

Evolutionary Algorithms for Multi-Objective Optimization: Performance Assessments and Comparisons

K. C. Tan[†], T. H. Lee and E. F. Khor

Department of Electrical and Computer Engineering

National University of Singapore

10 Kent Ridge Crescent Singapore 119260

[†]Email: eletankc@nus.edu.sg

Abstract The rapid advances of evolutionary methods for multi-objective (MO) optimization poses the difficulty of keeping track of the developments in this field as well as selecting an appropriate evolutionary approach that best suits the problem in-hand. This paper aims to analyze the strength and weakness of different evolutionary methods proposed in literatures. For this purpose, ten existing well-known evolutionary MO approaches have been experimented and compared extensively on two benchmark problems with different MO optimization difficulties and characteristics. Besides considering the usual two important aspects of MO performance, i.e., the spread across the Pareto-optimal front as well as the ability to attain the global optimum or final trade-offs, this paper also proposes a few useful performance measures for better and comprehensive examination of each approach both quantitatively and qualitatively. Simulation results for the comparisons are commented and summarized.

1 Introduction

Evolutionary techniques for multi-objective (MO) optimization are currently gaining significant attentions from researchers in various fields due to their effectiveness and robustness in searching for a set of global trade-off solutions. This growing interest is reflected by the significantly increase number of different evolutionary-based approaches and variations of existing techniques published in technical literatures. There have been many survey studies on evolutionary techniques for MO optimization (Fonseca and Fleming, 1995; Coello Coello, 1996; Bentley and Wakerfield, 1997; Horn, 1997; Coello Coello, 1998; Van Veldhuizen and Lamont, 2000). Among these, (Coello Coello, 1998) is a comprehensive survey aims to summarize and organize the information on different techniques. In their works, the techniques were classified into three main groups based on different implemented strategies in cost assignments and selection methods. They were naïve approach, non-aggregation approaches that are not Pareto-based and Pareto-based approaches. In each group, a fairly detailed implementation of the methods with relevant feedback

were given, which provides readers a wide understanding and motivation to look into the mathematical programming approaches for exploiting the search capabilities of evolutionary-based MO optimization. Besides, there were also other studies that attempt to classify existing approaches in different ways. For example, Fonseca and Fleming (1995) classified existing techniques from a broad algorithmic perspective, Bentley and Wakerfield (1997) from perspective of range dependency while others (i.e. Horn, 1997; Van Veldhuizen and Lamont, 2000) from decision maker's perspective.

This paper attempts to analyze the strength and weakness of different well-known evolutionary MO optimization algorithms based upon benchmark problems. Section 2 provides a general overview and features of exiting evolutionary approaches for MO optimization. The performance measures is given in Section 3, and Section 4 describes the test problems with different MO optimization difficulties and characteristics used in this comparison study. The simulation results and performances of various algorithms are compared and summarized in Section 5. Conclusions are drawn in Section 6.

2 Evolutionary Approaches for MO Optimization

In general, multi-objective (MO) optimization can be defined as the problem of optimizing a vector of non-commensurable and often competing objectives or cost functions, viz, it tends to find a parameter set \mathbf{P} for

$$\underset{\mathbf{P} \in \mathbf{F}}{\text{Min}} \mathbf{F}(\mathbf{P}) \quad (1)$$

where $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$ is a individual vector with n parameters and \mathbf{F} defines a set of individual vectors. $\{f_1, f_2, \dots, f_m\}$ are m objectives to be minimized and $\mathbf{F} = \{f_1, f_2, \dots, f_m\}$. Instead of a single optima, solution to MO optimization problem is often a family of points known as Pareto optimal set, where each objective component of any point along the Pareto-front can only be improved by degrading at least one of its other objective components (Richardson *et al.*, 1989; Srinivas and Deb, 1994). Evolutionary algorithms have been studied and

recognized to be well suited for MO optimization problems (Deb and Goldberg, 1989; Forrest *et al.*, 1993). Emulating the biological evolution mechanism and Darwin's principal on 'survival-of-the-fittest', evolutionary algorithms (EAs) have recently been growing in popularity owing to its powerful features in search adaptation and global optimization. Unlike conventional gradient-guided search methods, EAs require no gradient information, which makes it a unique and robust tool for solving multi-objective (MO) optimization problem.

