A Multi-objective Genetic Algorithm with Relative Distance: Method, Performance Measures and Constraint Handling

Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay and Sankar Kumar Pal
Machine Intelligence Unit,
Indian Statistical Institute,
203 B. T. Road, Kolkata 700108
Email: {praveen_r, sanghami, sankar}@isical.ac.in

Abstract

A novel Multi-Objective Evolutionary Algorithm (MOEA), called Multi-objective Genetic Algorithm with Relative Distance (MOGARD) is described. A novel relative distance parameter that ensures convergence to the Pareto optimal front and a nearest neighbour based method for maintaining diversity in the non-dominated set is used. Two novel performance measures are formulated to estimate the performance of the MOEAs. A penalty based constraint handling concept is introduced in MOGARD, for handling constraints. Experimental results demonstrate the superiority of MOGARD on several test problems, as compared to other recent and well known algorithms.

1. Introduction

The primary objective in multi-objective optimization (MOO) is to evolve a set of solutions which are as close as possible to the Pareto-optimal (PO) front, while being as diverse as possible. In this article, a new evolutionary MOO algorithm called "Multi-objective Genetic Algorithm with Relative Distance (MOGARD)" is proposed. MOGARD uses the concept of relative distance parameter as the fitness function that helps in better convergence to the true PO-front. A novel diversity parameter is used to ensure a wide spread of the solutions. MOGARD incorporates the concept of elitism, using the archive concept as in SPEA2 [9].

Two novel performance measures called *relative convergence* and *hypergrid measure* are proposed for the evaluation of MOEAs to assess their convergence and diversity characteristics respectively. Comparative results of MOGARD with NSGA-II [4], PESA-II [2] and SPEA2 are provided for different standard test functions of two and three objectives, and one to thirty variables, in terms of these performance measures.

Constraint handling is an important issue in MOO. In [4], the concept of constrained domination is proposed, where infeasible solutions are always discarded even when compared to a feasible but extremely poor solution. In order to overcome this limitation, MOGARD incorporates a penalty based approach, as suggested for single objective GAs [7].

2 Multi-objective Genetic Algorithm with Relative Distance (MOGARD)

The basic structure of MOGARD is given in **Algorithm:1**. Different steps are described below in brief. The parameters of the search space in MOGARD are encoded as string like structures called *chromosomes*. A *fitness* value is assigned to each chromosome that denotes the degree of goodness of the encoded solution. The fitness F_i of the i^{th} chromosome is computed as: $F_i = f^i_{rd}$, where f^i_{rd} is the relative distance parameter proposed in this article.

• Relative Distance Parameter (f^i_{rd}): The significance of relative distance parameter is to give higher priority to those dominated solutions, that are closer to the non-dominating solutions which are dominating them. As an example consider the minimization of two functions in Figure 1(a). The solutions 'g', 'h' and 'j' are dominated by the non-dominating solution 'c'. As per NSGA-II and SPEA2, 'g', 'h' and 'j' would have the same fitness. However, it can be observed that 'h' is much closer to the non-dominating solution c and hence, intuitively, 'h' should get higher priority over 'g' and 'j'. The parameter f^i_{rd} takes care of this fact. It is calculated as the normalized Euclidean distance of a dominated solution to its nearest non-dominated solution (if any). Note that the value of f^i_{rd} will be 0 for the non-domianted solutions, and smaller values of f^i_{rd} signifies better solutions.

• Archive Truncation and Density Parameter (d_i): MOGARD has *elitism* similar to that in SPEA2 [9]. An *archive* of fixed size is maintained, that gets updated at each iteration by non-dominated solutions of the current archive and population, followed by the best (in terms of F_is) remaining solutions till the archive is full. In case of archive overflow at any stage, the density paramater d_i is used to truncate the archive.

