A Simulated Annealing Based Multi-objective Optimization Algorithm: AMOSA

Sanghamitra Bandyopadhyay[1], Sriparna Saha[1], Ujjwal Maulik[2] and Kalyanmoy Deb[3]
[1]Machine Intelligence Unit, Indian Statistical Institute, Kolkata-700108, India.
Email: {sanghami,sriparna_r}@isical.ac.in
[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India
Email: drumaulik@cse.jdvu.ac.in
[3] KanGAL, Department of Mechanical Engineering,
Indian Institute of Technology Kanpur, Kanpur - 208016, India. Email: deb@iitk.ac.in

***Abstract*— This article describes a simulated annealing based multi-objective optimization algorithm that incorporates the concept of archive in order to provide a set of trade-off solutions of the problem under consideration. To determine the acceptance probability of a new solution *vis-a-vis* the current solution, an elaborate procedure is followed that takes into account the domination status of the new solution with the current solution, as well as those in the archive. A measure of the amount of domination between two solutions is also used for this purpose. A complexity analysis of the proposed algorithm is provided. An extensive comparative study of the proposed algorithm with two other existing and well-known multi-objective evolutionary algorithms (MOEAs) demonstrate the effectiveness of the former with respect to five existing performance measures, and several test problems of varying degrees of difficulties. In particular, the proposed algorithm is found to be significantly superior for many-objective test problems (e.g., 4, 5, 10 and 15 objective problems), while recent studies have indicated that the Pareto ranking-based MOEAs perform poorly for such problems. In a part of the investigation, comparison of the real-coded version of the proposed algorithm is conducted with a very recent multi-objective simulated annealing algorithm where the performance of the former is found to be generally superior to that of the latter.**

***Index Terms*— Amount of domination, archive, clustering, multi-objective optimization, Pareto-optimal, simulated annealing.**

I. INTRODUCTION

The multi-objective optimization (MOO) problem has a rather different perspective compared to one having a single objective. In single-objective optimization there is only one global optimum, but in multi-objective optimization there is a set of solutions, called the Pareto-optimal (PO) set, which are considered to be equally important; all of them constitute global optimum solutions. Over the decade, a number of Multi-objective Evolutionary Algorithms (MOEAs) have been suggested (see, [1], [2] for some reviews). The main reason for the popularity of Evolutionary algorithms (EAs) for solving multi-objective optimization is their population based nature and ability of finding multiple optima simultaneously.

Simulated Annealing (SA) [3] another popular search algorithm, utilizes the principles of statistical mechanics regarding the behavior of a large number of atoms at low temperature, for finding minimal cost solutions to large optimization problems by minimizing the associated energy. In statistical mechanics investigating the ground states or low energy states of matter is of fundamental importance. These states are achieved at very low temperatures. However, it is not sufficient to lower the temperature alone since this results in unstable states. In the annealing process, the temperature is first raised, then decreased gradually to a very low value ($Tmin$), while ensuring that one spends sufficient time at each temperature value. This process yields stable low energy states. Geman and Geman [4] provided a proof that SA, if annealed sufficiently slowly, converges to the global optimum. Being based on strong theory SA has been applied in diverse areas [5], [6], [7] by optimizing a single criterion. However there have been only a few attempts in extending SA to multi-objective optimization, primarily because of its search-from-a-point nature. In most of the earlier attempts, a single objective function is constructed by combining the different objectives into one using a weighted sum approach [8]-[13]. The problem here is how to choose the weights in advance. Some alternative approaches have also been used in this regard. In [11] and [12] different non-linear and stochastic composite energy functions have been investigated. In [11] six different criteria for energy difference calculation are suggested and evaluated. These are (1) minimum cost criterion, (2) maximum cost criteria, (3) random cost criteria, (4) self cost criteria, (5) average cost criteria, and (6) fixed cost criteria. Since each run of the SA provides just a single solution, the algorithm attempted to evolve the set of PO solutions by using multiple SA runs. As a result of the independent runs, the diversity of the set of solutions suffered.

Multi-objective simulated annealing with a composite energy clearly converges to the true Pareto front if the objectives have ratios given by w_i^{-1}, if such points, in general, exist. Here w_i is the weight assigned to the ith objective. In [14], it has been proved that part of the front will be inaccessible with fixed weights. In [15] several different schemes were explored for adapting the w_is during the annealing process to encourage exploration along the front. However, a proper choice of the w_is remains challenging task.

In addition to the earlier aggregating approaches of multi-objective SA, there have been a few techniques that incorporate the concept of Pareto-dominance. Some such methods are proposed in [16], [17] which use Pareto-domination based acceptance criterion in multi-objective SA. A good review of several multi-objective simulated annealing algorithms and their comparative performance analysis can be found in [18]. Since the technique in [17] has been used in this article for the purpose of comparison, it is described in detail later.

In Pareto-domination-based multi-objective SAs developed so far, the acceptance criterion between the current and a new solution has been formulated in terms of the difference in the number of solutions that they dominate [16], [17]. The amount by which this domination takes place is not taken into consideration. In this article, a new multi-objective SA is proposed, hereafter referred to as AMOSA (Archived Multi-objective Simulated Annealing), which incorporates a novel concept of amount of dominance in order to determine the acceptance of a new solution. The PO solutions are stored in an archive. A complexity analysis of the proposed AMOSA is provided. The performance of the newly proposed AMOSA is compared to two other well-known MOEA's, namely NSGA-II [19] and PAES [20] for several function optimization problems when binary encoding is used. The comparison is made in terms of several performance measures, namely *Convergence* [19], *Purity* [21], [22], *Spacing* [23] and *MinimalSpacing* [21]. Another measure called *displacement* [8], [24], that reflects both the proximity to and the coverage of the true PO front, is also used here for the purpose of comparison. This measure is especially useful for discontinuous fronts where we can estimate if the solution set is able to approximate all the sub-fronts. Many existing measures are unable to achieve this.

It may be noted that the multi-objective SA methods developed in [16], [17] are on lines similar to ours. The concept of archive or a set of potentially PO solutions is also utilized in [16], [17] for storing the non-dominated solutions. Instead of scalarizing the multiple objectives, a domination based energy function is defined. However there are notable differences. Firstly, while the number of solutions that dominate the new solution *x* determines the acceptance probability of *x* in the earlier attempts, in the present article this is based on the amount of domination of *x* with respect to the solutions in the archive and the current solution. In contrast to the works in [16], [17] where a single form of acceptance probability is considered, the present article deals with different forms of acceptance probabilities depending on the domination status, the choice of which are explained intuitively later on.

In [17] an unconstrained archive is maintained. Note that theoretically, the number of Pareto-optimal solutions can be infinite. Since the ultimate purpose of an MOO algorithm is to provide the user with a set of solutions to choose from, it is necessary to limit the size of this set for it to be usable by the user. Though maintaining unconstrained archives as in [17] is novel and interesting, it is still necessary to finally reduce it to a manageable set. Limiting the size of the population (as in NSGA-II) or the Archive (as in AMOSA) is an approach in this direction. Clustering appears to be a natural choice for reducing the loss of diversity, and this is incorporated in the proposed AMOSA. Clustering has also been used earlier in [25].

For comparing the performance of real-coded AMOSA with that of the multi-objective SA (MOSA) [17], six three objective test problems, namely, DTLZ1-DTLZ6 are used. Results demonstrate that the performance of AMOSA is comparable to, often better than, that of MOSA in terms of *Purity*, *Convergence* and *Minimal Spacing*. Comparison is also made with real-coded NSGA-II for the above mentioned six problems, as well as for some 4, 5, 10 and 15 objective test problems. Results show that the performance of AMOSA is superior to that of NSGA-II specially for the test problems with many objective functions. This is an interesting and the most desirable feature of AMOSA since Pareto ranking-based MOEAs, such as NSGA-II [19] do not work well on many-objective optimization problems as pointed out in some recent studies [26], [27].

II. Multi-objective Algorithms

The multi-objective optimization can be formally stated as follows [1]. Find the vectors $\overline{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ of decision variables that simultaneously optimize the *M* objective values $\{f_1(\overline{x}), f_2(\overline{x}), \ldots, f_M(\overline{x})\}$, while satisfying the constraints, if any.

An important concept of multi-objective optimization is that of domination. In the context of a maximization problem, a solution $\overline{x_i}$ is said to dominate $\overline{x_j}$ if $\forall k \in 1, 2, \ldots, M,\ f_k(\overline{x_i}) \geq f_k(\overline{x_j})$ and $\exists k \in 1, 2, \ldots, M,$ such that $f_k(\overline{x_i}) > f_k(\overline{x_j})$. Among a set of solutions P, the nondominated set of solutions $P^{'}$ are those that are not dominated by any member of the set P. The non-dominated set of the entire search space *S* is the globally Pareto-optimal set. In general, a multi-objective optimization algorithm usually admits a set of solutions that are not dominated by any solution encountered by it.

A. Recent MOEA algorithms

During 1993-2003, a number of different evolutionary algorithms were suggested to solve multi-objective optimization problems. Among these, two well-known ones namely, PAES [20] and NSGA-II [19], are used in this article for the purpose of comparison. These are described in brief.