In this section, existing evolutionary-based approaches for MO optimization were looked in different perspective

from previous surveys. Instead of trying to classify the existing approaches, the feature elements of each existing approach were extracted and discussed. In addition, more related citations, including the latest, were included in this discussion. Generally the MO handling technique for each algorithm can be broken down into at least one MO handling element. These elements cannot be split into simpler element since they themselves represent the basis of MO handling tools in evolutionary techniques. This finding is depicted in Table 1, which lists out all the MO handling elements available in existing evolutionary techniques. They are *Weights*, *Min-Max*, *Pareto*, *Ranking*, *Goals*, *Pref.*, *Gene*, *Sub-pop.*, *Fuzzy*, *Agents* and *Others*.

Table 1: Feature elements in MO evolutionary approaches

Label		Descriptions
MO Handling Elements	<i>Weights</i>	Multiple objectives are combined into scalar objective via weight vector. Weights may be assigned through: direct assignment, eigenvector method, entropy method, minimal information method, randomly determined or adaptively determined. It is difficult to precisely pre-determine the weights. If the objective functions are simply weighted and added to produce a single fitness, the function with the largest range would dominate evolution. A poor input value for the objective with the larger range makes the overall value much worse than a poor value for the objective with the smaller range (Bently and Wakefield, 1997). It suffers the disadvantage of missing concave portions of the trade-off curve (Coello Coello, 1996).
	<i>Min-Max</i>	It uses the distance between an efficient design and a pre-defined ideal design. It attempts to find from the feasible domain an efficient design which is nearest to the ideal design in the minimax sense. It is able of discovering all efficient solutions of a multiobjective problem whether the problem is convex or non-convex.
	<i>Pareto</i>	It uses Pareto dominance scheme (Fonseca and Fleming, 1993) for individual comparison. The comparison results among the individuals will influence the selection and reproduction process in the evolution.
	<i>Ranking</i>	Individuals are sorted from the most preferable to the least preferable, or vice versa. Individuals are then assigned rank according to their preferences. If two or more individuals are equally preferable, they will be assigned the same rank.
	<i>Goals</i>	It requires a designer to set goals for the objectives that he wishes to achieve and adopts the decision rule that the best compromise design should be the one which minimizes the deviation from the set goals.
	<i>Pref.</i>	It requires a designer to set preferences/priorities of the objectives to optimize and adopts the decision rule that the objectives with higher priorities are given higher privilege to optimize than the objectives with lower priorities.
	<i>Gene</i>	Chromosome genes do not only store the information of decision variables or parameter values for each individual but they also influence the way where fitness/cost assignment process for each individual is performed. The genes for the latter purpose can be either altered stochastically through normal/special evolution process or assigned through deterministic rule.
	<i>Sub-pop.</i>	The main population is divided into several sub-populations where each sub-population is optimized based on similar/different selection criteria. If different selection criteria are applied, it may refer to either different objective component or utility function of the objectives. The shuffling and merging of all sub-populations are in fact corresponding to fitness averaging for each of the objective component (Richardson <i>et al.</i> , 1989). If gender is applied to classify the population, a relative large population size with lots of computational effort is required to maintain a reasonably diverse spread of genders across the entire population (Coello Coello, 1998).
	<i>Fuzzy</i>	Fuzzy rules and fuzzy membership functions are applied to combine the multiple objectives to handle the vague term of user's specifications. The resulted fuzzy reasoning process is then used in selection process.
	<i>Agents</i> <i>Others</i>	It involves the use of a family of agent where each agent participates in improving the individuals for its corresponding objective. The objective handling techniques that are apart from the above techniques.

Fig. 1 illustrates the development trends of MO handling elements starting from the first publication of MO evolutionary approaches until the most recent ones, according to the authors' best knowledge. The y-axis represents the cumulative number of methods applying the corresponding MO handling elements along the year of development, while the x-axis represents the years which is not according to linear scale. As can be seen from the figure that, the MO handling element of *goals* is the earliest while *Agents* is the latest to be applied. As time goes on, the elements of *Min-Max*, *Gene*, *Sub-pop.*, *Fuzzy*, *Agents*, and *Others* receive less interest from

researchers as compared to the elements of *Weights*, *Pareto*, *Ranking*, *Goals* and *Pref.* Among the former set of elements, *Weights* has attracted significant interest from researchers between the period of 1985-1997 and was the most popular method before 1997 (Fonseca and Flaming, 1997). On the other hand, the popularity of *Pareto* as MO handling elements continues to grow at the highest rate since 1987 (when it was first proposed by Goldberg and Richardson (1987)) as compared to others, and it is believed to be continuously growing in the future.