Algorithm 1: O_f= MOGARD(P_t,A_t,N,C)
/* P_t: pop. at iter. t, A_t: archive at iter. t, N: size of pop. and archive, C: max. no. of iter. O_f: final front */

```
1. $t = 0$, randomly initialize $P_0$, $A_0 = \Phi$.
2. $A_{t+1} \leftarrow$ non-dominated solutions from $P_t \cup A_t$
3. $\forall i, \quad i \in \{P_t \cup A_t\}$, calculate $f^i_{rd}$ and $d_i$.
   if $(i \notin (A_{t+1}))$     /* $A^j_{t+1}$ is $j^{th}$ element of $A_{t+1}$*/
       • $f^i_{rd} = \min_{j=1,\ldots,|A_{t+1}|} \left(||i - A^j_{t+1}||\right)^{1/2}$
   else
       • $f^i_{rd} = 0$
   /* for computation of $d_i$, see text and Figure 1(b). */
4. Update $A_{t+1}$
       • if $\{|A_{t+1}|\} > N$, truncate $A_{t+1}$ to size $N$ using $d_i$
       • if $\{|A_{t+1}|\} < N$, copy $\{N - |A_{t+1}|\}$ best dominated solutions from $P_t \cup A_t$ to $A_{t+1}$.
5. If $t \geq C$ or other terminating condition is fulfilled, $O_f \leftarrow A_{t+1}$ and stop.
6. Genetic Operations:
       • Density-based binary tournament selection on $A_{t+1}$ to get the mating pool.
       • Single-point crossover and bitwise mutation performed on mating pool to get $P_{t+1}$, compute objectives of $P_{t+1}$.
   $t \leftarrow t + 1$; goto (2).
```

The parameter (d_i) (which is similar to crowded-distance [4], with some modification) is computed as the distance of each solution to its immediate next neighbour summed over each of the M objectives. An example in Figure 1(b), illustrates the computation of d_i parameter. MOGARD uses *density-based binary tournament selection*, similar to *crowded-tournament selection* [4], on the archive to create the mating pool. From the mating pool the next generation population is generated using *single-point crossover* and *bit-wise mutation* operation.

3 Performance Measures

The two novel performance measures introduced in this article are give below in detail.

• **Relative Convergence Measure** This measure is introduced in this article in order to solve some problems inherent in the convergence measures Υ [4] and *purity* P_i [1], [6]. The convergence measures Υ [4] reflects the closeness of an obtained front from the true PO front. It needs the prior knowledge of the PO front (which is unavailable in most real life problems) and also involves limited sampling of the true PO front (that may also lead to misleading results). Another measure purity P_i, [1] compares solutions of several MOO algorithms only on the basis of domination (not the amount of domination). Consequently it may often fail to identify an obviously poor front. For example Figure 2 represents two non-dominated fronts obtained by algorithms c_1 and c_2, on a two objective minimization problem. In terms of $purity$, both the algorithms are equivalent. However it is evident from Figure 2(a) that the front represented by c_2 is better, if the distance between its dominated solutions to the corresponding non-dominated solutions, in the combined non-dominated front c_3 in Figure 2(b) is considered.

To resolve the aforementioned problems a new measure called relative convergence has been formulated that combines the non-dominated points obtained by all the algorithms to form *combined front* $C_f = \left\{\bigcup_{i=1}^{N_A} S_i\right\}$, where N_A algorithms are combined. Let algorithm A_i produced the solution set S_i. If the non-dominated solutions obtained from C_f be C^*_f, then the relative convergence for i^{th} algorithm, i.e., R^i_Υ, is defined as:

$$R^i_\Upsilon = \frac{1}{n_i} \sum_{\forall x_j \in S_i} \min_{\forall c_k \in C^*_f} ||x_j - c_k||, \qquad (1)$$

where $n_i = |S_i|$. This measure is free from the aforementioned problems. Figure 2(b) illustrates the method for the

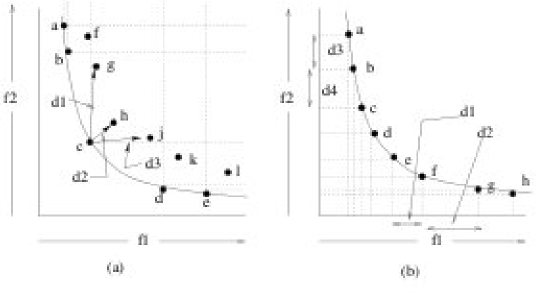


Figure 1. (a): f^i_{rd} parameter (b): diversity parameter

computation of R_Υ^i. The smaller the value of this measure, better is the performance.

• **Hypergrid Measure** In MOEA literature the performance measures to evaluate the spread and diversity of the solutions on the non-dominated set like spacing [3], minimal spacing [1] and Δ [4] are basically distance based and very often need the prior information regarding the nature of the true Patero front. Moreover, where the surface itself is discontinuous, having large gaps in between, these measures may be misled as in Figure 3. A hypergrid measure has been formulated in this article, where objective space is first partitioned into uniform sized hypergrids. The count of the number of occupied grids, N_g, gives the information about the spread of the solutions. Then the average number of solutions occupying a grid i.e., $n_{av} = \frac{N}{N_g}$, is calculated, where N is the number of solutions in the non-dominated set. Let the number of solutions occupying the i^{th} hypergrid be n_i, $\forall i,\ i \in \{1, 2 \ldots N_g\}$. The hypergrid measure hg, is then defined as:

$$hg = \sqrt{\frac{1}{N_g}\sum_{i=1}^{N_g}(n_{av} - n_i)^2}. \qquad (2)$$

The hg measure that estimates the standard deviation of the n_i values provides the information about the distribution of the solutions on the non-dominated set. Figure 3 illustrates the method for the computation of this measure. The lower value of hg whereas a higher value of N_g parameter signifies better performance. The advantage with this measure is that it is free from the aforementioned problems.

