12-20-2005

An Evolutionary Multi-Objective Crowding Algorithm (EMOCA): Benchmark Test Function Results

Ramesh Rajagopalan
Syracuse University, rarajago@syr.edu

Chilukuri K. Mohan
Syracuse University, ckmohan@syr.edu

Kishan G. Mehrotra
Syracuse University, mehrotra@syr.edu

Pramod K. Varshney
Syracuse University, varshney@syr.edu

Recommended Citation

R. Rajagopalan, C. K. Mohan, K. G. Mehrotra, and P. K. Varshney, "An evolutionary multi-objective crowding algorithm (EMOCA): Benchmark test function results," in 2nd Indian International Conference on Artificial Intelligence, IICAI 2005, December 20, 2005 - December 22, 2005, Pune, India, 2005, pp. 1488-1506.

An Evolutionary Multi-Objective Crowding Algorithm (EMOCA): Benchmark Test Function Results

Ramesh Rajagopalan, Chilukuri K. Mohan, Kishan G. Mehrotra, and

Pramod K. Varshney

Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, NY 13244-4100,USA
{rarajago, ckmohan, mehrotra, varshney}@syr.edu

Abstract.

A new *evolutionary multi-objective crowding algorithm (EMOCA)* is evaluated using nine benchmark multi-objective optimization problems, and shown to produce non-dominated solutions with significant diversity, outperforming state-of-the-art multi-objective evolutionary algorithms viz., Non-dominated Sorting Genetic Algorithm – II (NSGA-II), Strength Pareto Evolutionary algorithm II (SPEA-II) and Pareto Archived Evolution Strategy (PAES) on most of the test problems. The key new approach in EMOCA is to use a diversity-emphasizing probabilistic approach in determining whether an offspring individual is considered in the replacement selection phase, along with the use of a non-domination ranking scheme. This approach appears to provide a useful compromise between the two concerns of dominance and diversity in the evolving population.

1. Introduction

Many scientific and engineering applications require simultaneous consideration of two or more objectives that must be optimized together while evaluating the tradeoffs between these objectives. Considerable research efforts have recently been devoted to developing efficient algorithms for solving such *Multi-Objective Optimization* (*MOO*) problems.

Classical optimization methods suggest converting the multi-objective optimization problem into a single objective optimization problem, e.g., by attempting to minimize a weighted sum of the various objective functions, using weights that represent relative "preference strengths." However, this approach yields only a single solution rather than a collection of non-dominated or Pareto-optimal solutions[1] corresponding to different tradeoff points. In addition, this approach cannot be used if the relative importance or weights associated with different objectives are not known. Iterating the weighted approach with different weight values is also unsuccessful in discovering the non-dominated solutions if the Pareto-optimal front is non-convex [5]. Considerable success in solving MOO problems has been obtained in recent years by using *Multi-Objective Evolutionary Algorithms (MOEAs*), e.g., [1], [2], [3], [4], [6]; a comprehensive introduction to evolutionary multi-objective optimization is given in [5]. Their success in addressing MOO problems arises from the fact that they work with a population of candidate solutions,

[1] If each objective function f_j is to be maximized, then a solution vector ***a*** *dominates* ***b***, written $a » b$, if and only if $\forall i \in \{1,\ldots m\} : f_i(\mathbf{a}) \geq f_i(\mathbf{b})$, and $\exists j \in \{1,\ldots m\} : f_j(\mathbf{a}) > f_j(\mathbf{b})$. A solution is *Non-dominated* or *Pareto-optimal* if it is not dominated by any other candidate solution. The *Pareto-optimal front* or *Pareto set* is the collection of all Pareto-optimal solutions for a problem.

exploring multiple non-dominated solutions in parallel, whereas traditional approaches are crippled by focusing on one solution at a time. MOEAs have been found to be successful in handling complex problem features such as discontinuity, multi-modality and disjoint objective spaces, and are evaluated based on their ability to find Pareto-optimal solutions, as well as discover well-distributed and widely spread solutions that represent the variety of tradeoff points inherent in the problem description.

Recently, we formulated some practical path planning problems as MOO problems and explored their solution using *MOEA*s; in the process, we formulated a new *evolutionary multi-objective crowding algorithm (EMOCA),* using a probabilistic approach that considers both domination and diversity criteria in determining whether an offspring individual is considered for the replacement selection phase [7,8]. This approach appears to provide a useful compromise between the two concerns of dominance and diversity in the evolving population, and we subsequently found that it was successful in other real world applications such as sensor placement and mobile agent routing in sensor networks [9,10]. This led to the question of whether EMOCA is of general applicability, and how it compares to other well-known and established *MOEA*s on benchmark problems. This paper addresses this question, evaluating the performance of EMOCA on several test functions suggested by Deb et al. [5] and Zitzler et al. [11]. We compare the performance of EMOCA with NSGA-II, SPEA-II and PAES using metrics that evaluate convergence and diversity of solutions, on nine benchmark problems used elsewhere to compare *MOEA*s. Our simulation results indicate that EMOCA outperforms the other algorithms in most of the test problems. In the light of the “No-Free-Lunch” theorems, it is impossible to argue for the superiority of any single algorithm over others, but we suggest that EMOCA is an important first candidate to be considered when one is hungry for a good algorithm to a difficult MOO problem.

In Section 2, we review existing literature on some recently proposed multi- objective evolutionary algorithms. In Section 3, we describe EMOCA in detail. The benchmark test problems used in our simulations are given in Section 4. Simulation results and conclusions are presented in Sections 5 and 6, respectively.

2. Evolutionary Multi-objective Optimization Algorithms

*MOEA*s usually employ Pareto-based fitness assignment to guide the search towards the true Pareto-optimal front. Density estimation methods such as crowding distance and squeeze factor are used to preserve population diversity [6]. We briefly review three state-of-the-art *MOEA*s – PAES, NSGA-II and SPEA-II which incorporates these features. For details, readers are encouraged to refer to the original studies.

