and comparing competing algorithms robustly, (2) optimizing parameters through experimental design and response surface optimization when heuristics which have a stochastic nature are employed, and (3) determining the stopping criteria of heuristic algorithms based on the approximate convergence of the solution quality. For these purposes, the Integrated Convex Preference (ICP) family of functions was suggested and the performance of four ICP measures and two cardinality measures were tested to verify the appropriateness of the measures in evaluating the quality of sets of Pareto-optimal solutions.

Through the experiments for a multiple criteria parallel machine scheduling problem, we were able to show that the two cardinality measures can misjudge the quality of near Pareto-optimal solutions. Also, we saw that there can be large objective value range differences between objective values (e.g. makespan and total weighted tardiness) in a set of Pareto-optimal solutions. This range difference can lead to misleading results when non-scaled ICP (ICP_U, ICP_T) measures are used. Scaled ICP measures (ICP_U_S or ICP_T_S) give more robust comparison results in such cases and specifically ICP_T_S can be used to check the solution quality in the elbow area.

Experimental results show that MPGA outperformed MOGA in overall performance for the 36 problem instance sets. However, we found that the solution quality of the algorithms is dependent on the problem instance through the comparison results of the scaled ICP measures. MOGA works better than MPGA when one objective can be optimized much easier than the other objective. This is because MPGA may waste a sub-population that was assigned to improve an already optimized objective function.

ICP measures use only efficient extreme points among a set of Pareto-optimal solutions due to the assumption that the decision maker's value function is a convex combination of objective functions. Hence, ICP has a limitation in comparing sets of Pareto-optimal solutions, which include all the same efficient extreme solutions and different non-supported solutions. In this case, different types of value functions can be considered to overcome this limitation. For example, if a weighted Tchebycheff metric is assumed as the decision maker's value function, then all of the Pareto-optimal solutions in a set will be considered in evaluating the solution quality of sets of near Pareto-optimal solutions (Carlyle et al., 2003; Miettinen, 1999). However, this case seems to occur rarely in comparing approximate algorithms. We observed no such case in our experiments. Also, considering the fact that a set of efficient extreme solutions provides the boundary information of a set of Pareto-optimal solutions, it can be concluded that ICP delivers a good representative scalar value of a set of non-dominated solutions from a geometric point of view. The experimental results in this paper support this conclusion.

Further research is needed on three topics. The first is to extend the ICP measure for three or more criteria cases and for non-convex value functions such as the weighted Tchebycheff metric. To extend the ICP for three or more objectives under the assumption of a convex value function, we need to develop an efficient convex-hull algorithm for three or higher dimensions by utilizing the fact that input points are a set of non-dominated solutions and an efficient method to integrate over the disjoint polytope regions of the parameter space. The weighted Tchebycheff metric as discussed above, provides consideration of non-dominated points that are not extreme solutions. Hence, we will develop a version of ICP that uses the weighted Tchebycheff metric as the decision maker's value function.

The second area for future research is to develop more robust *a posteriori* solution techniques or metaheuristics using the ICP measure. The ICP for a set of near Pareto-optimal solutions will be a response value of a parameter optimization procedure. ICP will also be used to determine when to stop an algorithm by considering the convergence of solution quality.

The third future research topic is to embed ICP and its extensions into a computational framework that considers multiple algorithms for attacking complex multiple criteria combinatorial optimization problems. When applied over the entire range of solutions as in this paper, ICP can identify which procedure or algorithm is the best overall. Alternatively, ICP can be applied over certain sub-regions of the Pareto-optimal solution set (e.g., for solutions where the makespan is less than 200 time units or for solutions which may be optimal within some weight interval (0.4, 0.6), and etc.). And it can identify which of two or more algorithms (or parameter settings for a single algorithm) is best for problems in that region of objective space. Also, the calculations for ICP provide an optimal weight interval (or region) for each Pareto-optimal solution in the parameter space, which will suggest the robustness of the solutions. This can help to discover the decision maker's weight density function in an interactive manner.

SUMMARY

We present the use of a new measure called Integrated Convex Preference (ICP) to determine the solution quality of approximate solution algorithms for multiple objective combinatorial optimization problems. The performance of ICP is compared to that of other measures appearing in the literature by comparing the solution sets generated by two approximate solution techniques (genetic algorithms) for a bi-criteria parallel machine scheduling problem.