One key issue involved in the computation of hg is the number of intervals along each dimensions of the objective space. This should be properly choosen for the measure to provide relevant information. An approach, adopted in this article, is to divide each dimension by the size of the archive.

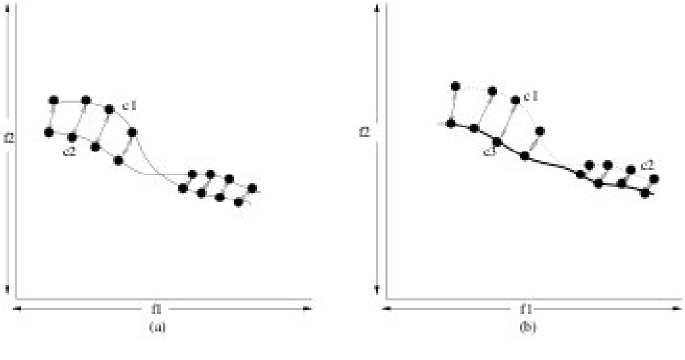
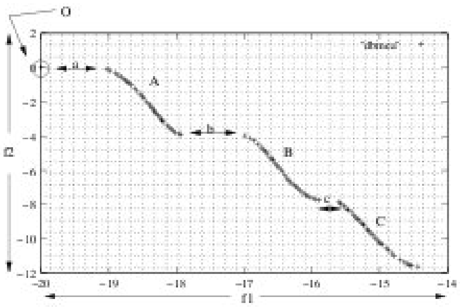


Figure 2. Relative Convergence measure calculation



* A,B,C and O are the Disconnected Pareto optimal segments
* a,b and c are the the gaps on Pareto optimal front

Figure 3. Hypergrid measure calculation

4 Simulation Study

MOGARD is compared with NSGA-II, PESA-II and SPEA2 on 11 standard test problems including 8 two-objective and 3 three-objective problems. The binary encoded implementation of the algorithms have been studied. The parameters used in these experiments are: *population size: 100*, *number of iterations: 350* (for NSGA-II, SPEA2 and MOGARD) and *archive size: 100* (for SPEA2, PESA-II and MOGARD). The parameters for PESA-II are *population size:10*, *number of iterations: 3500* and *hypergrid size* 32×32 as suggested in [2]. *Crossover-probability: 0.9*, *mutation probability: inversely proportional to the chromosome length*, *chromosome-length: 10 bits per variable*. Twenty runs of the algorithms were executed on each test problem, the mean and variance of the performance measures over these runs are reported.

• **Test Problems:** The set of test problems used in this article includes SCH1, SCH2, FON [3], ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 [8], DTLZ1, DTLZ2 and DTLZ7 [5].

Results: Considering Table 2 for R_Υ, MOGARD performed the best for $SCH1$, $SCH2$, FON, $ZDT1$, $ZDT2$, $ZDT3$, $ZDT4$, $ZDT6$ and $DLTZ1$. NSGA-II has also given best results for $SCH1$ and $SCH2$, while PESA-II does so for $DLTZ2$ and $DLTZ7$. Considering Table 1 for N_g, it can be seen that MOGARD has the best values corresponding to all the functions, except $ZDT3$, $DLTZ2$ and $DLTZ7$. Even for these the values are very close to the best ones (provided by NSGA-II for $ZDT3$ and PESA-II for $DTLZ2$ and $DTLZ7$). Similarly for hg (Table 1), MOGARD outperforms the other algorithms for all except $DLTZ2$ and $DLTZ7$ (where, as earlier, PESA-II performs the best). Again performance of MOGARD is second best. The results demonstrate the superiority of proposed algorithm for a range of test functions. Similar performance of MOGARD has been observed using other performance

