Performance Assessment of the Hybrid Archive-based Micro Genetic Algorithm (AMGA) on the CEC09 Test Problems

Santosh Tiwari[1], Georges Fadel[1], Patrick Koch[2], and Kalyanmoy Deb[3]
[1]Department of Mechanical Engineering, Clemson University, SC, USA
[2]Dassault Systemes, Simulia Corp., Cary, NC, USA
[3]Indian Institute of Technology, Kanpur, UP, India

Abstract— **In this paper, the performance assessment of the hybrid Archive-based Micro Genetic Algorithm (AMGA) on a set of bound-constrained synthetic test problems is reported. The hybrid AMGA proposed in this paper is a combination of a classical gradient based single-objective optimization algorithm and an evolutionary multi-objective optimization algorithm. The gradient based optimizer is used for a fast local search and is a variant of the sequential quadratic programming method. The Matlab implementation of the SQP (provided by the fmincon optimization function) is used in this paper. The evolutionary multi-objective optimization algorithm AMGA is used as the global optimizer. A scalarization scheme based on the weighted objectives is proposed which is designed to facilitate the simultaneous improvement of all the objectives. The scalarization scheme proposed in this paper also utilizes reference points as constraints to enable the algorithm to solve non-convex optimization problems. The gradient based optimizer is used as the mutation operator of the evolutionary algorithm and a suitable scheme to switch between the genetic mutation and the gradient based mutation is proposed. The hybrid AMGA is designed to balance local versus global search strategies so as to obtain a set of diverse non-dominated solutions as quickly as possible. The simulation results of the hybrid AMGA are reported on the bound-constrained test problems described in the CEC09 benchmark suite.**

I. INTRODUCTION

Multi-objective optimization has become mainstream in recent years and many algorithms to solve multi-objective optimization problems have been suggested. The use of multi-objective optimization in the industry has been accelerated by the availability of faster processing units and computational analysis tools for various engineering problems and disciplines. The ever increasing popularity of multi-objective optimization in industry and the need for faster optimization algorithms has led to the development of several multi-objective optimization algorithms (MOEAs) in the recent past [1], [2], [3], [4], [5], [6], [7], [8]. Whenever, a new multi-objective optimization algorithm is proposed, a performance comparison of the proposed algorithm with the current state-of-the-art optimization algorithms is performed, and generally the new algorithm is shown to be faster than the existing algorithms on a set of carefully chosen synthetic benchmark problems. A special session on the performance assessment of different MOEAs thus provides an opportunity for a fair and unbiased comparison of different multi-objective optimizers. In this report, the performance of the hybrid Archive-based Micro Genetic Algorithm (AMGA) on the unconstrained test problems described in the CEC09 technical report [9] is reported.

The AMGA [10] is a constrained multi-objective evolutionary optimization algorithm. It is a generational genetic algorithm since during a particular iteration (generation), only solutions created before that iteration take part in the selection process. AMGA uses genetic variation operators such as crossover and mutation to create new solutions. For the purpose of selection, AMGA uses a two tier fitness assignment mechanism; the primary fitness is the rank which is based on the domination level and the secondary fitness is based on the diversity of the solutions in the entire population. This is in contrast to NSGA-II, where the diversity is computed only among the solutions belonging to the same rank. The AMGA generates a very small number of new solutions at every iteration and can therefore be classified as a micro genetic algorithm. Generation of a very small number of solutions at every iteration helps in reducing the number of function evaluations by minimizing exploration of less promising search regions and directions. The AMGA maintains an external archive of *good* solutions obtained. Use of the external archive helps AMGA in reporting a large number of non-dominated solutions at the end of the simulation and also provides information about its search history which is exploited by the algorithm during the selection operation. At every iteration, the parent population is created from the archive and binary tournament selection is performed on the parent population to create the mating population. The offspring population is created from the mating pool, and is used to update the archive. The size of the archive determines the computational complexity of the AMGA, however for computationally expensive optimization problems, the actual time taken by the algorithm is negligible as compared to the time taken by the analysis routines. The design of the algorithm is independent of the encoding of the variables and thus the proposed algorithm can work with almost any kind of encoding (so long as suitable genetic variation operators are provided to the algorithm). The algorithm uses the concept of Pareto ranking borrowed from NSGA-II [3] and includes improved diversity computation and preservation techniques. The diversity measure is based on efficient nearest neighbor search [11] and modified crowding distance formulation [10]. A more detailed description of the AMGA can be found in the original study [10]. In this paper, only the modifications done to AMGA to couple a gradient based optimizer are discussed.