2.1 Pareto Archived Evolutionary Strategy (PAES)

Knowles et al. [1] developed a *MOEA* called PAES. PAES uses a (1+1)-ES. PAES is a local search algorithm that simulates a random mutation hill climbing strategy. In PAES, the offspring is compared with the parent in each iteration. If the offspring dominates the parent, the offspring is accepted as the parent for the next iteration. If the parent dominates the offspring, the offspring is discarded and a new mutated solution is created. If the parent and offspring are mutually non-dominating, the decision is made by comparing the parent and offspring with an archive of best solutions found so far. PAES performs better in problems having search space with non-uniformly dense solutions. For our simulations, we used PAES code from the following website: http://www.rdg.ac.uk/~ssr97jdk/multi/PAES.html

2.2 Strength Pareto Evolutionary Algorithm II (SPEA-II)

Zitzler et al. have proposed SPEA-II [4] as an improved version of SPEA [3] and showed that SPEA-II obtains better performance than SPEA over all test problems considered in [4]. Each individual in SPEA-II is assigned a strength value which represents the number of solutions it dominates. The fitness of an individual is the sum of strengths of the individuals that dominate the current one. Binary tournament selection is employed to generate the mating population. The individuals with identical fitness are compared based on the *density*, a decreasing function of the distance to the k^{th} nearest data point. SPEA-II is a generational algorithm with an elitist strategy which maintains an archive of non-dominated individuals at each generation. A truncation method is used in the elitist archive to maintain a constant number of elitists. For our simulations, we used SPEA-II code from the following website: http://www.tik.ee.ethz.ch/pisa/

2.3. Non-dominated Sorting Genetic Algorithm II (NSGA-II)

NSGA –II [2] assigns a Pareto rank to each individual based on a non-dominated sorting approach. This approach performs a non-dominated sorting of individuals in the population and identifies different non-dominated fronts $F_1, F_2, \ldots F_k$, where elements of F_1 are non-dominated, and each element of F_{i+1} is dominated only by some elements of $F_1, F_2, \ldots, F_i$. NSGA-II employs binary tournament selection based on Pareto rank of the individuals with ties broken based on the crowding distance. The crowding distance method is employed to estimate the diversity of a solution. The algorithm combines attractive features such as elitism, fast non-dominated sorting and parameter less fitness sharing. NSGA- II has been widely used in several real world optimization problems. For our simulations, we used NSGA-II code from the following website: www.iitk.ac.in/kangal/soft.htm

3. Evolutionary multi-objective crowding algorithm (EMOCA)

An efficient MOEA should produce several widely spread and uniformly spaced solutions while guaranteeing convergence to Pareto-optimal front. Figure 1 shows the general overview of EMOCA. The individual steps of EMOCA are then described below.

1. Initialize;
2. For the number of iterations determined by computational bounds, do:
    - 2.1. Generate Mating Population;
    - 2.2. Generate offspring by crossover followed by mutation;
    - 2.3. Create a new pool consisting of parents and some offspring;
    - 2.4. Trim new pool to generate the population for the next iteration;
    - 2.5. Update archive to contain all non-dominated solutions

Figure 1: EMOCA algorithm overview

3.1 Mating Population Generation

EMOCA employs binary tournament selection to fill the mating population . Each solution is assigned a fitness value equal to its *total rank* defined as the sum of its *non-domination rank* and *diversity rank*, defined below:

Non-domination rank:

The non-dominated sorting procedure suggested by Deb et al. [2] is employed to calculate the non-domination rank of individuals in the population. According to this approach, each individual in the population is compared with every other individual to find if it is dominated. This process yields the solutions belonging to the first non-dominated front. The solutions of the first front are temporarily discounted and the above procedure is repeated to find the solutions of the next front. The process is repeated until all non-dominated fronts are identified. A non-domination rank j is assigned to each element of front F_j. Hence, solutions belonging to the same non-dominated front are assigned the same rank.

Diversity rank:

The crowding density of each solution is determined by using the crowding distance measure employed in NSGA-II. To estimate the density of solutions surrounding a particular solution x_i in a front, we compute the average distance of two solutions on either side of solution x_i along each of the objectives. A very large value of crowding distance is assigned to the boundary solutions in a front. For all other solutions within a front, the following procedure is used to assign the crowding distance [6].

For each solution x_i of front F, initialize crowding distance $d(x_i)$ to be 0;
For each objective function f_m do:
  Sort the solutions in F along objective f_m;
  $d(x_i) = d(x_i) + f_m$(*the individual that precedes* x_i *in the sorted sequence)*
   - f_m(*the individual that follows* x_i *in the sorted sequence)*

A solution with a higher crowding distance indicates better diversity. The solutions in the population are sorted and ranked based on the crowding distance. The solution with the highest crowding distance is assigned the best (lowest) diversity rank.

3.2 New Pool Generation

A "New Pool" is generated consisting of all the parents and some of the offspring, following a comparison of each offspring with its parents, considering both domination and crowding density. There are three possible cases:

Case 1: If the offspring dominates the parent, the offspring is added to the new pool.

Case 2: If the parent dominates the offspring, the crowding distance measure is used to calculate the probability of acceptance of the offspring. If the offspring has a higher crowding distance than the parent, it is added to the new pool with a probability P given by

$$P = 1 - \exp(\psi\ (\text{parent}) - \psi\ (\text{offspring}))$$

where ψ denotes the crowding distance of a solution. An offspring with a higher crowding distance (better diversity) than the parent has a high probability of acceptance. This strategy rewards solutions with higher diversity by allowing them to survive in future generations.

Case 3: If the parent and offspring are mutually non-dominating, then the offspring is added to the new pool if it has a higher crowding distance than the parent.