Table 1. Mean (M) and Variance (Var) of R^i_Υ measure for the test problems

Algorithm	SCH1	SCH2	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DLTZ1	DLTZ2	DLTZ7
NSGA-II(M)	**0.00000**	**0.00005**	0.00123	0.00075	0.00053	0.00067	1.03725	**0.00000**	3.38103	0.54103	0.00715
(VAR)	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.03200	0.00000	12.97700	0.08884	0.00001
PESA-II(M)	0.00006	0.00021	0.01559	0.00004	0.00003	0.00113	7.88181	0.00021	6.14686	**0.00014**	**0.00066**
(VAR)	0.00000	0.00000	0.00005	0.00000	0.00000	0.00002	18.66300	0.00000	67.31000	0.00000	0.00000
SPEA2(M)	0.00225	0.00018	0.00203	0.00358	0.00508	0.00467	0.68202	**0.00000**	6.53331	0.00071	0.00666
(VAR)	0.00006	0.00000	0.00000	0.00000	0.00000	0.00002	0.39100	0.00000	51.16900	0.00000	0.00001
MOGARD(M)	**0.00000**	**0.00005**	**0.00014**	**0.00000**	**0.00000**	**0.00005**	**0.04837**	**0.00000**	**0.11494**	0.45209	0.00279
(VAR)	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.02087	0.00000	0.07975	0.16900	0.00000

Table 2. Mean (M) and Variance (Var) of N_g and hg measure for the test problems

N_g Measure											
Algorithm	SCH1	SCH2	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DLTZ1	DLTZ2	DLTZ7
NSGA-II(M)	76.750	77.800	93.900	89.200	88.650	**85.200**	85.700	87.550	90.200	91.750	93.950
(VAR)	14.408	5.221	4.411	6.484	9.397	12.800	7.379	3.839	11.011	7.250	4.115
PESA-II(M)	56.650	83.100	90.350	70.100	65.750	47.650	60.800	95.700	83.400	**99.700**	**99.500**
(VAR)	24.345	7.147	12.134	15.884	21.671	27.818	86.274	2.432	298.147	0.432	0.787
SPEA2(M)	79.400	89.050	84.550	89.100	91.350	82.100	76.850	92.250	84.800	82.250	88.000
(VAR)	19.937	3.629	8.471	7.463	14.976	14.726	59.818	5.461	17.116	14.197	10.632
MOGARD(M)	**88.050**	**90.950**	**95.400**	**95.650**	**95.650**	85.150	**93.650**	**99.900**	**98.200**	97.600	98.150
(VAR)	1.734	1.418	2.779	5.713	3.503	21.187	16.766	0.095	2.168	3.621	1.713
hg Measure											
NSGA-II(M)	0.482	0.465	0.242	0.325	0.329	**0.379**	0.392	0.351	0.316	0.286	0.242
(VAR)	0.002	0.001	0.002	0.002	0.002	0.002	0.009	0.001	0.003	0.002	0.002
PESA-II(M)	0.848	0.409	0.316	0.656	0.671	1.418	0.868	0.204	0.360	**0.024**	**0.033**
(VAR)	0.006	0.001	0.006	0.028	0.086	0.113	0.001	0.013	0.007	0.003	0.004
SPEA2(M)	0.520	0.327	0.396	0.334	0.293	0.425	0.568	0.274	0.434	0.473	0.352
(VAR)	0.010	0.001	0.002	0.002	0.007	0.003	0.044	0.002	0.008	0.005	0.003
MOGARD(M)	**0.341**	**0.299**	**0.211**	**0.196**	**0.198**	**0.379**	**0.241**	**0.010**	**0.119**	0.133	0.121
(VAR)	0.000	0.000	0.002	0.005	0.003	0.003	0.004	0.001	0.004	0.007	0.004

measures, but due to the lack of the space those results have been omitted.

5 Constraint Handling

Constraint handling is a requirement in many real-life problems. In [4] the authors have introduced the notion of *constrained domination*, where an infeasible solution always gets eliminated when compared to a feasible one, irrespective of the goodness of the objective value. Therefore a poor feasible solution will be preferred as compared to a very good solution which is infeasible, although it may be very close to the feasible region. It may also be the case that the path to the best solutions lies through the infeasible space, whence this approach may be at a disadvantage.