Knowles and Corne [20] suggested a simple MOEA using a single parent, single child evolutionary algorithm, similar to (1+1) evolutionary strategy. Instead of using real parameters, binary strings and bit-wise mutation are used in this algorithm to create the offspring. After creating the child and evaluating its objectives, it is compared with respect to the parent. If the child dominates the parent, then the child is accepted as the next parent and the iteration continues. On the other hand if parent dominates the child, the child is discarded and a new mutated solution (a new solution) is generated from the parent. However if the parent and the child are nondominating to each other, then the choice between the child and the parent is resolved by comparing them with an archive of best solutions found so far. The child is compared with all members of the archive to check if it dominates any member of the archive. If yes, the child is accepted as the new parent and all the dominated solutions are eliminated from the archive. If the child does not dominate any member of the archive, both the parent and the child are checked for their *nearness* with the solutions of the archive. If the child resides in a less crowded region in the parameter space, it is accepted as a parent and a copy is added to the archive. Generally this crowding concept is implemented by dividing the whole solution space into d^M subspaces where d is the depth parameter and M is the number of objective functions. The subspaces are updated dynamically.

The other popular algorithm for multi-objective optimization is NSGA-II proposed by Deb *et al.* [19]. Here, initially a random parent population P_0 of size N is created. Then the population is sorted based on the non-domination relation. Each solution of the population is assigned a fitness which is equal to its non-domination level. A child population Q_0 is created from the parent population P_0 by using binary tournament selection, recombination, and mutation operators. Generally according to this algorithm, initially a combined population $R_t = P_t + Q_t$ is formed of size R_t, which is $2N$. Now all the solutions of R_t are sorted based on their non-domination status. If the total number of solutions belonging to the best non-dominated set F_1 is smaller than N, F_1 is completely included into $P_{(t+1)}$. The remaining members of the population $P_{(t+1)}$ are chosen from subsequent non-dominated fronts in the order of their ranking. To choose exactly N solutions, the solutions of the last included front are sorted using the crowded comparison operator and the best among them (i.e., those with larger values of the crowding distance) are selected to fill in the available slots in $P_{(t+1)}$. The new population $P_{(t+1)}$ is now used for selection, crossover and mutation to create a new population $Q_{(t+1)}$ of size N, and the process continues. The crowding distance operator is also used in the parent selection phase in order to break a tie in the binary tournament selection. This operator is basically responsible for maintaining diversity in the Pareto front.

B. Recent MOSA algorithm [17]

One of the recently developed MOSA algorithm is by Smith et al. [17]. Here a dominance based energy function is used. If the true Pareto front is available then the energy of a particular solution x is calculated as the total number of solutions that dominates x. However as the true Pareto front is not available all the time a proposal has been made to estimate the energy based on the current estimate of the Pareto front, $F^{'}$, which is the set of mutually non-dominating solutions found thus far in the process. Then the energy of the current solution x is the total number of solutions in the estimated front which dominates x. If $\|F^{'}_{x^{'}}\|$ is the energy of the new solution $x^{'}$ and $\|F^{'}_{x}\|$ is the energy of the current solution x, then energy difference between the current and the proposed solution is calculated as $\delta E(x^{'},x) = (\|F^{'}_{x^{'}}\| - \|F^{'}_{x}\|)/\|F^{'}\|$. Division by $\|F^{'}\|$ ensures that δE is always less than unity and provides some robustness against fluctuations in the number of solutions in $F^{'}$. If the size of $F^{'}$ is less than some threshold, then attainment surface sampling method is adopted to increase the number of solutions in the final Pareto front. Authors have perturbed a decision variable with a random number generated from the laplacian distribution. Two different sets of scaling factors, traversal scaling which generates moves to a non-dominated proposal within a front, and location scaling which locates a front closer to the original front, are kept. These scaling factors are updated with the iterations.

III. Archived Multi-objective Simulated Annealing (AMOSA)

As mentioned earlier, the AMOSA algorithm is based on the principle of SA [3]. In this article at a given temperature T, a new state, s, is selected with a probability

$$p_{qs} = \frac{1}{1 + e^{\frac{-(E(q,T)-E(s,T))}{T}}}, \quad (1)$$

where q is the current state and $E(s,T)$ and $E(q,T)$ are the corresponding energy values of s and q, respectively. Note that the above equation automatically ensures that the probability value lies in between 0 and 1. AMOSA incorporates the concept of an *Archive* where the non-dominated solutions seen so far are stored. In [28] the use of unconstrained *Archive* size to reduce the loss of diversity is discussed in detail. In our approach we have kept the archive size limited since finally only a limited number of well distributed Pareto-optimal solutions are needed. Two limits are kept on the size of the *Archive*: a hard or strict limit denoted by *HL*, and a soft limit denoted by *SL*. During the process, the non-dominated solutions are stored in the *Archive* as and when they are generated until the size of the *Archive* increases to *SL*. Thereafter if more non-dominated solutions are generated, these are added to the *Archive*, the size of which is thereafter reduced to *HL* by applying clustering. The structure of the proposed simulated annealing based AMOSA is shown in Figure 1. The parameters that need to be set *a priori* are mentioned below.

- *HL*: The maximum size of the *Archive* on termination. This set is equal to the maximum number of non-dominated solutions required by the user.
- *SL*: The maximum size to which the *Archive* may be filled before clustering is used to reduce its size to *HL*.
- *Tmax*: Maximum (initial) temperature.
- *Tmin*: Minimal (final) temperature.
- *iter*: Number of iterations at each temperature.
- α: The cooling rate in SA

The different steps of the algorithm are now explained in detail.

A. Archive Initialization

The algorithm begins with the initialization of a number $\gamma\times$ *SL* ($\gamma > 1$) of solutions. Each of these solutions is refined by using a simple hill-climbing technique, accepting a new solution only if it dominates the previous one. This is continued for a number of iterations. Thereafter the non-dominated solutions (*ND*) that are obtained are stored in the *Archive*, up to a maximum of *HL*. In case the number of nondominated solutions exceeds *HL*, clustering is applied to

Algorithm AMOSA

Set *Tmax*, *Tmin*, *HL*, *SL*, *iter*, α, temp=*Tmax*.
Initialize the *Archive*.
current-pt = random(*Archive*). /* randomly chosen solution from the *Archive**/
while (temp $>$ *Tmin*)
  for (i=0; i< iter; i++)
    new-pt=perturb(*current-pt*).
    Check the domination status of *new-pt* and *current-pt*.
    /* Code for different cases */
    if (*current-pt* dominates *new-pt*) /* Case 1*/
      $\Delta dom_{avg} = \frac{\left(\sum_{i=1}^{k} \Delta dom_{i,new-pt}\right) + \Delta dom_{current-pt,new-pt}}{(k+1)}$.
      /* *k*=total-no-of points in the *Archive* which dominate *new-pt*, $k \geq 0$. */
      prob= $\frac{1}{1+exp(\Delta dom_{avg} * temp)}$.
      Set *new-pt* as *current-pt* with probability=*prob*
    if (*current-pt* and *new-pt* are non-dominating to each other) /* Case 2*/
      Check the domination status of *new-pt* and points in the *Archive*.
      if (*new-pt* is dominated by k ($k \geq$1) points in the *Archive*) /* Case 2(a)*/
        prob= $\frac{1}{1+exp(\Delta dom_{avg} * temp)}$.
        $\Delta dom_{avg} = \frac{\left(\sum_{i=1}^{k} \Delta dom_{i,new-pt}\right)}{k}$.

        Set *new-pt* as *current-pt* with probability=*prob*.
      if (*new-pt* is non-dominating w.r.t all the points in the *Archive*) /* Case 2(b)*/
        Set *new-pt* as *current-pt* and add *new-pt* to the *Archive*.
        if *Archive-size* $>$ *SL*
          Cluster *Archive* to *HL* number of clusters.
      if (*new-pt* dominates *k*, ($k \geq$1) points of the *Archive*) /* Case 2(c)*/
        Set *new-pt* as *current-pt* and add it to the *Archive*.
        Remove all the *k* dominated points from the *Archive*.
    if (*new-pt* dominates *current-pt*) /* Case 3 */
      Check the domination status of *new-pt* and points in the *Archive*.
      if (*new-pt* is dominated by k ($k \geq$1) points in the *Archive*) /* Case 3(a)*/
        Δdom_{min} = minimum of the difference of domination amounts between the *new-pt*
              and the k points
        prob= $\frac{1}{1+exp(-\Delta dom_{min})}$.
        Set point of the archive which corresponds to Δdom_{min} as *current-pt* with probability=*prob*
        else set *new-pt* as *current-pt*.
      if (*new-pt* is non-dominating with respect to the points in the *Archive*) /* Case 3(b) */
        Set *new-pt* as the *current-pt* and add it to the *Archive*.
        if *current-pt* is in the *Archive*, remove it from the *Archive*.
        else if *Archive-size*$>$ *SL*.
          Cluster *Archive* to *HL* number of clusters.
      if (*new-pt* dominates k other points in the *Archive*) /* Case 3(c)*/
        Set *new-pt* as *current-pt* and add it to the *Archive*.
        Remove all the k dominated points from the *Archive*.
  End for
temp= $\alpha *$temp.
End while
if *Archive-size* $>$ *SL*
  Cluster *Archive* to *HL* number of clusters.

Fig. 1. The AMOSA Algorithm

reduce the size to *HL* (the clustering procedure is explained below). That means initially *Archive* contains a maximum of *HL* number of solutions.

In the initialization phase it is possible to get an *Archive* of size one. In MOSA [17], in such cases, other newly generated solutions which are dominated by the archival solution will be indistinguishable. In contrast, the amount of domination as incorporated in AMOSA will distinguish between "more dominated" and "less dominated" solutions. However, in future we intend to use a more sophisticated scheme, in line with that adopted in MOSA.