3.3. Trimming New Pool

The new pool is sorted based on the primary criterion of non-domination rank and the secondary criterion of diversity rank. In other words, solutions with the same non-domination rank are compared based on diversity rank. The new population will consist of the first μ elements of the sorted list containing solutions grouped into different fronts: $F_1, F_2, \ldots F_n$ where elements of F_{i+1} are dominated only by elements in $F_1, F2, \ldots F_i$. EMOCA maintains an archive of non-dominated solutions at every generation.

The above procedure indicates that EMOCA is a simple algorithm which maintains a balance between convergence and diversity. Convergence is emphasized by the concept of non-domination rank. Diversity is maintained in the population by using diversity rank in the tournament selection and population reduction phase. The crowding distance can also be implemented in the parameter space [12]. However, in our approach we calculate the crowding distance in the objective space. The computational complexity of NSGA-II and PAES are $O(MN^2)$ [2] and $O(MN^2d)$ [1] per iteration respectively where M is the number of objectives, N is the population size and d is the depth parameter. SPEA-II has a computational complexity of $O(\overline{N}^2 \log \overline{N})$ [4] per iteration where $\overline{N}$ is the sum of population size and archive size. The computational complexity of EMOCA is similar to NSGA-II of the order of $O(MN^2)$ per iteration.

4. Test Problems

We choose four widely used test problems called FON, POL, KUR and SCH from [15] and five test problems called ZDT1, ZDT2, ZDT3, ZDT4, ZDT5 and ZDT6 from [11]. These bi-objective problems are summarized in Table 1.

Table 1:Summary of test problems and the associated Pareto optimal fronts

Problem	n, range	Objective functions and their parameters	Characteristics of the Pareto-optimal front[2]
KUR	3, [-5,5]	$f_1(x)=\sum_{i=1}^{n-1} -10\exp(-0.2\sqrt{x_i^2+x_{i+1}^2})$, $f_2(x)=\sum_{i=1}^{n} \|x_i\|^8+5\sin x_i^3)$	$\bar{1}$, $\bar{2}$
FON	3, [-4,4]	$f_i(x)=1-\exp(-\sum_{j=1}^{3}(x_j+(-1)^i\times 1/\sqrt{3})^2)$, i=1,2	$\bar{1}$, 2
SCH	1,[10^3,10^3]	$f_1(x)=x^2$, $f_2(x)=(x-2)^2$	1,2
POL	2,[-π,π]	$f_1(x)=[1+(g_1-h_1)^2+(g_2-h_2)^2]$ $f_2(x)=[(x_1+3)^2+(x_2+1)^2]$ $g_1=0.5\sin 1-2\cos 1+\sin 2-1.5\cos 2$ $g_2=1.5\sin 1-\cos 1+2\sin 2-0.5\cos 2$ $h_1=0.5\sin x_1-2\cos x_1+\sin x_2-1.5\cos x_2$ $h_2=1.5\sin x_1-\cos x_1+2\sin x_2-0.5\cos x_2$	$\bar{1}$, $\bar{2}$
ZDT1	30, [0,1]	$f_1(x)=x_1$, $f_2(x)=g(x)[1-\sqrt{x_1/g(x)}\ ])$	1,2
ZDT2	30, [0,1]	$f_1(x)=x_1$, $f_2(x)=g(x)[1-(x_1/g(x))^2]$	$\bar{1}$, 2
ZDT3	30, [0,1]	$f_1(x)=x_1$, $f_2(x)=g(x)[1-\sqrt{x\ /g(x)}-(x_1/g(x))\times\sin(10\pi x_1)$ where $g(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$ for ZDT1,ZDT2 and ZDT3.	1, $\bar{2}$
ZDT4	$x_1\in[0,1]$ $x_i\in[-5,5]$	$f_1(x)=x_1$, $f_2(x)=g(x)[1-\sqrt{x_1/g(x)}]$ $g(x)=1+10(n-1)+\sum_{i=2}^{n}[x_i^2\ \ -10\cos(4\pi x_i)]$	$\bar{1}$, 2,3
ZDT6	10,[0,1]	$f_1(x)=1-\exp(-4x_1)\sin^6(4\pi x_1)$ $f_2(x)=g(x)[1-(f_1(x)/g(x))^2]$ $g(x)=1+9[(\sum_{i=2}^{n}x_i)/(n-1)]^{0.25}$	$\bar{1}$, 2

2 1: convex , $\bar{1}$: non-convex , 2:connected, $\bar{2}$: disconnected, 3: numerous local Pareto-optimal solutions

The test problems chosen by us span a wide variety of characteristics of the Pareto-optimal front such as non-convexity and disconnectivity. We have used a real coded representation for individuals. In the implementation of NSGA-II [2], simulated binary crossover (*SBX*) and parameter based mutation have been used. The studies in [13] showed that *SBX* outperforms binary coded genetic algorithms for continuous search space problems. Since the test problems in this study have continuous search space, we have employed *SBX* and parameter based mutation for EMOCA, NSGA-II, SPEA-II and PAES.

5. Simulation Results

In this section, we present the simulation results of EMOCA, NSGA-II, SPEA-II and PAES for the different test problems. For fairness of comparison with the work in [2], we have used a population size of 100. The algorithms were simulated for 250 generations over 30 trials. No further improvements in the performance of the algorithms were observed after 250 generations. The crossover probability was 0.9 and mutation probability was chosen to be $1/n$ where n is the number of decision variables. The algorithms exhibited similar performances for small variations in population size, crossover and mutation probabilities. We have used distribution indices of 20 for the simulated binary crossover and mutation [2]. For PAES, we have used a depth value of 4. We have used an archive size of 100 for all approaches. We performed experiments on an Intel Pentium 4 processor (3.2 GHz, 2 GB RAM). All approaches required an average computational time of 1 second per trial.