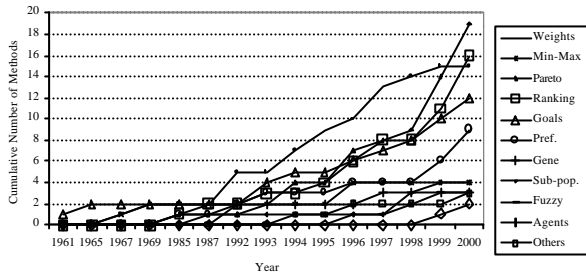


Fig. 1 Development trends of objective handling elements

3 Performance Measures

Until 1998, there is a lack of well-established approach for quantitative measure of performance in multi-objective evolutionary algorithms. The arguments for and against the methods are based on qualitative aspects of the evolution process and population distribution. Since then, a few works (Van Veldhuizen and Lamont, 1999; Deb, 1999a; Shaw *et al.*, 1999; Zitzler and Thiele, 1999) began to explore the field of MOEA comparison and tried to formalize the quantitative performance measures as a common comparative basis. However, as stated by (Shaw *et al.*, 1999), there are only two aspects being covered: the first is according to their spread across the available trade-off surface while the second is the ability in terms of attainment of optimum goal in multi-objective sense. In this paper, a set of performance measures is recommended and applied to compare various evolutionary MO methods. Besides covering the above two aspects of performance, the proposed measures also consider the fraction of useful individuals in the population in multi-objective sense, the uniform distribution of individuals on the found trade-off surface as well as the computational effort of the optimization method.

There are all together five different quantitative measures of MO optimization performances used in this paper. Some of these measures are referred from other articles including Veldhuizen and Lamont, (1998); (1999); Zitzler and Thiele, (1999), while others are designed carefully in this paper for a more comprehensive comparison. Note that these measures are chosen since they are widely used for performance comparisons in MO optimization. Furthermore, these measures need not compute the actual trade-off that often needs to be obtained through deterministic enumeration or is not always practically implementable in some problems (Veldhuizen and Lamont, 1999).

i) Algorithm Effort(AE)

The algorithm effort (AE) can be defined as the ratio of the total number of function evaluations N_{eval} over a fixed period of simulation time T_{run} ,

$$AE = \frac{T_{run}}{N_{eval}} \quad (2)$$

As shown in eqn. 2, for a fixed period of T_{run} , more number of function evaluations being performed indirectly indicates that less computational effort is required by the optimization algorithm and hence resulting in a smaller AE. Similarly, for a fixed value of N_{eval} , any optimization algorithm that requires a bigger value of T_{run} reflects the need of a larger algorithm effort, and vice versa.

ii) Ratio of Non-dominated Individuals (RNI)

For more choices of non-dominated solutions, it is always desired to have as many as possible the useful candidate solutions known as the Pareto-front from a given population size. The performance measure is denoted here as the ratio of non-dominated individuals (RNI) for a given population X and is mathematically formulated as:

$$RNI(X) = \frac{nondom_indiv}{P} \quad (3)$$

where $nondom_indiv$ is the number of non-dominated individuals in population X while P is the size of population X . Therefore the value $RNI = 1$ means all the individuals in the population are non-dominated while the opposite, $RNI = 0$ represents the situation where none of the individuals in the population are non-dominated.

iii) Size of Space Covered (SSC)

Zitzler and Thiele (1999) has proposed a quantitative measure of size of space covered (SSC) to evaluate the overall size of phenotype space covered by all the optimized solutions.

iv) Uniform Distribution (UD)

A measure of uniform distribution (UD) to measure the distribution of non-dominated individuals is proposed here. Mathematically, $UD(X')$ for a given set of non-dominated individuals X' in a population X , where $X' \subseteq X$, is defined as,

$$UD(X') = \frac{1}{1 + S_{nc}} \quad (4)$$

where S_{nc} is the standard deviation of niche count of the overall set of non-dominated individuals X' , and is formulated as,

$$S_{nc} = \sqrt{\frac{\sum_i^{N_{X'}} (nc(x'_i) - \overline{nc}(X'))^2}{N_{X'} - 1}} \quad (5)$$

where $N_{x'}$ is the size of the set X' ; $nc(x'_i)$ is the niche count of i -th individual x'_i where $x'_i \in X'$; and $\overline{nc}(X')$ is the mean value of $nc(x'_i)$, $\forall i = 1, 2, \dots, N_{x'}$, as shown in the following equations,

$$nc(x'_i) = \sum_{j, j \neq i}^{N_{x'}} f(i, j), \quad (6a)$$

$$\text{where } f(i, j) = \begin{cases} 1, & \text{dis}(i, j) < \mathbf{s}_{share} \\ 0, & \text{else} \end{cases} \quad (6b)$$

$$\overline{nc}(X') = \frac{\sum_{i=1}^{N_{x'}} nc(x'_i)}{N_{x'}} \quad (7)$$

where $dis(i, j)$ is the distance between individual i and j in the objective domain.