B. Clustering the Archive Solutions

Clustering of the solutions in the *Archive* has been incorporated in AMOSA in order to explicitly enforce the diversity of the non-dominated solutions. In general, the size of the *Archive* is allowed to increase up to *SL* ($>$ *HL*), after which the solutions are clustered for grouping the solutions into *HL* clusters. Allowing the *Archive* size to increase upto *SL* not only reduces excessive calls to clustering, but also enables the formation of more spread out clusters and hence better diversity. Note that in the initialization phase, clustering is executed once even if the number of solutions in the *Archive* is less than *SL*, as long as it is greater than *HL*. This enables it to start with atmost *HL* non-dominated solutions and reduces excessive calls to clustering in the initial stages of the AMOSA process.

For clustering, the well-known Single linkage algorithm [29] is used. Here, the distance between any two clusters corresponds to the length of the shortest link between them. This is similar to the clustering algorithm used in SPEA [25], except that they have used average linkage method [29]. After *HL* clusters are obtained, the member within each cluster whose average distance to the other members is the minimum, is considered as the representative member of the cluster. A tie is resolved arbitrarily. The representative points of all the *HL* clusters are thereafter stored in the *Archive*.

C. Amount of Domination

As already mentioned, AMOSA uses the concept of amount of domination in computing the acceptance probability of a new solution. Given two solutions a and b, the amount of domination is defined as $\Delta dom_{a,b} = \prod_{i=1, f_i(a) \neq f_i(b)}^{M} \frac{|f_i(a) - f_i(b)|}{R_i}$ where M = number of objectives and R_i is the range of the ith objective. Note that in several cases, R_i may not be known *a priori*. In these situations, the solutions present in the *Archive* along with the new and the current solutions are used for computing it. The concept of $\Delta dom_{a,b}$ is illustrated pictorially in Figure 2 for a two objective case. $\Delta dom_{a,b}$ is used in AMOSA while computing the probability of acceptance of a newly generated solution.

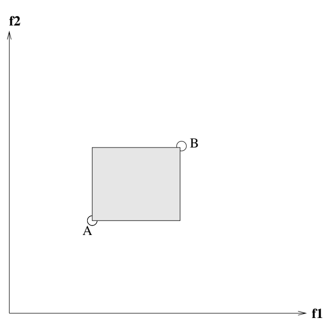


Fig. 2. Total amount of domination between the two solutions *A* and *B* = the area of the shaded rectangle

D. The Main AMOSA Process

One of the points, called *current-pt*, is randomly selected from $Archive$ as the initial solution at temperature *temp*=*Tmax*. The *current-pt* is perturbed to generate a new solution called *new-pt*. The domination status of *new-pt* is checked with respect to the *current-pt* and solutions in *Archive*.

Based on the domination status between *current-pt* and *new-pt* three different cases may arise. These are enumerated below.

- Case 1: *current-pt* dominates the *new-pt* and *k* ($k \geq 0$) points from the *Archive* dominate the *new-pt*.
  This situation is shown in Figure 3. Here Figure 3(a) and (b) denote the situations where $k = 0$ and $k \geq 1$ respectively. (Note that all the figures correspond to a two objective maximization problem.) In this case, the *new-pt* is selected as the *current-pt* with
  $$probability = \frac{1}{1 + exp(\Delta dom_{avg} * temp)}, \quad (2)$$
  where Δdom_{avg}=$\left(\left(\sum_{i=1}^{k} \Delta dom_{i,new-pt}\right) + \Delta dom_{current-pt,new-p}\right.$ $1)$. Note that Δdom_{avg} denotes the average amount of domination of the *new-pt* by $(k+1)$ points, namely, the *current-pt* and k points of the *Archive*. Also, as k increases, Δdom_{avg} will increase since here the dominating points that are farther away from the *new-pt* are contributing to its value.
  Lemma: When $k = 0$, the *current-pt* is on the archival front.
  Proof: In case this is not the case, then some point, say *A*, in the *Archive* dominates it. Since *current-pt* dominates the *new-pt*, by transitivity, *A* will also dominate the *new-pt*. However, we have considered that no other point in the *Archive* dominates the *new-pt* as $k = 0$. Hence proved.
  However if $k \geq 1$, this may or may not be true.
- Case 2: *current-pt* and *new-pt* are non-dominating with respect to each other.
  Now, based on the domination status of *new-pt* and members of *Archive*, the following three situations may arise.
  1) *new-pt* is dominated by *k* points in the *Archive* where $k \geq 1$. This situation is shown in Figure 4(a). The *new-pt* is selected as the *current-pt* with
  $$probability = \frac{1}{(1 + exp(\Delta dom_{avg} * temp))}, \quad (3)$$
  where $\Delta dom_{avg} = \sum_{i=1}^{k} (\Delta dom_{i,new-pt})/k$. Note that here the *current-pt* may or may not be on the archival front.
  2) *new-pt* is non-dominating with respect to the other points in the *Archive* as well. In this case the *new-pt* is on the same front as the *Archive* as shown in Figure 4(b). So the *new-pt* is selected as the *current-pt* and added to the *Archive*. In case the *Archive* becomes overfull (i.e., the *SL* is exceeded), clustering is performed to reduce the number of points to *HL*.
  3) *new-pt* dominates *k* ($k \geq$1) points of the *Archive*. This situation is shown in Figure 4(c). Again, the *new-pt* is selected as the *current-pt*, and added to the *Archive*. All the *k* dominated points are removed from the *Archive*. Note that here too the *current-pt* may or may not be on the archival front.
- Case 3: *new-pt* dominates *current-pt*
  Now, based on the domination status of *new-pt* and members of *Archive*, the following three situations may arise.
  1) *new-pt* dominates the *current-pt* but *k* ($k \geq 1$) points in the *Archive* dominate this *new-pt*. Note that this situation (shown in Figure 5(a)) may arise only if the

current-pt is not a member of the *Archive*. Here, the minimum of the difference of domination amounts between the *new-pt* and the k points, denoted by Δdom_{min}, of the *Archive* is computed. The point from the *Archive* which corresponds to the minimum difference is selected as the *current-pt* with $prob = \frac{1}{1+exp(-\Delta dom_{min})}$. Otherwise the *new-pt* is selected as the *current-pt*. Note that according to the SA paradigm, the *new-pt* should have been selected with probability 1. However, due to the presence of *Archive*, it is evident that solutions still better than *new-pt* exist. Therefore the *new-pt* and the dominating points in the *Archive* that is closest to the *new-pt* (corresponding to Δdom_{min}) compete for acceptance. This may be considered a form of informed reseeding of the annealer only if the *Archive* point is accepted, but with a solution closest to the one which would otherwise have survived in the normal SA paradigm.

2) *new-pt* is non-dominating with respect to the points in the *Archive* except the *current-pt* if it belongs to the *Archive*. This situation is shown in Figure 5(b). Thus *new-pt*, which is now accepted as the *current-pt*, can be considered as a new nondominated solution that must be stored in *Archive*. Hence *new-pt* is added to the *Archive*. If the *current-pt* is in the *Archive*, then it is removed. Otherwise, if the number of points in the *Archive* becomes more than the *SL*, clustering is performed to reduce the number of points to *HL*.
   Note that here the *current-pt* may or may not be on the archival front.
3) *new-pt* also dominates k ($k \geq 1$), other points, in the *Archive* (see Figure 5(c)). Hence, the *new-pt* is selected as the *current-pt* and added to the *Archive*, while all the dominated points of the *Archive* are removed. Note that here the *current-pt* may or may not be on the archival front.

The above process is repeated *iter* times for each temperature (*temp*). Temperature is reduced to $\alpha \times temp$, using the cooling rate of α till the minimum temperature, *Tmin*, is attained. The process thereafter stops, and the *Archive* contains the final non-dominated solutions.

Note that in AMOSA, as in other versions of multi-objective SA algorithms, there is a possibility that a new solution worse than the current solution may be selected. In most other MOEAs, e.g., NSGA-II, PAES, if a choice needs to be made between two solutions $\overline{x}$ and $\overline{y}$, and if $\overline{x}$ dominates $\overline{y}$, then $\overline{x}$ is always selected. It may be noted that in single objective EAs or SA, usually a worse solution also has a non-zero chance of surviving in subsequent generations; this leads to a reduced possibility of getting stuck at suboptimal regions. However, most of the MOEAs have been so designed that this characteristics is lost. The present simulated annealing based algorithm provides a way of incorporating this feature.

E. Complexity Analysis

The complexity analysis of AMOSA is provided in this section. The basic operations and their worst case complexities are as follows:

1) Archive initialization: O(*SL*).
2) Procedure to check the domination status of any two solutions: O(*M*), *M* = # objectives.
3) Procedure to check the domination status between a particular solution and the *Archive* members: O(*M* ×*SL*).
4) Single linkage clustering: O($SL^2 \times \log(SL)$) [30].
5) Clustering procedure is executed
   - once after initialization if $|\text{ND}| > HL$
   - after each (*SL*−*HL*) number of iterations.
   - at the end if final $|Archive| > HL$

   So maximum number of times the Clustering procedure is called=(*TotalIter*/(*SL*−*HL*))+2.
   Therefore, total complexity due to Clustering procedure is O((*TotalIter*/(*SL*−*HL*)) × $SL^2 \times \log(SL)$).