Performance comparison of *MOEA*s is a very difficult task as discussed in [14]. Several performance metrics have been proposed. According to the study in [14], no single satisfactory metric exists. While evaluating the performance of *MOEA*s, we should take into account both convergence to the Pareto optimal front and diversity. However, for benchmark test function study, certain metrics have been found to be appropriate for performance comparison [2]. As suggested in [2], we use the generational distance and the spread metric to evaluate the performance of different algorithms.

5.1. Generational Distance

Veldhuizen et al. [15] proposed the generational distance (*GD*) metric which evaluates the convergence of the non-dominated solutions obtained by the algorithm to the true Pareto-optimal front. A lower value of the metric indicates a better convergence. The *GD* metric is defined as $GD = (\sum_{i=1}^{|Q|} d_i^{\,2})^{0.5}$ where Q indicates the set of non-dominated solutions obtained by the algorithm and d_i is the Euclidean distance between the solution $i \in Q$ and the nearest solution of the Pareto-optimal front P^*. The generational distance measures the extent of convergence to the Pareto-optimal front. Since the Pareto-optimal solutions for the test functions are known, we find a set of 500 uniformly spaced solutions [2] from the true Pareto-optimal front in the objective space. For each solution obtained with the algorithm, we compute the minimum Euclidean distance of it from the solutions of the Pareto-optimal front. The average of these distances gives the generational distance. A lower value of the generational distance indicates a better performance.

Table 2 shows the mean and variance of the generational distance values obtained over 30 trials. The results clearly indicate that EMOCA outperforms NSGA-II in all test problems except in FON. EMOCA outperforms SPEA-II and PAES in all test problems except ZDT3. The performance of EMOCA is identical to SPEA-II and PAES in ZDT3 as observed from Table 1. These results confirm that EMOCA performs the best in a majority of the test problems. In particular, for problems ZDT4 and ZDT6, EMOCA obtains much lower generational distance values as compared to the other algorithms.

Table 2: Mean and variance of generational distance over 30 trials

Problem	EMOCA		NSGA-II		SPEA-II		PAES	
	mean	variance	mean	variance	mean	variance	mean	variance
KUR	**0.018**	0.0006	0.029	0.00034	0.097	0.033	1.669	0.0005
FON	0.0061	0.00007	**0.0021**	0	0.044	0.0001	0.030	0
SCH	**0.0019**	0.00016	0.0034	0.0002	0.003	0.0003	0.111	0.00007
POL	**0.013**	0.00029	0.015	0.00012	1.11	0.0081	0.92	0.017
ZDT1	**0.029**	0.0082	0.034	0.0055	0.0432	0	0.032	0.00001
ZDT2	**0.016**	0	0.075	0.00001	0.033	0.00001	0.029	0.00001
ZDT3	**0.039**	0.0004	0.11	0.0006	**0.04**	0	**0.04**	0
ZDT4	**0.022**	0.0001	0.523	0.001	0.15	0.0015	0.338	0.00001
ZDT6	**0.024**	0	0.3	0.0002	0.12	0.0007	0.335	0.00006

5.2. Set Coverage Metric

Zitzler et al. [16] have proposed the set coverage metric (C-metric). The C metric calculates the fraction of solutions in one non-dominated set (obtained by one algorithm) that are dominated by those obtained by the other algorithm. The C metric is defined as

$$C(A,B)= | \{b \in B | \exists a \in A: a \gg b\}| / |B|.$$

C(A,B)=1 indicates that every solution in B is dominated by solutions in A and C(A,B)=0 means that none of the solutions in B is dominated by any element in A. Since the C metric is asymmetric, it is necessary to examine both C(A,B) and C(B,A) . It can be argued that evolutionary algorithm A is better than evolutionary algorithm B, if over many trials we repeatedly and consistently observe that C(A,B) is significantly higher than C(B,A). Table 3 shows the mean and variance of the C metric values for different algorithms over 30 trials. The results show that *C(other algorithm, EMOCA)* is consistently low, with relatively high values for *C(EMOCA, other algorithm),* confirming the superior performance of EMOCA.

Table 3: Mean and variance of C metric values comparing EMOCA with various algorithms: all results are averages over 30 trials

Problem	C(NSGA-II,EMOCA) C(EMOCA,NSGA-II)		C(SPEA-II, EMOCA) C(EMOCA, SPEA-II)		C(PAES,EMOCA) C(EMOCA,PAES)	
	mean	variance	mean	variance	mean	variance
KUR	**0.12** 0.40	0.0017 0.0004	**0.09** 0.72	0.0001 0.016	**0.13** 0.69	0.18 0.0012
FON	0.63 **0.104**	0.0012 0.0003	**0.01** 0.68	0.0002 0.008	**0.12** 0.89	0.001 0.001
SCH	**0.01** 0.521	0.0052 0.003	**0.02** 0.9	0 0.02	**0.017** 0.83	0.005 0.00003
POL	**0.07** 0.627	0.0017 0.0002	**0.06** 0.6	0.02 0.004	**0.08** 0.85	0.014 0
ZDT1	**0.03** 0.619	0.0091 0.0006	**0.00** 0.64	0 0.01	**0.058** 0.72	0.0031 0.0017
ZDT2	**0.02** 0.41	0.0005 0.001	**0.001** 0.6	0 0.01	**0.05** 0.75	0.003 0.001
ZDT3	**0.05** 0.85	0.0026 0.0006	**0.003** 0.12	0 0.01	**0.053** 0.16	0.01 0.003
ZDT4	**0.04** 0.91	0.032 0.001	**0.003** 0.66	0 0.02	**0.043** 0.79	0.003 0.014
ZDT6	**0.01** 0.93	0.0073 0.0211	**0.01** 0.65	0.009 0.12	**0.1** 0.83	0.005 0.0003

5.3. Domination Metric

We have proposed a symmetric metric called the domination metric (*Dom metric*) to evaluate the relative performance of two *MOEA*s [8]. This metric is based on the number of solutions (obtained by one algorithm) dominated by each solution obtained by the other algorithm. The *Dom metric* is defined as:

$$Dom(A,B) = d(A,B)/(d(A,B)+d(B,A)), \text{ where } d(X,Y)= \sum_{x} |\{y \in Y \mid x \gg y\}|.$$

Mutually non-dominating solutions are ignored while evaluating the dominance factor *d(A,B)* so that *Dom (B,A)=1-Dom (A,B)*. Consequently, for two algorithms A and B, it is sufficient to measure the performance in terms of *Dom(A,B).* If each solution of algorithm A dominates every solution of algorithm B, then *Dom (A,B)=1* and *Dom(B,A)=0*. Table 4 shows *Dom*-metric values comparing EMOCA with other algorithms.