The gradient-based local optimizer used with AMGA is the Sequential Quadratic Programming (SQP) algorithm [12]. SQP is one of the most popular and robust algorithms for constrained nonlinear single-objective optimization. Although, the SQP is capable of solving constrained test problems, we report the simulation results only on the unconstrained problems because the scalarization scheme proposed in this paper works only with the unconstrained problems. The SQP algorithm attempts to approximate the objective function using a quadratic model and the constraint functions using a linear model of the optimization variables. The SQP algorithm requires the computation of the Hessian of the objective vector which is approximated using the BFGS method [12]. SQP has excellent local convergence properties and is shown to be faster than most other gradient based optimizers on a large set of test problems [13]. The application of SQP for multi-objective optimization requires scalarization of the objective vector. In order to ensure that the SQP works with non-convex problems, artificial constraints are added to ensure that improvement in all the objectives is observed. The additional constraints ensure that the solution obtained using SQP always dominates the initial (starting) solution. In the worst case (when the starting solutions happens to be the local optimum), no improvement is observed. It should be noted that the addition of SQP to the AMGA does not directly affect its global search capability. It however speeds up the local search process thereby allowing more function evaluations to be used for the global search.

It should be noted that hybridizing evolutionary algorithms with mathematical programming techniques has been attempted in the past [14], [15], [16], [17]. Hybrid evolutionary algorithms are also often referred to as memetic algorithms owing to their use of local search techniques which are traditionally faster than a typical evolutionary algorithm. The novel concept proposed in this paper is the use of a starting reference point to enable the local optimizer to attempt to simultaneously improve all the objectives and also work with non-convex problems. The reference point is used to formulate the additional constraints such that only the region dominated by the starting point is feasible. The scalarization scheme employed in this paper and the modification done to the AMGA to incorporate the SQP algorithm are discussed in the next section.

II. Hybridization of the AMGA

A. The Scalarization Scheme

The scalarization scheme to convert the multi-objective optimization problem to a single-objective optimization problem is discussed first. The scalarization scheme also includes the reference-point method to ensure its applicability to non-convex problems. The specific way in which the reference-point method is used here limits its applicability to unconstrained problems or constrained problems for which the starting (initial) solution is feasible. The general problem statement for the unconstrained (unconstrained in this case refers to bound-constrained) multi-objective optimization is given by Equation 1.

$$\begin{array}{ll} \text{Minimize} & \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right), \\ \text{Subject to} & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n. \end{array} \tag{1}$$

Let the starting (initial) solution for the optimization be $\mathbf{x}_{initial}$. Let the value of the objective vector at the starting point $\mathbf{x}_{initial}$ be $\mathbf{f}_{initial}$. We want to use SQP with the initial guess solution as $\mathbf{x}_{initial}$, and obtain a new solution $\mathbf{x}_{final}$ such that the corresponding objective vector $\mathbf{f}_{final}$ dominates $\mathbf{f}_{initial}$. We use $\mathbf{f}_{initial}$ as the reference point in the scalarization scheme. The corresponding single-objective optimization problem is given by Equation 2.

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{M} f_i \\ \text{Subject to} & f_j(\mathbf{x}) \leq (\mathbf{f}_{initial})_j, \quad j = 1, 2, \ldots, M, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n. \end{array} \tag{2}$$

The feasible search region for the scalarized test problem is shown in Figure 1. One inequality constraint is added for each objective function. The additional of inequality constraints ensures that the final obtained solution dominates the starting solution when using SQP. If the constraints are not introduced, and the problem is non-convex, application of SQP may result in a solution which is non-dominated with respect to the starting solution. This scenario is not desirable, since it would not speed-up the convergence towards the Pareto-optimal frontier. It should be noted that there exist several other scalarization schemes such as Epsilon-constraint method [18] and Normal-constraint method [18]. These scalarization schemes are designed to obtain a uniform distribution of points on the Pareto-optimal front. In this case however, the uniformity of distribution of the points on the Pareto-optimal front is taken care-of by the global optimizer AMGA. The SQP is used to speed-up the search process and obtain an improvement in the objective function value as quickly as possible. It is also possible to use other weighting schemes such as Tchebycheff metric [19] for scalarizing the optimization problem. No noticeable change in performance was observed by using different weighting schemes. For the scalarization scheme to work efficiently, it is important that the objectives be normalized before computing the weighted objective vector. The normalization of the objectives is done by linearly scaling the objectives for every solution in the current population in the range zero to one. The minimum and maximum value of every objective function in the current population is used for the scaling operation.