Total complexity of AMOSA becomes

$$(SL+M+M\times SL)\times(TotalIter)+\frac{TotalIter}{SL-HL}\times SL^2\times\log(SL). \quad (4)$$

Let *SL*= $\beta\times$*HL* where $\beta \geq 2$ and $HL = N$ where N is the population size in NSGA-II and archive size in PAES. Therefore overall complexity of the AMOSA becomes

$$(TotalIter) \times (\beta \times N + M + M \times \beta \times N + (\beta^2/(\beta-1)) \times N \times \log(\beta N)), \quad (5)$$

or,

$$O(TotalIter \times N \times (M + \log(N))). \quad (6)$$

Note that the total complexity of NSGA-II is O(*TotalIter*$\times M \times N^2$) and that of PAES is O(*TotalIter* $\times M \times N$). NSGA-II complexity depends on the complexity of non-dominated procedure. With the best procedure, the complexity is O(*TotalIter*$\times M \times N \times log(N)$).

IV. Simulation Results

In this section, we first describe comparison metrics used for the experiments. The performance analysis of both the binary-coded AMOSA and real-coded AMOSA are also provided in this section.

A. Comparison Metrics

In multi-objective optimization, there are basically two functionalities that an MOO strategy must achieve regarding the obtained solution set [1]. It should converge as close to the true PO front as possible and it should maintain as diverse a solution set as possible.

The first condition clearly ensures that the obtained solutions are near optimal and the second condition ensures that a wide range of trade-off solutions is obtained. Clearly, these two tasks cannot be measured with one performance measure adequately. A number of performance measures have been suggested in the past. Here we have mainly used three such

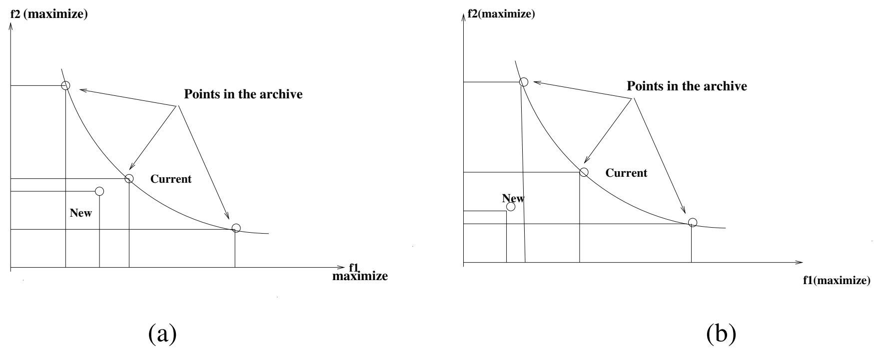


Fig. 3. Different cases when *New* is dominated by *Current* (a) *New* is non-dominating with respect to the solutions of *Archive* except *Current* if it is in the archive (b) Some solutions in the *Archive* dominate *New*

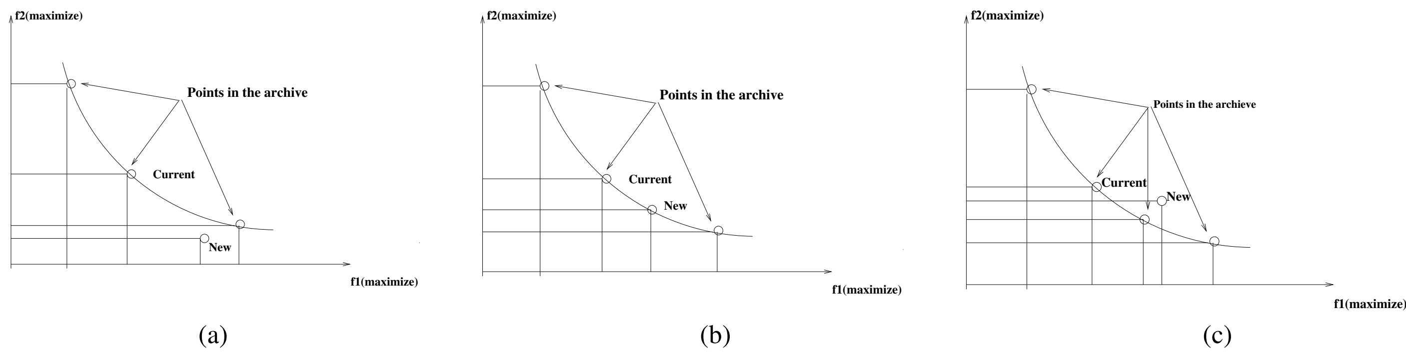


Fig. 4. Different cases when *New* and *Current* are non-dominating (a) Some solutions in *Archive* dominates *New* (b) *New* is non-dominating with respect to all the solutions of *Archive* (c) *New* dominates k ($k \geq 1$) solutions in the *Archive*

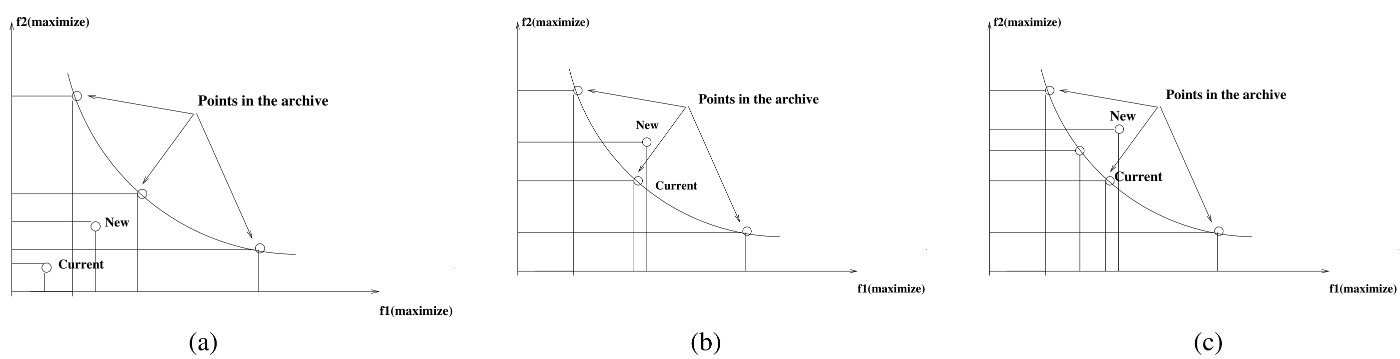


Fig. 5. Different cases when *New* dominates the *Current* (a) *New* is dominated by some solutions in *Archive* (b) *New* is non-dominating with respect to the solutions in the *Archive* except *Current*, if it is in the archive (c) *New* dominates some solutions of *Archive* other than *Current*

performance measures. The first measure is the *Convergence* measure γ [19]. It measures the extent of convergence of the obtained solution set to a known set of PO solutions. Lower the value of γ, better is the convergence of the obtained solution set to the true PO front. The second measure called *Purity* [21], [22] is used to compare the solutions obtained using different MOO strategies. It calculates the fraction of solutions from a particular method that remains nondominating when the final front solutions obtained from all the algorithms are combined. A value near to 1(0) indicates better (poorer) performance. The third measure named *Spacing* was first proposed by Schott [23]. It reflects the uniformity of the solutions over the non-dominated front. It is shown in [21] that this measure will fail to give adequate result in some situations. In order to overcome the above limitations, a modified measure, named *MinimalSpacing* is proposed in [21]. Smaller values of *Spacing* and *MinimalSpacing* indicate better performance.

It may be noted that if an algorithm is able to approximate only a portion of the true PO front, not its full extents, none of the existing measures, will be able to reflect this. In case of discontinuous PO front, this problem becomes severe when an algorithm totally misses a sub-front. Here a performance measure which is very similar to the measure used in [8] and [24] named *displacement* is used that is able to overcome this limitation. It measures how far the obtained solution set is from a known set of PO solutions. In order to compute *displacement* measure, a set P^* consisting of uniformly spaced solutions from the true PO front in the objective space is found (as is done while calculating γ). Then *displacement* is calculated as

$$displacement = \frac{1}{|P^*|} \times \sum_{i=1}^{|P^*|} (min_{j=1}^{|Q|} d(i,j)) \qquad (7)$$

where Q is the obtained set of final solutions, and $d(i,j)$ is the Euclidean distance between the *i*th solution of P^* and *j*th solution of Q. Lower the value of this measure, better is the convergence to and extent of coverage of the true PO front.

B. Comparison of Binary Encoded AMOSA with NSGA-II and PAES

Firstly, we have compared the binary encoded AMOSA with the binary-coded NSGA-II and PAES algorithm. For AMOSA binary mutation is used. Seven test problems have been considered for the comparison purpose. These are SCH1 and SCH2 [1], Deb1 and Deb4 [31], ZDT1, ZDT2, ZDT6 [1]. All the algorithms are executed ten times per problem and the results reported are the average values obtained for the ten runs. In NSGA-II the crossover probability (p_c) is kept equal to 0.9. For PAES the depth value *d* is set equal to 5. For AMOSA the cooling rate α is kept equal to 0.8. The number of bits assigned to encode each decision variable depends on the problem. For example in ZDT1, ZDT2 and ZDT6 which all are 30-variable problems, 10 bits are used to encode each variable, for SCH1 and SCH2 which are single variable problems and for Deb1 and Deb4 which are two variable problems, 20 bits are used to encode each decision variable. In all the approaches, binary mutation applied with a probability of $p_m = 1/l$, where l is the string length, is used as the perturbation operation. We have chosen the values of *Tmax* (maximum value of the temperature), *Tmin* (minimum value of the temperature) and *iter* (number of iterations at each temperature) so that total number of fitness evaluations of the three algorithms becomes approximately equal. For PAES and NSGA-II, identical parameter settings as suggested in the original studies have been used. Here the population size in NSGA-II, and archive sizes in AMOSA and PAES are set to 100. Maximum iterations for NSGA-II and PAES are 500 and 50000 respectively. For AMOSA, $Tmax = 200$, $Tmin = 10^{-7}$, $iter = 500$. The parameter values were determined after extensive sensitivity studies, which are omitted here to restrict the size of the article.