Table 4: Mean and variance of *Dom* metric values comparing EMOCA with NSGA-II, PAES and SPEA-II : all results are averages over 30 trials.

Problem	*Dom* (EMOCA,NSGA-II)		*Dom*(EMOCA,SPEA-II)		*Dom* (EMOCA,PAES)	
	mean	variance	mean	variance	mean	variance
KUR	**0.59**	0.59	**0.97**	0.008	**0.99**	0.0003
FON	0.40	0.021	**0.98**	0.0001	**0.73**	0.046
SCH	**0.63**	0.63	**1**	0	**0.75**	0.026
POL	**0.65**	0.061	**0.91**	0.009	**0.73**	0.037
ZDT1	**0.78**	0.035	**1**	0	**0.82**	0.03
ZDT2	**0.8**	0.037	**0.99**	0	**0.76**	0.041
ZDT3	**0.83**	0.009	0.42	0.0003	0.46	0.054
ZDT4	**0.80**	0.082	**0.99**	0.002	**0.70**	0.008
ZDT6	**0.96**	0.0332	**0.99**	0.0001	**0.85**	0.049

Consistently, we find that *Dom(EMOCA, other algorithm)* > 0.5, except for problem FON for which NSGA-II has better performance according to this metric, and ZDT3 for which SPEA-II and PAES have better performance. In other words, the solutions obtained by EMOCA tend to dominate those obtained by the other algorithms, more than the latter dominate EMOCA.

5.4. Spread Metric

To evaluate the diversity of the solutions obtained, Deb et al. [2] suggested the spread metric. The spread metric is defined as:

$$\Delta = \frac{\sum_{m=1}^{M} d_m^{\,e} + \sum_{i=}^{|Q|} | d_i - \bar{d} |}{\sum_{m=1}^{M} d_m^{\,e} + |Q| \bar{d}}$$

where d_i is the Euclidean distance between neighboring solutions and $\bar{d}$ is the mean value of the distances. The parameter $d_m^{\,e}$ is the distance between extreme solutions of the Pareto-optimal front and the nearest solution of Q corresponding to the m^{th} objective function. For an ideal distribution of solutions, $\Delta = 0$. A value of Δ close to zero indicates that the non-dominated solutions obtained by the algorithm are uniformly spaced.

Table 5: Mean and variance of the spread metric: all results are averages over 30 trials

Problem	EMOCA		NSGA-II		SPEA-II		PAES	
	mean	variance	mean	variance	mean	variance	mean	variance
KUR	**0.1032**	0.00087	0.4347	0.00092	0.3034	0.041	0.371	0.001
FON	**0.1593**	0.0041	0.3978	0.0073	0.6634	0.0207	0.562	0.054
SCH	**0.2321**	0.00309	0.4891	0.0047	0.981	0.002	0.286	0.012
POL	**0.1077**	0.0029	0.4561	0.00294	0.2359	0.059	0.251	0.032
ZDT1	**0.4024**	0.0047	0.4109	0.0018	0.6404	0.014	0.531	0.038
ZDT2	**0.2482**	0.0023	0.4476	0.0052	0.6437	0.019	0.581	0.066
ZDT3	**0.4853**	0.00169	0.6898	0.0096	0.7282	0.034	0.634	0.037
ZDT4	0.3072	0.0086	0.7451	0.00178	**0.195**	0.113	0.412	0.07
ZDT6	**0.5399**	0.0016	0.6981	0.0029	0.6543	0.056	0.831	0.054

Table 5 shows the mean and variance of the diversity metric (spread) obtained over 30 trials. The results indicate that EMOCA outperforms NSGA-II and PAES in all test problems in terms of the spread. In problem ZDT4, SPEA-II performs the best in terms of spread. In all other test problems EMOCA performs better than SPEA-II. These results show that EMOCA obtains a more uniformly spaced and a better spread of solutions compared to the other algorithms. EMOCA incorporates diversity preservation as an integral part of the algorithm. The Δ values obtained by EMOCA are much lower than those obtained by the other algorithms. This shows that EMOCA discovers a diverse set of non-dominated solutions with near- uniform spacing.

5.5. Plots of Non-Dominated Solutions Obtained

The averages and variances of the performance metrics give a clear indication of the performance of different algorithms. To appreciate the nature of solutions obtained by EMOCA, we plot the Pareto surfaces obtained by different algorithms for a few selective examples. Figure 2 shows the Pareto plots obtained by EMOCA, NSGA-II, SPEA-II and PAES on ZDT6. ZDT6 has a non-convex Pareto-optimal front with non-uniform spacing between the solutions. The plot shows that EMOCA performs the best for ZDT6. EMOCA obtains a better spread and convergence compared to the other algorithms. SPEA-II performs slightly worse than EMOCA while PAES has the worst performance on ZDT6. We can also observe that the Pareto front obtained by EMOCA covers a wider region of the objective space compared to the Pareto fronts obtained by the other algorithms.