B. Incorporating SQP into AMGA

The application of SQP to the scalarized single-objective optimization problem described in the previous subsection is akin to starting from an initial solution and obtaining a final solution such that it is a local optimum. It should be noted that if the starting point does not correspond to a local optimum, then there always exists a feasible search direction

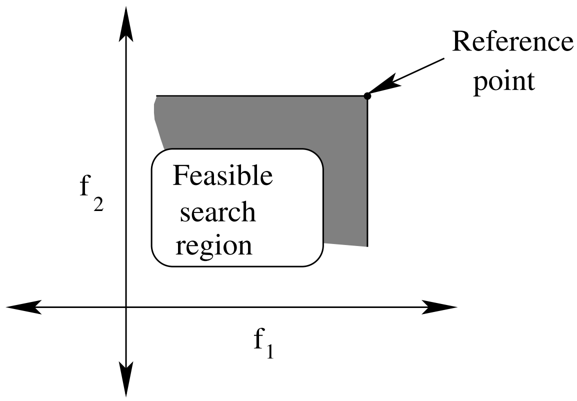

Fig. 1. Feasible search space

which will improve all the objectives simultaneously. This feasible search direction will guide the optimization process towards the nearest local optimum. Further, SQP is one of the fastest known methods [13] to find the local optimum of constrained nonlinear single-objective optimization problems. Thus, if this step is incorporated as the mutation step of an evolutionary optimization algorithm, mutating the offspring solution will drive that solution towards to nearest local optimum. A potential drawback of this approach is that every time a solution is mutated, it will hit its nearest local optimum. Also, all the solutions in the offspring population will get accumulated at their respective local optima. For highly multi-modal problems, it is desired that a disruptive genetic mutation operator is also incorporated such that a robust search for the global optimum is facilitated. Further, once the SQP is applied to a few selected solutions from the archive, a set of good solutions are obtained. It is then desired to explore the search space around those solutions and improve the diversity near the obtained solutions. Based on empirical investigation and experimentation with different schemes, it was observed that switching between the genetic polynomial mutation [20] operator and the SQP algorithm at regular intervals (after every few generations) resulted in a search strategy which balanced diversity of the obtained solutions, the convergence rate, and the global search capability of the optimization algorithm. The generation scheme of the AMGA was modified to incorporate the SQP algorithm.

The optimization process starts with an initial population generated randomly using Latin Hypercube [21] sampling. Selection, simulated binary crossover [22], and polynomial mutation [20] are then performed for a fixed number of generations. Selection operation creates the parent population from the archive and the genetic variation operators create the offspring population from the parent population. At every generation, the offspring population is used to update the archive. After a fixed number of generations has been completed, the genetic mutation operator is replaced by the SQP method and the multi-objective problem is scalarized. The maximum number of function evaluations allowed for SQP is also specified. This modification is applied for a single generation. SQP is then applied to every solution in the offspring population. Once the iteration of SQP has finished and new solutions are obtained, the archive is updated using the obtained solutions. The mutation operator is switched back to the polynomial mutation. Again, a fixed number of generations is performed, and the process is repeated until the allowed number of function evaluations is exhausted. If at any instant, the limit on number of function evaluations is reached, the optimization process is terminated, and the non-dominated solutions in the archive are reported as the final obtained solutions. The total number of function evaluations includes the function evaluations performed by AMGA and the SQP. The pseudo-code of the hybridized AMGA is as follows.

```
The hybrid AMGA pseudo-code:
1  Begin
2    Initialize optimization
       parameters.
3    Generate initial population.
4    Evaluate initial population.
5    Update the archive (using the
       initial population).
6    repeat
7      Create parent population from
         the archive.
8      Create mating pool from the
         parent population.
9      If generation count a multiple of
         switch frequency
10       Create offspring population from
         the mating pool by crossover.
11       Mutate the offspring population.
12       Evaluate offspring population.
13     Else
14       Use SQP on every individual in
           the offspring population.
15       Update the number of function
           evaluations.
16     Update the archive (using the
         off-spring population).
17   until (termination)
18   Report desired number of solutions
       from the archive.
19 End
```

From the above pseudo-code, it is evident that the basic flowchart of the AMGA remains unchanged except an IF statement at step 9. Hence, the addition of SQP does not change the generational scheme employed by the AMGA. From an algorithm design perspective, any local search strategy can be used in step 14 of AMGA so long as it facilitates relatively faster convergence towards to nearest local optimum.