1) Discussions of the Results: The *Convergence* and *Purity* values obtained using the three algorithms is shown in Table I. AMOSA performs best for ZDT1, ZDT2, ZDT6 and Deb1 in terms of γ. For SCH1 all three are performing equally well. NSGA-II performs well for SCH2 and Dev4. Interestingly, for all the functions, AMOSA is found to provide more number of overall nondominated solutions than NSGA-II. (This is evident from the quantities in parentheses in Table I where $\frac{x}{y}$ indicates that on an average the algorithm produced y solutions of which *x* remained good even when solutions from other MOO strategies are combined). AMOSA took 10 seconds to provide the first PO solution compared to 32 seconds for NSGA-II in case of ZDT1. From Table I it is again clear that AMOSA and PAES are always giving more number of distinct solutions than NSGA-II.

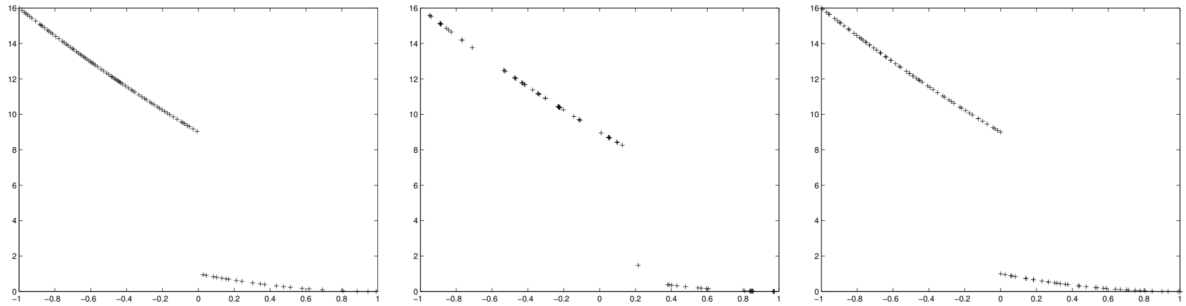
Table II shows the *Spacing* and *MinimalSpacing* measurements. AMOSA is giving the best performance of *Spacing* most of the times while PAES performs the worst. This is also evident from Figures 6 and 7 which show the final PO fronts of SCH2 and Deb4 obtained by the three methods for the purpose of illustration (due to lack of space final PO fronts given by three methods for some test problems are omitted). With respect to *MinimalSpacing* the performances of AMOSA and NSGA-II are comparable.

Table III shows the value of *displacement* for five problems, two with discontinuous and three with continuous PO fronts. AMOSA performs the best in almost all the cases. The utility of the new measure is evident in particular for Deb4 where PAES performs quite poorly (see Figure 7). Interestingly the *Convergence* value for PAES (Table I) is very good here, though the *displacement* correctly reflects that the PO front has been represented very poorly.

Table IV shows the time taken by the algorithms for the different test functions. It is seen that PAES takes less time in six of the seven problems because of its smaller complexity. AMOSA takes less time than NSGA-II in 30 variable problems like ZDT1, ZDT2, 10 variable problem ZDT6. But for single and two variable problems SCH1, SCH2, Deb1 and Deb4, AMOSA takes more time than NSGA-II. This may be due to complexity of its clustering procedure. Generally for single or two variable problems this procedure dominates the crossover and ranking procedures of NSGA-II. But for 30 variable problems the scenario is reversed. This is because of the increased complexity of ranking and crossover (due to increased string length) in NSGA-II.

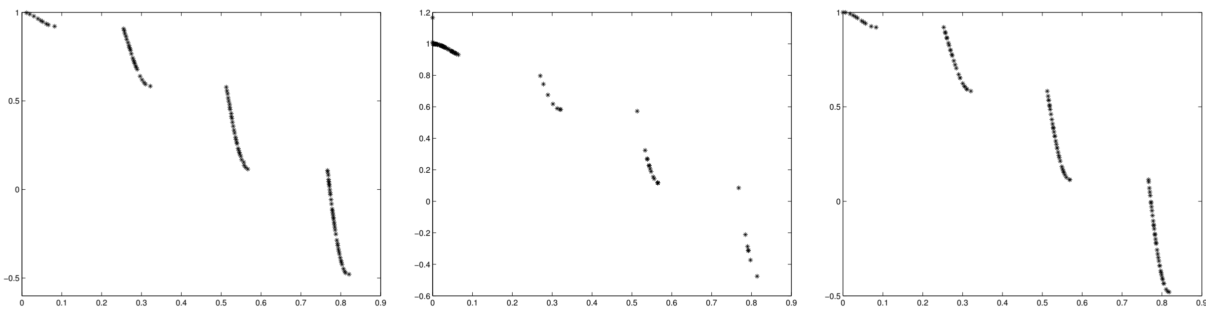
C. Comparison of Real-coded AMOSA with the Algorithm of Smith et al. [17] and Real-coded NSGA-II

The real-coded version of the proposed AMOSA has also been implemented. The mutation is done as suggested in [17]. Here a new string is generated from the the old string *x* by perturbing only one parameter or decision variable of *x*. The parameter to be perturbed is chosen at random and perturbed with a random variable ϵ drawn from a Laplacian distribution, $p(\epsilon) \propto e^{-\|\sigma\epsilon\|}$, where the scaling factor σ sets magnitude of the perturbation. A fixed scaling factor equals to 0.1 is used for mutation. The initial temperature is selected by the procedure mentioned in [17]. That is, the initial temperature, $Tmax$, is calculated by using a short 'burn-in' period during which all solutions are accepted and setting the temperature equal to the average positive change of energy divided by ln(2). Here $Tmin$ is kept equal to 10^{-5} and the temperature is adjusted according to $T_k = \alpha^k Tmax$, where α is set equal to 0.8. For NSGA-II population size is kept equal to 100 and total number of generations is set such that the total number of function evaluations of AMOSA and NSGA-II are the same. For AMOSA the archive size is set equal to 100. (However, in a part of investigations, the archive size



(a) (b) (c)

Fig. 6. The final non-dominated front obtained by (a) AMOSA (b) PAES (c) NSGA-II for SCH2



(a) (b) (c)

Fig. 7. The final non-dominated front for Deb4 obtained by (a) AMOSA (b) PAES (c) NSGA-II

TABLE I

Convergence AND *Purity* MEASURES ON THE TEST FUNCTIONS FOR BINARY ENCODING

Test	$Convergence$			$Purity$		
Problem	AMOSA	PAES	NSGA-II	AMOSA	PAES	NSGA-II
SCH1	0.0016	0.0016	0.0016	0.9950(99.5/100)	0.9850(98.5/100)	1(94/94)
SCH2	0.0031	0.0015	0.0025	0.9950(99.5/100)	0.9670(96.7/100)	0.9974(97/97.3)
ZDT1	0.0019	0.0025	0.0046	0.8350(83.5/100)	0.6535(65.4/100)	0.970(68.64/70.6)
ZDT2	0.0028	0.0048	0.0390	0.8845(88.5/100)	0.4050(38.5/94.9)	0.7421(56.4/76)
ZDT6	0.0026	0.0053	0.0036	1(100/100)	0.9949(98.8/99.3)	0.9880(66.5/67.3)
Deb1	0.0046	0.0539	0.0432	0.91(91/100)	0.718(71.8/100)	0.77(71/92)
Deb4	0.0026	0.0025	0.0022	0.98(98/100)	0.9522(95.2/100)	0.985(88.7/90)

TABLE II

Spacing AND *MinimalSpacing* MEASURES ON THE TEST FUNCTIONS FOR BINARY ENCODING

Test	$Spacing$			$MinimalSpacing$		
Problem	AMOSA	PAES	NSGA-II	AMOSA	PAES	NSGA-II
$SCH1$	0.0167	0.0519	0.0235	0.0078	0.0530	0.0125
$SCH2$	0.0239	0.5289	0.0495	$N.A.$	$N.A.$	$N.A.$
$ZDT1$	0.0097	0.0264	0.0084	0.0156	0.0265	0.0147
$ZDT2$	0.0083	0.0205	0.0079	0.0151	0.0370	0.0130
$ZDT6$	0.0051	0.0399	0.0089	0.0130	0.0340	0.0162
$Deb1$	0.0166	0.0848	0.0475	0.0159	0.0424	0.0116
$Deb4$	0.0053	0.0253	0.0089	$N.A.$	$N.A.$	$N.A.$

is kept unlimited as in [17]. The results are compared to those obtained by MOSA [17] and provided in [32].) AMOSA is executed for a total of 5000, 1000, 15000, 5000, 1000, 5000 and 9000 run lengths respectively for DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ5 and DTLZ6. Total number of iterations, $iter$, per temperature is set accordingly. We have run real-coded NSGA-II (code obtained from KANGAL site: http://www.iitk.ac.in/kangal/codes.html). For NSGA-II the following parameter setting is used: probability of crossover =0.99, probability of mutation=(1/l), where l is the string