v) Coverage of Two Set (C)

The Coverage of Two Set (C) is a measure to compare the domination of two populations in a pair-wise manner, i.e., how good population i dominates population j as well as how good population j dominates population i . This measure was proposed by Zitzler and Thiele (1999).

4 Test Problems

Concerning the formation of appropriate test problems for multi-objective evolutionary algorithms, Deb (1999b) has pointed out a few features that cause difficulties in converging to the Pareto-optimal front and maintaining population diversity in the current non-dominated front. These features include multi-modality, deception, isolated optimum and collateral noise. In this paper, some of the above features are accounted in our test functions.

(i) Test Problem 1

In test problem 1, the Fonseca's two-objective minimization problem which has been widely studied in (Fonseca and Fleming, 1993) is applied here. Besides its non-convex Pareto optimal front, this test function is chosen since it has large and non-linear trade-off curve that should challenge the MO evolutionary algorithm's ability to find and maintain the entire front uniformly. Besides, it is easy used for visualization and comparison. The two-objective functions, f_1 and f_2 , to be minimized are given as

$$f_1(x_1, \dots, x_8) = 1 - \exp\left(-\sum_{i=1}^8 \left(x_i - \frac{1}{\sqrt{8}}\right)^2\right) \quad (8a)$$

$$f_2(x_1, \dots, x_8) = 1 - \exp\left(-\sum_{i=1}^8 \left(x_i + \frac{1}{\sqrt{8}}\right)^2\right) \quad (8b)$$

where $-2 \leq x_i < 2$, $\forall i = 1, 2, \dots, 8$.

(ii) Test Problem 2

In this test problem, the search algorithms are evaluated in the noisy environment to test their robustness in the sense that the disappearance of important individuals from the population has little effect on the global system behavior (Collard and Escazut, 1995). In order to investigate the relative abilities of MO search algorithms in noisy environment, noisy version of two-objective optimization with three variables is constructed here where the function being optimized contained elements of noise as formulated below:

$$f_1 = x'_1, \quad (9a)$$

$$f_2 = \frac{1}{x_1} \left\{ 1 + (x'_2{}^2 + x'_3{}^2)^{0.25} [g + 1.0] \right\}, \quad (9b)$$

$$g \sin^2 \left(50 (x'_2{}^2 + x'_3{}^2)^{0.1} \right) \quad (9c)$$

Instead of performing the optimization on real-valued parameters, x_i , the optimization is performed on the 'corrupted' parameters with additive noise element:

$$x'_i = x_i + N(\mathbf{s}, \mathbf{m}), \quad (9d)$$

where $0.1 \leq x_1 \leq 1$ and $-100 \leq x_i \leq 100 \forall i = 2, 3$.

$N(\mathbf{s}, \mathbf{m})$ is the white noise. The population distribution density of the noise is given as normal distribution (Grimm, 1993) as indicated in eqn. 10:

$$P(x | N(\mathbf{s}, \mathbf{m})) = \frac{1}{\sqrt{2\pi\mathbf{s}^2}} \exp\left(-\frac{(x - \mathbf{m})^2}{2\mathbf{s}^2}\right) \quad (10)$$

where \mathbf{m} and \mathbf{s} are the mean and variance of the probability density distribution. In the normal curve, approximately 68% of the scores of the distribution lie between $\mathbf{m} \pm \mathbf{s}$. In this test problem, both \mathbf{m} and \mathbf{s} are given as 0 and 0.1, respectively.

5 Performance Comparisons

In this section, various evolutionary MO optimization methods have been compared using the benchmark problems in Section 4, and based upon the performance measures in Section 3. The MO evolutionary optimization

methods under studied include VEGA from (Schaffer, 1985), MIMOGA from (Murata and Ishibuchi, 1995), HLGGA from (Hajela and Lin, 1992), NPGA from (Horn *et al.*, 1994), MOGA from (Fonseca and Fleming, 1993), NSGA from (Srinivas and Deb, 1994), SPEA from (Zitzler and Thiele, 1999), MOEA from (Tan *et al.*, 1999), IMOEA from (Tan *et al.*, 2000) and EMOEA from (Tan *et al.*, 2001). Most of these methods have often been applied or taken as reference in literature for comparing different population-based multi-objective evolutionary algorithms, and hence they are chosen here for the study.