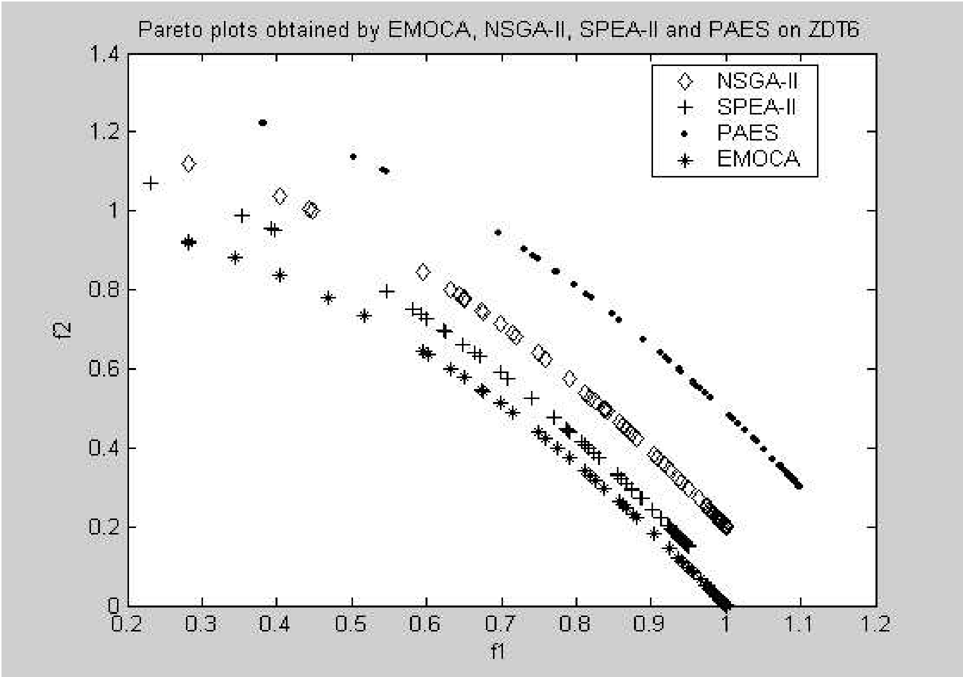

Figure 2: Performance comparison of EMOCA, NSGA-II, SPEA-II and PAES on ZDT6.

The remaining figures show pairwise comparisons of EMOCA with other algorithms to reduce clutter. Figure 3 shows the non-dominated solutions obtained by EMOCA and NSGA-II on KUR. The problem KUR has discontinuous regions in the Pareto-optimal front. The plots show that EMOCA is able to discover several diverse non-dominated solutions compared to NSGA-II.

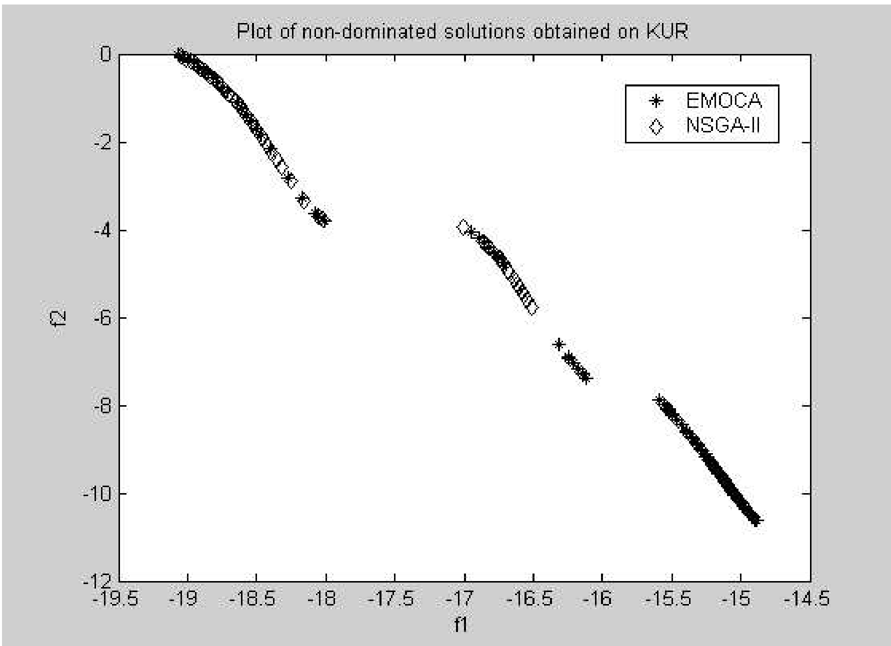

Figure 3: Performance comparison of EMOCA and NSGA-II on KUR

The problem ZDT2 has a non-convex Pareto-optimal front. Figure 4 shows the non-dominated solutions obtained by EMOCA and NSGA-II on ZDT2. The plot indicates that EMOCA obtains uniformly spaced diverse set of non-dominated solutions compared to NSGA-II. This observation is confirmed by a very low value of the spread metric for EMOCA as shown in Table 4.