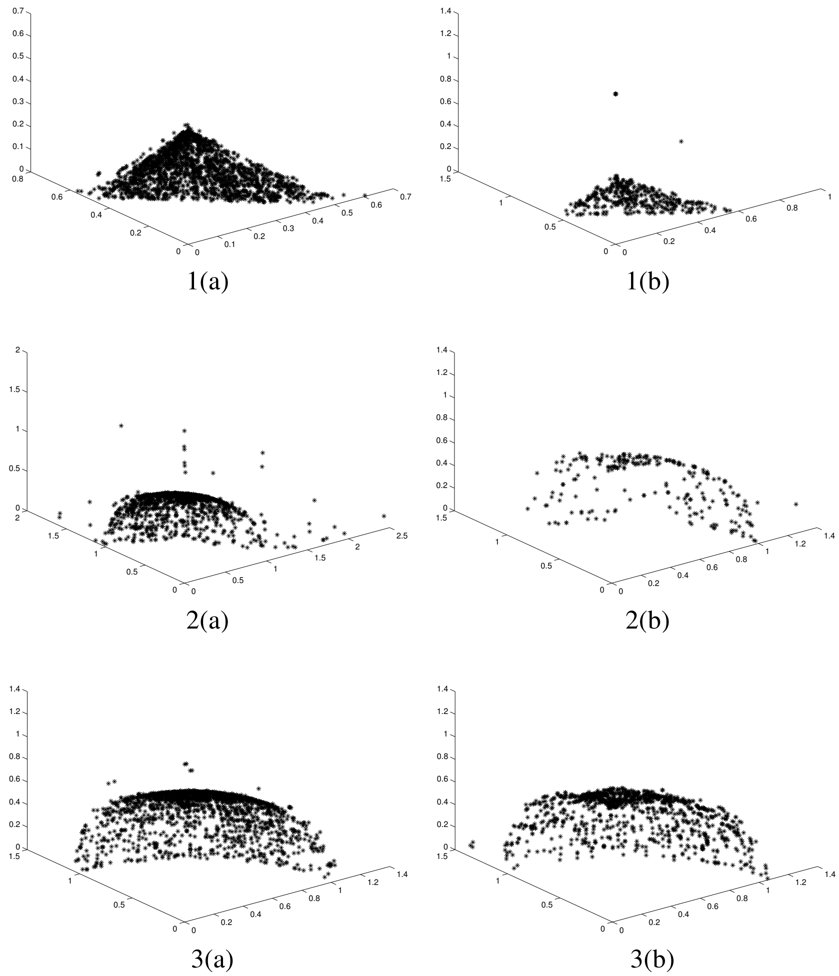


Fig. 8. The final non-dominated front obtained by (a) AMOSA (b) MOSA for the test problems (1) DTLZ1 (2) DTLZ2 (3) DTLZ3

TABLE III

NEW MEASURE *displacement* ON THE TEST FUNCTIONS FOR BINARY ENCODING

$Algorithm$	$SCH2$	$Deb4$	$ZDT1$	$ZDT2$	$ZDT6$
AMOSA	0.0230	0.0047	0.0057	0.0058	0.0029
PAES	0.6660	0.0153	0.0082	0.0176	0.0048
NSGA-II	0.0240	0.0050	0.0157	0.0096	0.0046

TABLE IV

TIME TAKEN BY DIFFERENT PROGRAMS (IN SEC) FOR BINARY ENCODING

$Algorithm$	$SCH1$	$SCH2$	$Deb1$	$Deb4$	$ZDT1$	$ZDT2$	$ZDT6$
AMOSA	15	14.5	20	20	58	56	12
PAES	6	5	5	15	17	18	16
NSGA-II	11	11	14	14	77	60	21

length, distribution index for the crossover operation=10, distribution index for the mutation operation=100.

In MOSA [17] authors have used unconstrained archive size. Note that the archive size of AMOSA and the population size of NSGA-II are both 100. For the purpose of comparison with MOSA that has an unlimited archive [17], the clustering procedure (adopted for AMOSA), is used to reduce the number of solutions of [32] to 100. Comparison is performed in terms of *Purity*, *Convergence* and *Minimal Spacing*. Table V shows the *Purity*, *Convergence*, *Minimal Spacing* measurements for DTLZ1-DTLZ6 problems obtained after application of AMOSA, MOSA and NSGA-II. It can

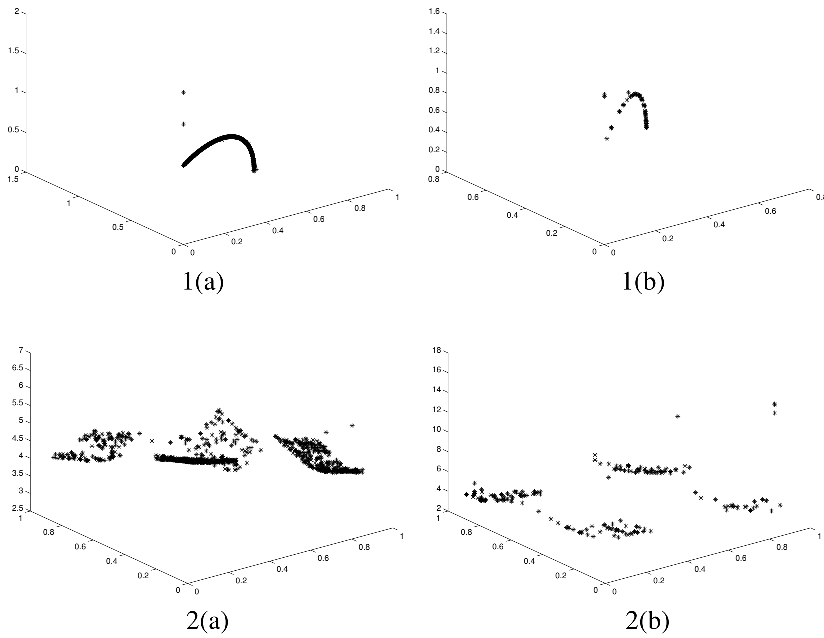


Fig. 9. The final non-dominated front obtained by (a) AMOSA (b) MOSA for the test problems (1) DTLZ5 (2) DTLZ6

be seen from this table that AMOSA performs the best in terms of *Purity* and *Convergence* for DTLZ1, DTLZ3, DTLZ5, DTLZ6. In DTLZ2 and DTLZ4 the performance of MOSA is marginally better than that of AMOSA. NSGA-II performs the worst among all. Table V shows the *Minimal Spacing* values obtained by the 3 algorithms for DTLZ1-DTLZ6. AMOSA performs the best in all the cases.

As mentioned earlier, for comparing the performance of MOSA (by considering the results reported in [32]), a version of AMOSA without clustering and with unconstrained archive is executed. The results reported here are the average over 10 runs. Table VI shows the corresponding *Purity*, *Convergence* and *Minimal Spacing* values. Again AMOSA performs much better than MOSA for all the test problems except DTLZ4. For DTLZ4, the MOSA performs better than that of AMOSA in terms of both *Purity* and *Convergence* values. Figure 8 shows final Pareto-optimal front obtained by AMOSA and MOSA for DTLZ1-DTLZ3 while Figure 9 shows the same for DTLZ5 and DTLZ6. As can be seen from the figures, AMOSA appears to be able to better approximate the front with more dense solutions as compared to MOSA.

It was mentioned in [33] that for a particular test problem, almost 40% of the solutions provided by an algorithm with truncation of archive got dominated by the solutions provided by an algorithm without archive truncation. However the experiments we conducted did not adequately justify this finding. Let us denote the set of solutions of AMOSA with and without clustering as S_c and S_{wc} respectively. We found that for DTLZ1, 12.6% of S_c were dominated by S_{wc}, while 4% of S_{wc} were dominated by S_c. For DTLZ2, 5.1% of S_{wc} were dominated by S_c while 5.4% of S_c were dominated by S_{wc}. For DTLZ3, 22.38% of S_{wc} were dominated by S_c while 0.53% of S_c were dominated by S_{wc}. For DTLZ4, all the members of S_{wc} and S_c are non-dominating to each other and the solutions are same. Because execution of AMOSA without clustering doesn't provide more than 100 solutions. For DTLZ5, 10.4% of S_{wc} were dominated by S_c while 0.5% of S_c were dominated by S_{wc}. For DTLZ6, all the members of S_{wc} and S_c are non-dominating to each other.

To have a look at the performance of the AMOSA on a four-objective problem, we apply AMOSA and NSGA-II to the 13-variable DTLZ2 test problem. This is referred to as DTLZ2_4. The problem has a spherical Pareto-front in four dimensions given by the equation: $f_1^2+f_2^2+f_3^2+f_4^2=1$ with $f_i \in [0,1]$ for $i=1$ to 4. Both the algorithms are applied for a total of 30,000 function evaluations (for NSGA-II popsize=100 and number of generations=300) and the *Purity*, *Convergence* and *Minimal Spacing* values are shown in Table VII. AMOSA performs much better than NSGA-II in terms of all the three measures.

The proposed AMOSA and NSGA-II are also compared for DTLZ1_5 (9-variable 5 objective version of the test problem DTLZ1), DTLZ1_10 (14-variable 10 objective version of DTLZ1) and DTLZ1_15 (19 variable 15 objective version of DTLZ1). The three problems have a spherical Pareto-front in their respective dimensions given by the equation $\sum_{i=1}^{M} f_i = 0.5$ where M is the total number of objective functions. Both the algorithms are executed for a total of 1,00,000 function evaluations for these three test problems (for NSGA-II popsize=200, number of generations=500) and the corresponding *Purity*, *Convergence* and *Minimal Spacing* values are shown in Table VII. *Convergence* value indicates that NSGA-II doesn't converge to the true PO front where as AMOSA reaches the true PO front for all the three cases.