In the simulations, the setting of parameter values are based upon two principles: 1) the values of the parameter that are commonly used by several algorithms are identical in those algorithms and, 2) the values of the parameter that are used in specific algorithm are decided based upon the recommended values from their original literature. Phenotype sharing is applied in all algorithms that apply the sharing operation. The sharing distance for HLGGA, NPGA, MOGA, NSGA as well as the performance measure of *UD* for all methods are set as 0.01 in the normalized space. Since dynamic sharing (Tan *et al.*, 1999) was used for both MOEA and IMOEA, the sharing distance is computed dynamically at each generation, whereas no distance parameter is needed for SPEA and EMOEA as proposed by (Zitzler and Thiele, 1999) and (Tan *et al.*, 2001). Tournament selection scheme with tournament size of 2 is used in MOGA, SPEA, MOEA IMOEA and EMOEA as suggested in their respective literatures. The Pareto tournament selection scheme with $t_{dom} = 10\%$ of the population size is used in NPGA for a tight and complete population distribution as recommended by (Horn *et al.*, 1994). In order to guarantee a fair comparison, all algorithms considered are implemented with the same coding scheme, crossover and mutation. Note that each parameter is represented by 3-digit decimal and concatenated to form the chromosomes, which gives a shorter chromosome length and avoids the *Hamming-cliff* effect as encountered in traditional binary-based coding scheme (Tan *et al.*, 1999). In all cases, standard mutation with a probability of 0.01 and standard crossover with two-point crossover and a probability of 0.7 are used.

All methods under comparison were implemented with the same common sub-functions using the same programming language in Matlab (The Math Works, 1998) on an Intel Pentium II 450 MHz computer. Each of the simulation was terminated automatically when a fixed pre-specified simulation period (for each test problem) is reached, in the same platform that is free from other computation or being interrupted by other programs. The period for all algorithms being compared for test problems 1 and 2 are 180 and 100 sec., respectively. These were pre-determined based on the criteria that they were found most appropriate to clearly observe the difference of simulation results among the various

methods, in which at least one method has converged satisfactory. 30 independent simulation runs have been performed for each method in each test problem so as to study the statistical performance such as consistency and robustness of the methods. Here, a random initial population was created for each of the 30 runs. Except SPEA and EMOEA which include the secondary population, the population size of 100 is applied on test problems 1 while population size of 30 on 2. For SPEA and EMOEA, four combinations of {primary population size, secondary population size} are applied. {100, 20} on test problems 1 and {30, 10} on 2. All the methods being compared are indexed according to the sequence listed in first paragraph of this section.

Fig. 2 summarizes the performances of each algorithm in each test problem with respect to the number of function evaluations (N_{eval}), algorithm effort (AE), ratio of non-dominated individuals (RNI), size of space covered (SSC) and uniform distribution (UD) of non-dominated population. The distribution simulation data of 30 independent runs is represented in box plot format, which has been applied by (Zitzler and Thiele, 1999) to visualize the median, upper quartile, lower quartile, outside value for each distribution of the simulation data. In each graph, the sequence of box plots from left to right is based upon the above mentioned indexes of algorithms.

In the aspect of N_{eval} and AE per run under a fixed CPU time, it can be seen from the first and second columns of Fig. 2 that on almost all test problems, VEGA and MIMOGA have a relatively high N_{eval} and low AE . This indicates that these algorithms are less computational expensive as compared to others and hence more iterations were being performed per run within the fixed period of CPU time. On the other hand, SPEA is observed to give the highest algorithm effort (especially on test problems 1 and 2) since they have a lowest N_{eval} and highest AE on all test problems. The large computational effort of SPEA could mainly be due to its clustering algorithm. This is followed by EMOEA. Meanwhile, the N_{eval} and AE for HLGGA, NPGA, MOGA, NSGA, MOEA and IMOEA are moderate, the computational effort of SPEA is, however, comparatively large.

Concerning the RNI (3rd column in Fig. 2), MOEA and IMOEA have the highest RNI on almost all the test problems (except test problem 2) as compared to others. This clearly indicates their abilities to provide more non-dominated solutions from a given size of population based on the incorporation of switching preserved strategy (SPS) (Tan *et al.*, 1999) where all the non-dominated individuals are preserved for the next generation. The RNI for SPEA and EMOEA are mainly depending on the ratio of the size of secondary (best-found) population to the size of main population. For the rest of algorithms where no elitism strategy is applied, their values of RNI are not in clear difference and are relatively low in general, especially on test problems 1 and 2.