In the present article a penalty based approach to constraint handling is incorporated that overcomes this problem to a large extent. This approach is motivated by the suggestions in [7] for single objective GAs. The Constrained MOO problem can be defined as

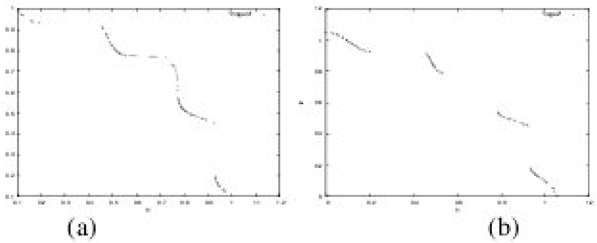
- Minimize : $f(x) = [f_i(x), i = 1, \dots, M]$,
- Subject to: $g_j(x) \geq 0, \quad j = 1, 2, \dots, J$.

here M is the number of objectives, $f_i(x)$ is the i^{th} objective function and $g_j(x)$ is the j^{th} inequality constraint. If δ_j is the constraint violation corresponding to j^{th} inequality constraint then the penalty function $p(\mathbf{x})$, is defined as

$$p(\mathbf{x}) = n_c \times \max_j \{\delta_j\}, \forall\, [j = 1, 2, \dots, J], \qquad (3)$$

where n_c is given by $n_c = |\{j : g_j(\mathbf{x}) < 0\}|$. Thus the fitness of $\mathbf{x}$ corresponding to each objective now gets updated as

$$f(\mathbf{x}) = [f_i(\mathbf{x}) + p(\mathbf{x}), i = 1, \dots, M]. \qquad (4)$$



(a) (b)

Figure 4. Final PO-fronts of (a): MOGARD and (b): NSGA-II on tnk

Table 3. Results on Constraint test problems

P_i Measure					
Algorithm	constr	tnk	srn	ctp1	ctp2
NSGA-II	0.20	0.10	0.43	0.81	0.23
MOGARD	0.90	0.80	0.83	0.96	0.49
S_m Measure					
NSGA-II	0.092	0.043	0.011	0.009	0.022
MOGARD	0.067	0.041	0.009	0.007	0.021

This approach gives more priority to the feasible solutions, while degrading the infeasible solutions but not totally disqualifying them. In most complex constrained search spaces, with disconnected feasible regions, the infeasible solutions may also play a vital role. MOGARD and NSGA-II algorthms have been compared for their relative performance on $CONSTR, SRN, TNK$ [4], $ctp1$ and $ctp2$ [3] constrained test problems.

The comparative results of MOGARD with NSGA-II, in terms of two existing performance measures namely purity (P_i) and minimal spacing (S_m), are provided in Table 3. A higher value of purity measure and a lower value of S_m signifies better performance. Figure 4 representing the final non-dominated fronts obtained by MOGARD and NSGA-II for tnk problem, clearly shows better performance of MOGARD, where NSGA-II is found to fail to capture the true non-dominated front fully. Out of the five disconnected front segments NSGA-II obtained four segments, while MOGARD is successful in obtaining all the five segments. Results with other performance measures also demonstrated similar performance.

6 Discussion and Conclusions

An MOEA called MOGARD has been described that has a novel relative distance parameter, diversity method and a constraint handling concept integrated in MOO. Two novel performance measures namely, relative convergence and hypergrid measures have been proposed. The performance of MOGARD, based on these concepts, is compared with that of other MOEAs (NSGA-II, PESA-II and SPEA2) on several test problems with different characteristics. It is found that, in general, MOGARD outperforms the other algorithms in terms of the new measures (as also existings whose results could not be included) for most of the test problems. Constraint handling in MOO problems has been investigated using a penalty based concept. The comparative performance evaluation of constrained-MOGARD is done with NSGA-II using some constrained test problems. Again, the performance of MOGARD is found to be significantly better than that of NSGA-II.

References

[1] S. Bandyopadhyay, S. K. Pal, and B. Aruna. Multi-Objective GAs, Quantitative Indices, and Pattern Classification. *IEEE Trans. on Sys., Man, and Cyber.-Part B: Cybernetics*, 34:2088–2099, 2004.

[2] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In *Proc. of the Genetic and Evol. Comp. Conf. (GECCO-2001)*, pages 283–290, 2001.

[3] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, New York, 2001.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Trans. On Evol. Comp.*, 6(2):182–197, April 2002.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report TIK-Technical Report No. 112, Institut fur Technische Informatik und Kommunikationsnetze,, July 2001.

[6] H. Ishibuchi and T. Murata. A Multi-Objective Genetic Local Search Algorithm and its Application to Flowshop Scheduling. *IEEE Trans. on Sys., Man, and Cyber.-Part C: Applications and Reviews*, 28:392–403, 1998.

[7] J. T. Richardson, M. R. Palmer, G. Liepins, and M. Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. Schaffer, editor, *Proc. of the First Int'l Conf. on Genetic Algorithms*, pages 191–197, 1989.

[8] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comp.*, 8(2):125–148, 2000.

[9] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Proc. of the EUROGEN2001 Conf., Athens, Greece, September 19-21, 2001*, pages 95–100, 2002.