TABLE V

Convergence AND *Purity* MEASURES ON THE 3 OBJECTIVE TEST FUNCTIONS WHILE *Archive* IS BOUNDED TO 100

Test Problem	$Convergence$			$Purity$			$MinimalSpacing$		
	AMOSA	MOSA	NSGA-II	AMOSA	MOSA	NSGA-II	AMOSA	MOSA	NSGA-II
$DTLZ1$	0.01235	0.159	13.695	0.857 (85.7/100)	0.56 (28.35/75)	0.378 (55.7/100)	0.0107	0.1529	0.2119
$DTLZ2$	0.014	0.01215	0.165	0.937 (93.37/100)	0.9637 (96.37/100)	0.23 (23.25/100)	0.0969	0.1069	0.1236
$DTLZ3$	0.0167	0.71	20.19	0.98 (93/95)	0.84 (84.99/100)	0.232 (23.2/70.6)	0.1015	0.152	0.14084
$DTLZ4$	0.28	0.21	0.45	0.833 (60/72)	0.97 (97/100)	0.7 (70/100)	0.20	0.242	0.318
$DTLZ5$	0.00044	0.0044	0.1036	1 (97/97)	0.638 (53.37/83.6)	0.05 (5/100)	0.0159	0.0579	0.128
$DTLZ6$	0.043	0.3722	0.329	0.9212 (92.12/100)	0.7175 (71.75/100)	0.505 (50.5/100)	0.1148	0.127	0.1266

TABLE VI

Convergence, *Purity* AND *Minimal Spacing* MEASURES ON THE 3 OBJECTIVES TEST FUNCTIONS BY AMOSA AND MOSA WHILE *Archive* IS UNBOUNDED

Test Problem	$Convergence$		$Purity$		$MinimalSpacing$	
	AMOSA	MOSA	AMOSA	MOSA	AMOSA	MOSA
$DTLZ1$	0.010	0.1275	0.99(1253.87/1262.62)	0.189(54.87/289.62)	0.064	0.083.84
$DTLZ2$	0.0073	0.0089	0.96(1074.8/1116.3)	0.94(225/239.2)	0.07598	0.09595
$DTLZ3$	0.013	0.025	0.858(1212/1412.8)	0.81(1719/2003.9)	0.0399	0.05
$DTLZ4$	0.032	0.024	0.8845(88.5/100)	0.4050(38.5/94.9)	0.1536	0.089
$DTLZ5$	0.0025	0.0047	0.92(298/323.66)	0.684(58.5/85.5)	0.018	0.05826
$DTLZ6$	0.0403	0.208	0.9979(738.25/739.75)	0.287(55.75/194.25)	0.0465	0.0111

TABLE VII

Convergence, *Purity* AND *Minimal Spacing* MEASURES ON THE DTLZ2_4, DTLZ1_5, DTLZ1_10 AND DTLZ1_15 TEST FUNCTIONS BY AMOSA AND NSGA-II

Test Problem	$Convergence$		$Purity$		$MinimalSpacing$	
	AMOSA	NSGA-II	AMOSA	NSGA-II	AMOSA	NSGA-II
$DTLZ2\_4$	0.2982	0.4563	0.9875(98.75/100)	0.903(90.3/100)	0.1876	0.22
$DTLZ1\_5$	0.0234	306.917	1	0	0.1078	0.165
$DTLZ1\_10$	0.0779	355.957	1	0	0.1056	0.2616
$DTLZ1\_15$	0.193	357.77	1	0	0.1	0.271

The *Purity* measure also indicates this. The results on many-objective optimization problems show that AMOSA peforms much better than NSGA-II. These results support the fact that Pareto ranking-based MOEAs such as NSGA-II do not work well on many-objective optimization problems as pointed out in some recent studies [26], [27].

D. Discussion on Annealing Schedule

The annealing schedule of an SA algorithm consists of (i) initial value of temperature ($Tmax$), (ii) cooling schedule, (iii) number of iterations to be performed at each temperature and (iv) stopping criterion to terminate the algorithm. Initial value of the temperature should be so chosen that it allows the SA to perform a random walk over the landscape. Some methods to select the initial temperature are given in detail in [18]. In this article, as in [17], we have set the initial temperature to achieve an initial acceptance rate of approximately 50% on derogatory proposals. This is described in Section VI.C.

Cooling schedule determines the functional form of the change in temperature required in SA. The most frequently used decrement rule, also used in this article, is the geometric schedule given by: $T_{k+1} = \alpha \times T_k$, where α $(0 < \alpha < 1)$ denotes the cooling factor. Typically the value of α is chosen in the range between 0.5 and 0.99. This cooling schedule has the advantage of being very simple. Some other cooling schedules available in the literature are logarithmic, Cauchy and exponential. More details about these schedules are available in [18]. The cooling schedule should be so chosen that it is able to strike a good balance between exploration and exploitation of the search space. In order to investigate the performance of AMOSA with another cooling schedule, the following is considered (obtained from http://members.aol.com/btluke/simanf1.htm):

$$T_i = T_0 \left(\frac{T_N}{T_0}\right)^{i/N}.$$

Here N is the total number of iterations, T_N is the final temperature and T_0 is the initial temperature. T_i is the temperature at iteration i. AMOSA with the above cooling schedule is applied on ZDT1. The *Convergence* and *Minimal Spacing* values obtained are 0.008665 and 0.017 respectively. Comparing with the corresponding values in Table I and II it is found that the results with this cooling schedule are somewhat poorer. However, an exhaustive sensitivity study needs to be performed for AMOSA.

The third component of an annealing schedule is the number of iterations performed at each temperature. It should be so chosen that the system is sufficiently close to the stationary distribution at that temperature. As suggested in [18], the value of the number of iterations should be chosen depending on the nature of the problem. Several criteria for termination of an SA process have been developed. In some of them, the total number of iterations that the SA procedure must execute is given, where as in some other, the minimum value of the temperature is specified. Detailed discussion on this issue can be found in [18].

V. Discussion and Conclusions

In this article a simulated annealing based multi-objective optimization algorithm has been proposed. The concept of amount of domination is used in solving the multi-objective optimization problems. In contrast to most other MOO algorithms, AMOSA selects dominated solutions with a probability that is dependent on the amount of domination measured in terms of the hypervolume between the two solutions in the objective space. The results of binary-coded AMOSA are compared with those of two existing well-known multi-objective optimization algorithms - NSGA-II (binary-coded) [19] and PAES [20] for a suite of seven 2-objective test problems having different complexity levels. In a part of the investigation, comparison of the real-coded version of the proposed algorithm is conducted with a very recent multi-objective simulated annealing algorithm MOSA [17] and real-coded NSGA-II for six 3-objective test problems. Real-coded AMOSA is also compared with real-coded NSGA-II for some 4, 5, 10 and 15 objective test problems. Several different comparison measures like *Convergence*, *Purity*, *MinimalSpacing*, and *Spacing*, and the time taken are used for the purpose of comparison. In this regard, a measure called *displacement* has also been used that is able to reflect whether a front is close to the PO front as well as its extent of coverage. A complexity analysis of AMOSA is performed. It is found that its complexity is more than that of PAES but smaller than that of NSGA-II.

It is seen from the given results that the performance of the proposed AMOSA is better than that of MOSA and NSGA-II in a majority of the cases, while PAES performs poorly in general. AMOSA is found to provide more distinct solutions than NSGA-II in each run for all the problems; this is a desirable feature in MOO. AMOSA is less time consuming than NSGA-II for complex problems like ZDT1, ZDT2 and ZDT6. Moreover, for problems with many objectives, the performance of AMOSA is found to be much better than that of NSGA-II. This is an interesting and appealing feature of AMOSA since Pareto ranking-based MOEAs, such as NSGA-II [19] do not work well on many-objective optimization problems as pointed out in some recent studies [26], [27]. An interesting feature of AMOSA, as in other versions of multi-objective SA algorithms, is that it has a non-zero probability of allowing a dominated solution to be chosen as the current solution in favour of a dominating solution. This makes the problem less greedy in nature; thereby leading to better performance for complex and/or deceptive problems. Note that it may be possible to incorporate this feature as well as the concept of amount of domination in other MOO algorithms in order to improve the performance.

There are several ways in which the proposed AMOSA algorithm may be extended in future. The main time consuming procedure in AMOSA is the clustering part. Some other more efficient clustering techniques or even the PAES like grid based strategy, can be incorporated for improving its performance. Implementation of AMOSA with unconstrained archive is another interesting area to pursue in future. An algorithm, unless analyzed theoretically, is good for only the experiments conducted. Thus a theoretical analysis of AMOSA needs to be performed in the future in order to study its convergence properties. Authors are currently trying to develop a proof for the convergence of AMOSA in the lines of the proof for single objective SA given by Geman and Geman [4]. As has been mentioned in [18], there are no firm guidelines for choosing the parameters in an SA-based algorithm. Thus, an extensive sensitivity study of AMOSA with respect to its different parameters, notably the annealing schedule, needs to be performed. Finally, application of AMOSA to several real-life domains e.g., VLSI system design [34], remote sensing imagery [35], data mining and Bioinformatics [36], needs to be demonstrated. The authors are currently working in this direction.

References

[1] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. England: John Wiley and Sons, Ltd, 2001.

[2] C. Coello, D. V. Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston: Kluwer Academic Publishers, 2002.

[3] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[4] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE-Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[5] S. Bandyopadhyay, U. Maulik, and M. K. Pakhira, "Clustering using simulated annealing with probabilistic redistribution," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 2, pp. 269–285, 2001.