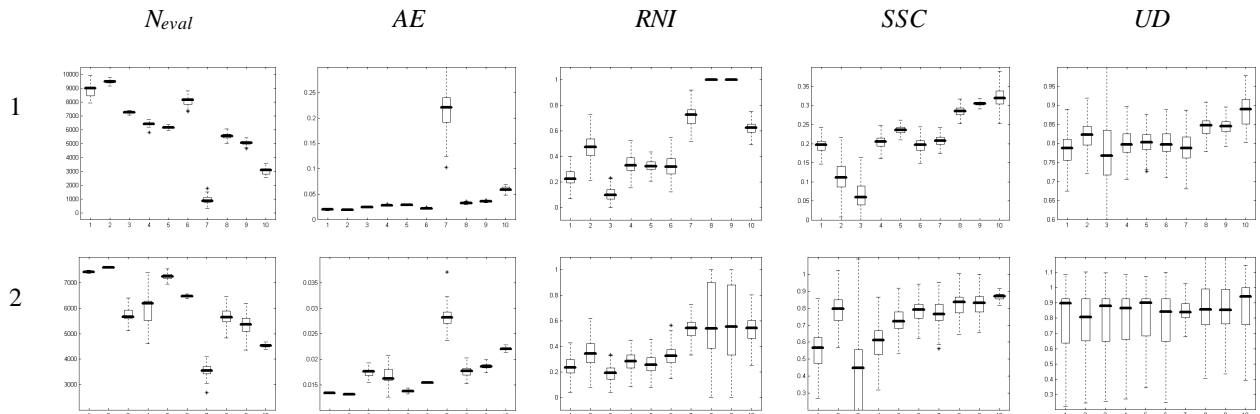


Fig. 2 Box plots for the measures of number of function evaluations (N_{eval}), algorithm effort (AE), ratio of non-dominated individuals (RNI), size of space covered (SSC) and uniform distribution (UD).

In SSC , the performances of MOEA, IMOEA and EMOEA in approaching the actual trade-off region are outstanding in almost all test problems as compared to other algorithms. As for the rest of the algorithms, no significant difference of the SSC values is observed. In the context of UD , the performance of all the methods are about the same on test problem 2 (noisy environment). This indicates that in noisy environment, none of the algorithm is significantly superior to the others. It can also be observed that EMOEA shows its superiority on test problems 1 for non-convex trade-off curve. Besides IMOEA and MOEA, SPEA has also shown to be performing well on many test problems, especially test problems 1 for the measure of SSC .

The performance measures of $C(X_i, X_j)$ for the comparison sets between algorithms i and j , where $i, j = 1, 2, \dots, 10$, are shown in Fig. 3. Again, box plots are used to summarize the sample distributions of 30 independent runs for each case. In each rectangle containing box plots, the sequence of box plots from left to right is based on the same indexes of algorithm as mentioned above. The ranges of y - and x - axis of each graph are $[0, 1]$ and $[1, \text{number of compared algorithms in the respective test problem}]$. As can be seen, $C(X_i, X_j)$ for $i = j$ always takes the value of zero since two identical populations cannot dominate each other. Generally, there is no clear evidence that, for all the test problems, the population of any of the methods totally dominating any other method(s) in all the 30 runs since, for all test problems, there appears to be no cases where $C(X_i, X_j) = 0$, and $C(X_j, X_i) = 1$ for $i \neq j$.

From these experiments, it can be concluded that although VEGA and MIMOGA require less computational effort, and are able to perform more number of iterations in a given fixed period of CPU time, their performance to discover the entire trade-off curve uniformly are less superior than other methods. This can be evident by their measures of SSC and UD in Fig. 2 and C in Fig. 3. This may be due to the absence of preserved strategy in these algorithms as well as other genetic

operations that are different from other methods. The similarity of both methods is that there are no explicit operators to diversify the population along the discovered trade-off curve. The results also show that the methods with preserved strategy, such as SPEA, MOEA, IMOEA and EMOEA have better convergence and population distribution along the discovered trade-off region. The implementation of preserved strategy may significantly burden the computation and increase the algorithm effort.