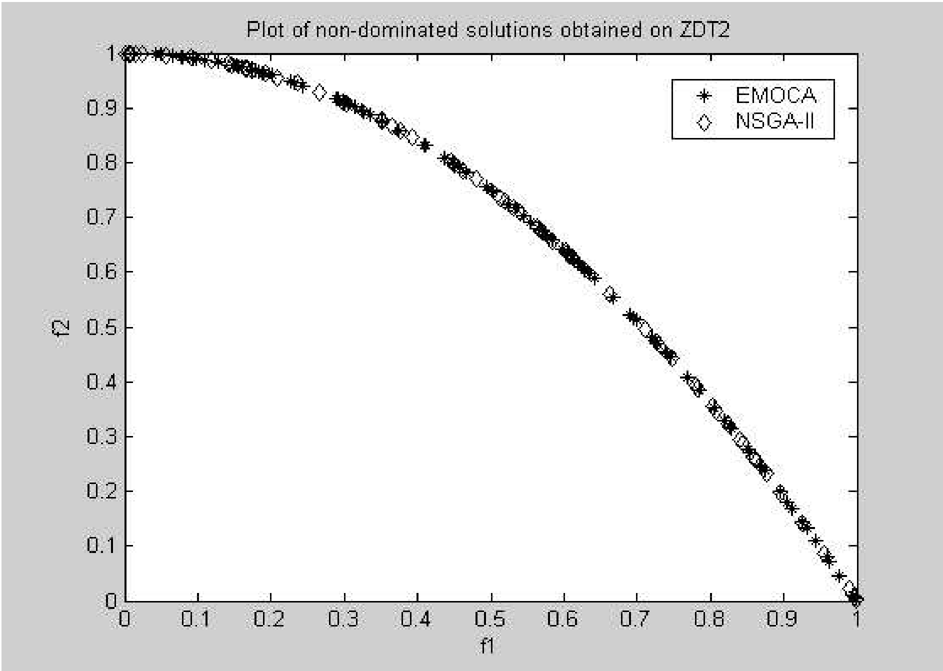

Figure 4: Non-dominated solutions obtained by EMOCA on ZDT2

Figure 5 shows the performance comparison of EMOCA and NSGA-II on ZDT4. The problem ZDT4 has 21^9 different local Pareto-optimal fronts [6]. This is a challenging problem for *MOEA*s. The plots clearly show that EMOCA has a better convergence and diversity compared to NSGA-II.

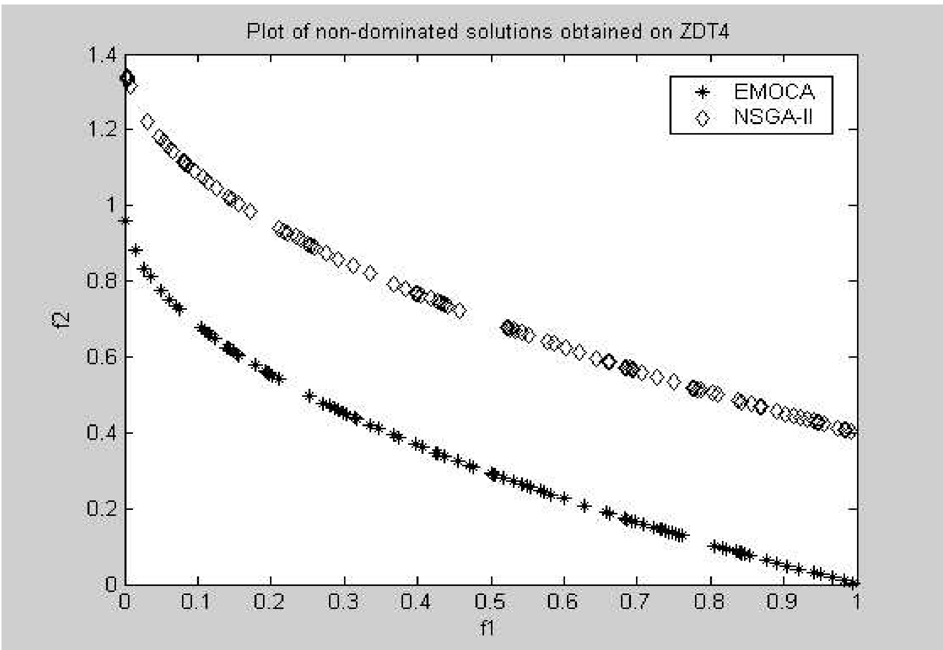

Figure 5: Performance comparison of EMOCA and NSGA-II on ZDT4

Figure 6 shows the Pareto plots obtained by EMOCA and SPEA-II on POL. The function POL has a non-convex and disconnected Pareto-optimal front. From the plot, we observe that EMOCA obtains solutions with better convergence and spacing compared to the solutions obtained by SPEA-II.

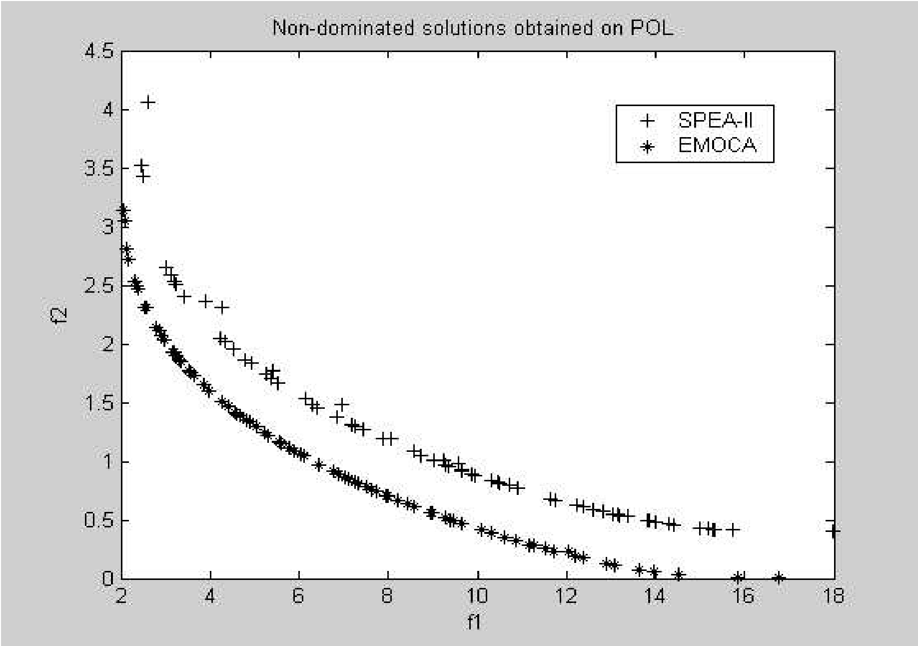

Figure 6: Performance comparison of EMOCA and SPEA-II on POL

Figure 7 shows the Pareto plots obtained by EMOCA and PAES on SCH. The plots show that EMOCA obtains a better convergence towards the Pareto-optimal front than PAES.