[6] R. Caves, S. Quegan, and R. White, "Quantitative comparison of the performance of SAR segmentation algorithms," *IEEE Transactions on Image Proc.*, vol. 7, no. 11, pp. 1534–1546, 1998.

[7] U. Maulik, S. Bandyopadhyay, and J. Trinder, "SAFE: An efficient feature extraction technique," *Journal of Knowledge and Information Systems*, vol. 3, pp. 374–387, 2001.

[8] P. Czyzak and A. Jaszkiewicz, "Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, pp. 34–47, 1998.

[9] M. Hapke, A. Jaszkiewicz, and R. Slowinski, "Pareto simulated annealing for fuzzy multi-objective combinatorial optimization," *Journal of Heuristics*, vol. 6, no. 3, pp. 329–345, 2000.

[10] A. Suppapitnarm, K. A. Seffen, G. T. Parks, and P. Clarkson, "A simulated annealing algorithm for multiobjective optimization," *Engineering Optimization*, vol. 33, pp. 59–85, 2000.

[11] D. K. Nam and C. Park, "Multiobjective simulated annealing: a comparative study to evolutionary algorithms," *Int. J. Fuzzy Systems*, vol. 2, no. 2, pp. 87–97, 2000.

[12] E. L. Ulungu, J. Teghaem, P. Fortemps, and D. Tuyttens, "MOSA method: a tool for solving multiobjective combinatorial decision problems," *J. multi-criteria decision analysis*, vol. 8, pp. 221–236, 1999.

[13] D. Tuyttens, J. Teghem, and N.El-Sherbeny, "A particular multiobjective vehicle routing problem solved by simulated annealing," *Metaheuristics for multiobjective optimization*, vol. 535, pp. 133–152, 2003.

[14] I. Das and J. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," *Structural Optimization*, vol. 14, no. 1, pp. 63–69, 1997.

[15] A. Jaszkiewicz, "Comparison of local search-based metaheuristics on the multiple objective knapsack problem," *Foundations of computer and decision sciences*, vol. 26, no. 1, pp. 99–120, 2001.

[16] B. Suman, "Study of self-stopping PDMOSA and performance measure in multiobjective optimization," *Computers and Chemical Engineering*, vol. 29, no. 5, pp. 1131–1147, 15 April 2005.

[17] K. Smith, R. Everson, and J. Fieldsend, "Dominance measures for multi-objective simulated annealing," in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC'04)*, 2004, pp. 23–30.

[18] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the Operational Research Society*, vol. 57, no. 10, pp. 1143–1160, 2006.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002.

[20] J. D. Knowles and D. W. Corne, "Approxmating the nondominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.

[21] S. Bandyopadhyay, S. K. Pal, and B. Aruna, "Multi-objective GAs, quantitative indices and pattern classification," *IEEE Transactions on Systems, Man and Cybernetics-B*, vol. 34, no. 5, pp. 2088–2099, 2004.

[22] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, August, 1998.

[23] J. R. Schott, "Fault tolerant design using single and multi-criteria genetic algorithms," Ph.D. dissertation, 1995.

[24] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic serach and local serach in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE-Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 721–741, 1984.

[25] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization: The strength pareto approach," Gloriastrasse 35, CH-8092 Zurich, Switzerland, Tech. Rep. 43, 1998.

[26] E. J. Hughes, "Evolutionary many-objective optimization: Many once or one many?" in *Proceedings of 2005 Congress on Evolutionary Computation*, Edinburgh, Scotland, UK, September 2-5, 2005, pp. 222–227.

[27] H. Ishibuchi, T. Doi, and Y. Nojima, "Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms," *In Parallel Problem Solving from Nature IX (PPSN-IX)*, vol. 4193, pp. 493–502, September 2006.

[28] J. Fieldsend, R. Everson, and S. Singh, "Using unconstrained elite archives for multi-objective optimisation," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 305–323, 2003.

[29] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[30] G. Toussaint, "Pattern recognition and geometrical complexity," in *Proc. Fifth International Conf. on Pattern Recognition*, Miami Beach, December 1980, pp. 1324–1347.

[31] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.

[32] http://www.dcs.ex.ac.uk/people/kismith/mosa/results/tec/.

[33] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[34] S. Saha, S.Sur-Kolay, S. Bandyopadhyay, and P. Dasgupta, "Multi-objective genetic algorithm for k-way equipartitioning of a point set with application to CAD-VLSI," in *Proceedings of 9th International Conference on Information Technology (ICIT 2006)*. IEEE Computer Society, December, 2006, pp. 281–284.

[35] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *IEEE Transactions Geoscience and Remote Sensing (In Press)*, 2006.

[36] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Efficient two-stage fuzzy clustering of microarray gene expression data," in *Proceedings of 9th International Conference on Information Technology (ICIT 2006)*. IEEE Computer Society, December, 2006, pp. 11–14.

Sanghamitra Bandyopadhyay (SM'05) did her Bachelors in Physics and Computer Science in 1988 and 1991 respectively. Subsequently, she did her Masters in Computer Science from Indian Institute of Technology (IIT), Kharagpur in 1993 and Ph.D in Computer Science from Indian Statistical Institute, Calcutta in 1998. Currently she is an Associate Professor, Indian Statistical Institute, Kolkata, India. Dr. Bandyopadhyay is the first recipient of Dr Shanker Dayal Sharma Gold Medal and Institute Silver Medal for being adjudged the best all round post graduate performer in IIT, Kharagpur in 1994. She has worked in Los Alamos National Laboratory, Los Alamos, USA, in 1997, University of New South Wales, Sydney, Australia, in 1999, Department of Computer Science and Engineering, University of Texas at Arlington, USA, in 2001, University of Maryland, Baltimore County, USA, in 2004, Fraunhofer Institute AiS, St. Augustin, Germany, in 2005 and Tsinghua University, China, in 2006. She has delivered lectures at Imperial College, London, UK, Monash University, Australia, University of Aizu, Japan, University of Nice, France, University Kebangsaan Malaysia, Kuala Lumpur, Malaysia, and also made academic visits to many more Institutes/Universities around the world. She is a co-author of 3 books and more than 100 research publications. Dr. Bandyopadhyay received the Indian National Science Academy (INSA) and the Indian Science Congress Association (ISCA) Young Scientist Awards in 2000, as well as the Indian National Academy of Engineering (INAE) Young Engineers' Award in 2002. She has guest edited several journal special issues including IEEE Transactions on Systems, Man and Cybernetics, Part - B. Dr. Bandyopadhyay has been the Program Chair, Tutorial Chair, Associate Track Chair and a Member of the program committee of many international conferences. Her research interests include Pattern Recognition, Data Mining, Evolutionary and Soft Computation, Bioinformatics and Parallel and Distributed Systems.

Sriparna Saha received her B.Tech degree in Computer Science and Engineering from Kalyani Govt. Engineering College, University of Kalyani, India in 2003. She did her M.Tech in Computer Science from Indian Statistical Institute, Kolkata in 2005. She is the recipient of Lt Rashi Roy Memorial Gold Medal from Indian Statistical Institute for outstanding performance in M.Tech (Computer Science). At present she is pursuing her Ph.D. from Indian Statistical Institute, Kolkata, India. She is a co-author of about 10 papers. Her research interests include Multiobjective Optimization, Pattern Recognition, Evolutionary Algorithms, and Data Mining.

Ujjwal Maulik (SM'05) is currently a Professor in the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India. Dr. Maulik did his Bachelors in Physics and Computer Science in 1986 and 1989 respectively. Subsequently, he did his Masters and Ph.D in Computer Science in 1991 and 1997 respectively. He was the Head of the school of Computer Science and Technology Department of Kalyani Govt. Engg. College, Kalyani, India during 1996-1999. Dr. Maulik has worked in Center for Adaptive Systems Application, Los Alamos, New Mexico, USA in 1997, University of New South Wales, Sydney, Australia in 1999, University of Texas at Arlington, USA in 2001, Univ. of Maryland Baltimore County, USA in 2004 and Fraunhofer Institute AiS, St. Augustin, Germany in 2005. He has also visited many Institutes/Universities around the world for invited lectures and collaborative research. He is a senior member of IEEE and a Fellow of IETE, India. Dr. Maulik is a co-author of 2 books and about 100 research publications. He is the recipient of the Govt. of India BOYSCAST fellowship in 2001. Dr. Maulik has been the Program Chair, Tutorial Chair and a Member of the program committee of many international conferences and workshops. His research interests include Artificial Intelligence and Combinatorial Optimization, Soft Computing, Pattern Recognition, Data Mining, Bioinformatics, VLSI and Distributed System.

Kalyanmoy Deb is currently a Professor of Mechanical Engineering at Indian Institute of Technology Kanpur, India and is the director of Kanpur Genetic Algorithms Laboratory (KanGAL). He is the recipient of the prestigious Shanti Swarup Bhatnagar Prize in Engineering Sciences for the year 2005. He has also received the 'Thomson Citation Laureate Award' from Thompson Scientific for having highest number of citations in Computer Science during the past ten years in India. He is a fellow of Indian National Academy of Engineering (INAE), Indian National Academy of Sciences, and International Society of Genetic and Evolutionary Computation (ISGEC). He has received Fredrick Wilhelm Bessel Research award from Alexander von Humboldt Foundation in 2003. His main research interests are in the area of computational optimization, modeling and design, and evolutionary algorithms. He has written two text books on optimization and more than 190 international journal and conference research papers. He has pioneered and a leader in the field of evolutionary multi-objective optimization. He is associate editor of two major international journals and an editorial board members of five major journals. More information about his research can be found from http://www.iitk.ac.in/kangal/deb.htm.