6 Conclusions

Existing evolutionary algorithms for MO optimization have been surveyed and classified in this paper based upon the different features in each of the approaches. Ten existing well-known evolutionary MO approaches have been experimented and compared extensively on benchmark problems with different MO optimization difficulties and characteristics. Besides considering the usual two important aspects of MO performance, i.e., the spread across the Pareto-optimal front as well as the ability to attain the global optimum or final trade-offs, this paper has also proposed other performance measures.

The overall simulation results show that none of the methods is the most superior considering all aspects of the performance measures. Some conclusions can, however be drawn based upon the basis of each feature element applied in the algorithm. In general, preserve strategy (as implemented in SPEA, MOEA, IMOEA and EMOEA) is important for better convergence and population distribution along the discovered trade-off region in MO optimization. However, this requires more algorithm effort and hence results in less function evaluations within a fixed period of CPU time. Therefore preserve strategy can be greatly beneficial for problems with heavy function evaluation since its effect in the algorithm effort becomes less insignificant.

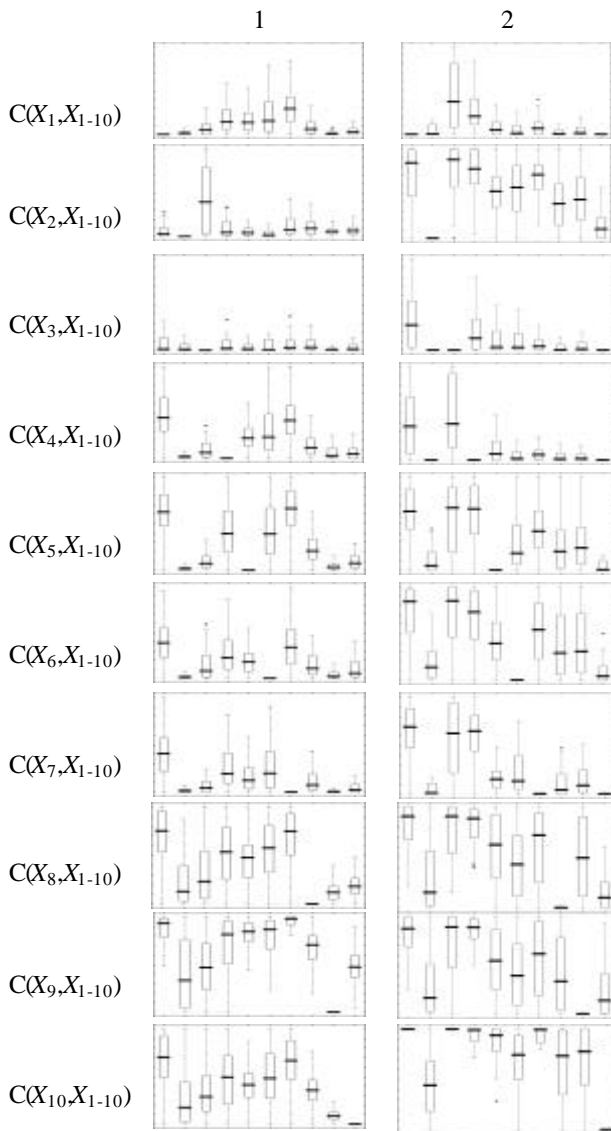


Fig. 3 Box plot for the C measure. Each rectangle refers to the measure of $C(X_i, X_{1-10})$

This study forms a good basis to match the existing approaches to the performance measures as well as the features of the optimization problems. This shall contribute to the effort of combining the promising elements from different approaches into a new approach that performs well for the current type of optimization problem we are dealing with. However, further investigation is needed to consider more kinds of test problems with different MO optimization difficulties, such as the heavy function evaluation, heavy constraint, dynamic search space and etc., Results and progress will be reported in due course.