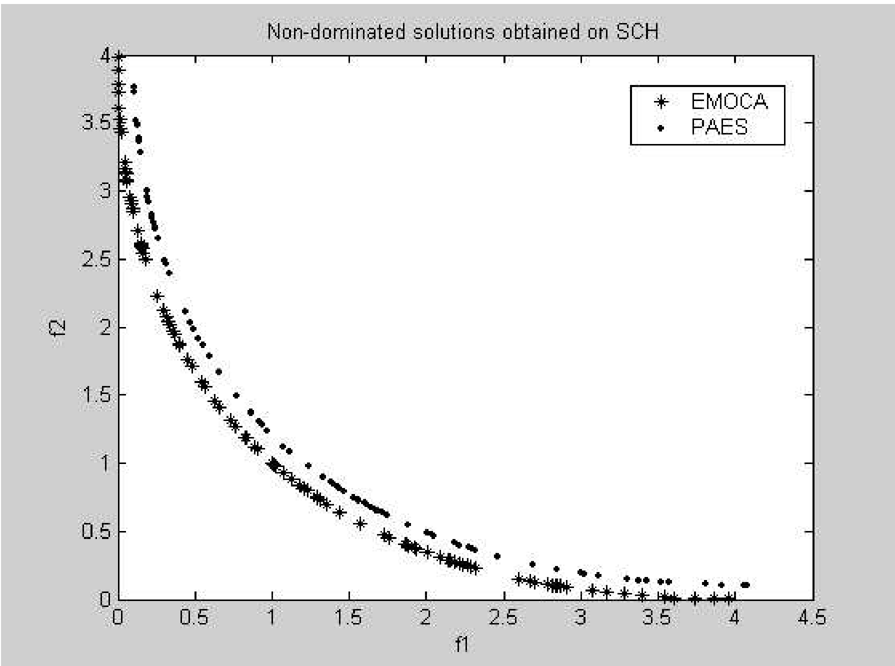

Figure 7: Performance comparison of EMOCA and PAES on SCH

7. Conclusions

We have proposed a new *MOEA* called EMOCA which employs a stochastic replacement selection strategy that considers both non-domination and diversity. EMOCA was successful in several real world applications such as path planning, sensor placement and mobile agent routing in wireless sensor networks [7, 8, 9, 10]. In this paper, we have compared the performance of EMOCA with NSGA-II, SPEA-II and PAES on nine difficult test problems with distinct features. Several performance measures were used to compare the algorithms. The simulation results show that EMOCA outperforms the other algorithms in eight out of the nine test problems in terms of convergence and diversity, consistently discovering a widely spread set of non-dominated solutions. The successful performance of EMOCA in these test problems shows that EMOCA is an efficient multi-objective optimization algorithm which should find extensive applications in optimization problems spanning a wide variety of areas from path planning to wireless networks.

References:

1. J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for multi-objective optimization," In Proceedings of the 1999 congress on Evolutionary Computation, pp.98-105, 1999.

2. K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," Proc. Parallel Problem Solving from Nature VI, pp.849-858, 2000.

3. E. Zitzler, and L. Thiele, "Multiobjective optimization using evolutionary algorithms – A comparative study," Proc. Parallel Problem Solving from Nature V, pp. 292-301, 1998.

4. E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Swiss Federal Institute of Technology, Tech. Rep. TIK-Rep. 103, 2001.

5. K. Deb, "Multi-objective optimization using evolutionary algorithms," Wiley and Sons, England, 2001.

6. C.A. Coello Coello and G. B. Lamont. *Application of multi-objective evolutionary algorithms*, World Scientific, Singapore, 2004.

7. R. Rajagopalan, "Path planning with evolutionary algorithms," M.S. Thesis, Department of Electrical Engineering & Computer Science, Syracuse University, July 2004.

8. R. Rajagopalan, C.K. Mohan, K.G. Mehrotra and P.K. Varshney, "Evolutionary Multi-objective crowding algorithm for path computations," Proc. Fifth International Conference on Knowledge based Computer Systems, pp. 46-65, Dec. 2004.

9. R. Rajagopalan, C.K. Mohan, P.K. Varshney and K.G. Mehrotra, "Sensor placement for energy efficient target detection in wireless sensor networks: A multi-objective optimization approach," 39 th Annual Conference on Information Sciences and Systems, Baltimore, MaryLand, March 2005.

10. R. Rajagoapalan, C.K. Mohan, P.K. Varshney and K.G. Mehrotra, "Multi-objective mobile agent routing in wireless sensor networks," IEEE Congress on Evolutionary Computation, September 2005 (accepted).

11. E. Zitzler, K. Deb, and L. Thiele, "Comparison of multi-objective evolutionary algorithm: Empirical results," Evolutionary Computation, vol. 8, pp. 173-195, 2000.

12. K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test functions," Evolutionary Computation, Vol.7, pp. 205-230, 1999.

13. K .Deb, and R.B. Agarwal, "Simulated binary crossover for continuous search space," Complex Systems, Vol. 9, pp.115-148, 1995.

14. J.D. Knowles, and D.W. Corne, "On metrics for comparing non-dominated sets," Proc. IEEE Congress on Evolutionary Computation, pp. 711-716, 2002.

15. D.V. Veldhuizen, "Multi-objective evolutionary algorithms: Classifications, analyses and new innovations," Ph.D. Thesis, Air Force Institute of Technology, Ohio, Technical Report AFIT/DS/ENG/99-01, 1999.

16. E. Zitzler, "Evolutionary algorithms for multi-objective optimization: Methods and applications," Ph.D. Thesis, Swiss Federal Institute of Technology (ETH) (Dissertation ETH No. 13398).