REFERENCES

Carlyle, W. M., J. W. Fowler, E. Gel, and B. Kim, "Evaluating the quality of approximate solution sets for multiple objective optimization problems," *Decision Sciences*, **34**(1), 63–82 (2003).

Cieniawski, S. E., J. W. Eheart, and S. Ranjithan, "Using genetic algorithms to solve a multi-objective groundwater monitoring problem," *Water Resources Research*, **31**(2), 399–409 (1995).

Cochran, J. K., S.-M. Horng, and J. W. Fowler, "A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines," *Computers & Operations Research*, **30**(7), 1087–1102 (2003).

Cunha, A. G., P. Oliveira, and J. A. Covas, "Use of genetic algorithms in multicriteria optimization to solve industrial problems," in *Proceeding of International Conference on Genetic Algorithms*, 1997, pp. 682–688.

Czyzak, P. and A. Jaszkiewicz, "Pareto simulated annealing—A metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, **7**, 34–47 (1998).

Daniels, R. L. in "Analytical evaluation of multi-criteria heuristics," *Management Science*, **38**(4), 501–513 (1992).

De, P., J. B. Ghosh, and C. E. Wells, "Heuristic estimation of the efficient frontier for a bi-criteria scheduling problem," *Decision Sciences*, **23**, 596–609 (1992).

Fonseca, C. M. and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," *Proceeding of International Conference on Genetic Algorithms*, 1993, pp. 416–423.

Fry, T. D., R. D. Armstrong, and H. Lewis, "A framework for single machine multiple objective sequencing research," *Omega International Journal of Management Science*, **17**(6), 594–607 (1989).

Hansen, P. and A. Jaszkiewiecz, "Evaluating the quality of approximations to the non-dominated set. Institute of mathematical modeling," Technical University of Denmark, Lyngby. Technical Report 1998, IMM-REP-1998-7.

Hyun, C. J., Y. Kim, and Y. K. Kim, "A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines," *Computers & Operations Research*, **25**(7/8), 675–690 (1998).

Lee, C.-Y. and G. L. Vairaktarakis, "Complexity of single machine hierarchical scheduling: A survey in complexity in numerical optimization," P. M. Pardalos (ed.), *World Scientific Publishing*, 1993, pp. 269–298.

Loughlin, D. H. and S. Ranjithan, "The neighborhood constraint method: A genetic algorithm-based multiobjective optimization technique," in *Proceeding of International Conference on Genetic Algorithms*, 1997, pp. 666–673.

Louis, S. J. and G. J. E. Rawlins, "Pareto-optimality, GA-easiness and deception," in *Proceeding of International Conference on Genetic Algorithms*, 1993, pp. 118–123.

Marett, R. and M. Wright, "A comparison of neighborhood search techniques for multi-objective combinatorial problems," *Computers & Operations Research*, **23**(5), 465–483 (1996).

Miettinen, K. M. *Nonlinear Multiobjective Optimization*, Kluwer Academic, Boston, 1999.

Murata, T., H. Ishibuchi, and H. Tanaka, "Multi-objective genetic algorithm and its application to flowshop scheduling," *Computers Industrial Engineering*, **30**(4), 957–968 (1996).

Nelson, R. T., R. K. Sarin, and R. L. Daniels, "Scheduling with multiple performance measures: The one-machine case," *Management Science*, **32**(4), 464–479 (1986).

Pinedo, M., *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, New Jersey, 1995.

Schaffer, J. D., "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceeding of the First International Conference on Genetic Algorithms*, 1985, pp. 93–100.

Schenkerman, S., "Relative objective functions in multiobjective decision support models," *Decision Sciences*, **21**, 727–737 (1990).

Serifoglu, F. S. and G. Ulusoy, "Parallel machine scheduling with earliness and tardiness penalties," *Computers & Operations Research*, **26**, 773–787 (1999).

Tuzikov, A., M. Makhaniok, and R. Manner, "Bicriteria scheduling of identical processing time jobs by uniform processors," *Computers & Operations Research*, **25**(1), 31–35 (1998).

Viana, A. and J. P. Sousa, "Using metaheuristics in multi-objective resource constrained project scheduling," *European Journal of Operational Research*, **120**, 359–374 (2000).