Bibliography

- Bentley, P. J. and Wakefield, J. P. (1997). Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. In *Proceedings of the 2nd On-Line World Conference on Soft Computing in Engineering Design and Manufacturing (WSC2)*, (<http://users.aol.com/docbentley/dispaper.htm>).
- Coello Coello C. A. (1998). *An updated survey of GA-based multiobjective optimization techniques*. Technical Report: Lania-RD-98-08, Laboratorio Nacional de Informatica Avanzada (LANIA), Xalapa, Veracruz, Mexico.
- Coello Coello, C. A. (1996). *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*. Ph.D. Thesis, Department of Computer Science, Tulane University, New Orleans, LA.
- Collard, P., and Escazut, C. (1995). Genetic operators in a dual genetic algorithm. *International Conference on Tools and Artificial Intelligence*, pp. 12-19.
- Deb, K. (1999a). Construction of test problems for multi-objective optimization. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, vol. 1, pp. 164-171.
- Deb, K. (1999b). Multi-objective genetic algorithms: Problem difficulties and construction of test problem. *Journal of Evolutionary Computation*, The MIT Press, vol. 7(3), pp 205-230.
- Deb, K. and Goldberg, D. E. (1989). An investigation on niche and species formation in genetic function optimization. *Proc. of Third Int. Conf. on Genetic Algorithms*, J. D. Schaffer, ed., San Mateo, CA: Morgan Kaufmann, pp. 42-50.
- Fonseca, C. M. and Fleming, P.J. (1993). Genetic algorithm for multiobjective optimization, formulation, discussion and generalization. In *Forrest, S., ed., Genetic Algorithms: Proceeding of the Fifth International Conference*. CA, pp. 416-423.
- Fonseca, C. M., and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, vol. 3, no. 1, pp. 1-16.
- Fonseca, C. M., and Fleming, P. J. (1997). Multiobjective optimization. In Bäck, T., Fogel, D., and Michalewicz, Z. eds., *Handbook of Evolutionary Computation*, vol. 1, Oxford University Press, pp. C4.5:1-C4.5:9.
- Forrest, S., Javornik, B., Smith, R. E., and Perelson, A. S. (1993). Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, MIT Journals, vol. 1, no. 3, pp 191-211.
- Goldberg, D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multi-modal function

- optimization. *Proc. Second Int. Conf. on Genetic Algorithms*, pp. 41-49, Lawrence Erlbaum.
- Grimm, L. G. (1993). *Statistical application for behavioral sciences*. J. Wiley, New York, pp. 99.
- Hajela, P., and Lin, C. Y. (1992). Genetic search strategies in multicriterion optimal design. *Journal of Structural Optimization*, vol. 4, pp. 99-107.
- Horn, J. (1997). Multicriterion decision making. In Bäck, T., Fogel, D., and Michalewicz, Z. eds., *Handbook of Evolutionary Computation*, Oxford University Press, Oxford, England, vol. 1, pp. F1.9:1-F1.9:15.
- Horn, J., Nafpliotis, N. and Goldberg, D.E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. *IEEE international Conference on Evolutionary Computation*, Vol. 1, pp. 82-87.
- Murata, T., and Ishibuchi, H. (1995). MOGA: Multi-objective genetic algorithms. *IEEE International Conference on Evolutionary computation*, vol. 1, pp. 289-294.
- Richardson, J. T., Palmer, M. R., Liepins, G., and Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions. In J.D. Schaffer ed., *Proc. of Third Int. Conf. on Genetic Algorithms*, pp. 191-197
- Schaffer, J. D. (1985). Multi-objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100, Lawrence Erlbaum.
- Shaw, K. J., Notcliffe, A. L., Thompson, M, Love, J., Fonseca, C. M., and Fleming, P. J. (1999). Assessing the performance of multiobjective genetic algorithms for optimization of batch process scheduling problem. *IEEE International Conference on Evolutionary Computation*, vol. 1, pp. 37-45.
- Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, MIT Press Journals, vol. 2, no. 3, pp. 221-248.
- Tan, K. C., Lee, T. H. and Khor, E. F. (1999). Evolutionary algorithms with goal and priority information for multi-objective optimization. *IEEE International Conference on Evolutionary Computation*, vol. 1, pp. 106-113.
- Tan, K. C., Lee, T. H. and Khor, E. F. (2001). Tabu-based exploratory evolutionary algorithm for effective multi-objective optimization. *Accepted by First Conference on Evolutionary Multi-Criterion Optimization, Zurich, Switzerland*.
- Tan, K. C., Lee, T. H., Khor, E. F., and Sathikannan. R. (2000). Incremented multi-objective evolutionary design automation of robust tracking thumbprint performances in QFT. *International Conference on Evolutionary Computing for Computer, Communication, Control and Power (ECCAP2000)*, pp. 137-142.
- The Math Works, Inc. (1998). *Using MATLAB*, The Math Works Inc., Version 5.
- Van Veldhuizen, D. A., and Lamont G. B. (1999). Multiobjective evolutionary algorithm test suites. *Symposium on Applied Computing*, San Antonio, Texas, pp. 351-357.
- Van Veldhuizen, D. A., and Lamont G. B. (2000). Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Journal of Evolutionary Computation*, The MIT Press, vol. 8, no. 2, pp. 125-147.
- Zitzler, E., and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp.257-271.