III. Simulation Results

The simulation results of the hybrid AMGA applied to bound-constrained multi-objective test problems are presented in this section. The following optimization tuning parameters are used to report the simulation results.

- Size of the initial population for 2 objectives = 100
- Size of the initial population for 3 objectives = 150
- Size of the parent population for 2 objectives = 32
- Size of the parent population for 3 objectives = 24
- Size of the archive = size of the initial population
- Number of solutions reported at the end of the simulation = size of the initial population
- Number of function evaluations = T = 300,000
- Probability of crossover = 1.0
- Probability of mutation = $1/N$, where N is the number of optimization variables
- Distribution index for crossover = 0.5
- Distribution index for mutation = 0.5
- Number of generations for switching the mutation operator = $T/100$
- Number of function evaluations allowed for each SQP iteration = $T/100$

The typical value of the crossover and mutation indices used with a genetic algorithm is in the range 5 to 50. The smaller the value of indices, the larger is the perturbation in the design variables. The indices are chosen so as to balance the disruptiveness (required for fast and robust search) of the genetic variation operators whilst attempting to find a fine-grained value for the objective functions. Since SQP is used with the AMGA, a fine-grained (accurate) value for the objective functions is ensured. Hence, in the present case, a small value (0.5) for the crossover and mutation index is used because it facilitates robust search and increases the resilience to premature convergence. It thus reduces the probability of getting stuck at a local optimum. The other simulation related parameters are as follows.

- Operating system: Windows XP Professional
- Programming language for AMGA: JAVA
- Runtime for SQP: Matlab JVM
- CPU: Core 2 Quad 2.4 GHz
- RAM: 4 GB DDR2 1066 MHz
- Execution time for single simulation for 2-objective test problems: 3 minutes approx.
- Execution time for single simulation for 3-objective test problems: 6 minutes approx.

The execution time for the problems with three objectives is higher because of the larger size of the archive. The CEC09 test problem suite also includes three 5-objective test problems. The simulation with 5-objective test problems could not be performed because of software related issues encountered when linking Matlab, C++, and JAVA. The performance indicator used to quantify the quality of the obtained results is the IGD metric [9]. The IGD metric measures "how well is the Pareto-optimal front represented by the obtained solution set". To quantify this information, a large set of evenly spaced points on the Pareto-optimal front is generated. Let the size of this set be H. The minimum Euclidean distance of each point in this set from the obtained solution set is computed. Let this distance be l_i for the i^{th} element of the Pareto-optimal set. Then the IGD metric is given by

$$\text{IGD metric} = \frac{1}{H}\sum_{i=1}^{H} l_i \quad (3)$$

The IGD metric for the case of two objectives is pictorially depicted in Figure 2. The IGD metric measures both the convergence and the spread of the obtained solutions. Smaller the value of the IGD metric, better is the obtained solution set.

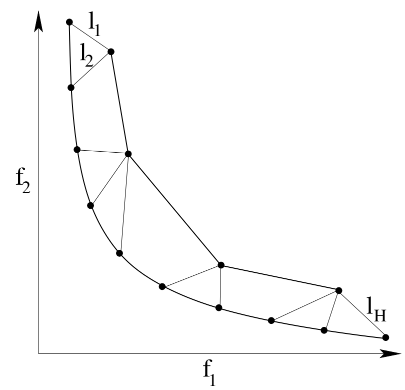

Fig. 2. IGD Metric

30 random simulations are performed for each problem and the minimum (Min), maximum (Max), mean, and standard deviation (Std) of the IGD metric are reported in Table I.

TABLE I

The IGD metric

Problem	Min	Max	Mean	Std
UF1	0.021023	0.059289	0.035886	0.010252
UF2	0.011635	0.024160	0.016236	0.003167
UF3	0.037659	0.089363	0.069980	0.013954
UF4	0.037688	0.044606	0.040621	0.001750
UF5	0.070599	0.134627	0.094057	0.012055
UF6	0.045115	0.230019	0.129425	0.056588
UF7	0.013147	0.247734	0.057076	0.065309
UF8	0.139957	0.206937	0.171251	0.017224
UF9	0.112624	0.265932	0.188610	0.042137
UF10	0.201427	0.547349	0.324186	0.095718

The plots of the Pareto-optimal front for all the test problems are shown in Figures 3 to 12.

The plots of the mean of the IGD metric with the number of function evaluations for all the test problems are shown in Figures 13 to 22.

IV. Discussion, Conclusion, and Future Work

As is evident from Table I, the hybrid AMGA is able to find an approximate solution set near the the global Pareto-optimal front for most problems. In most test problems,

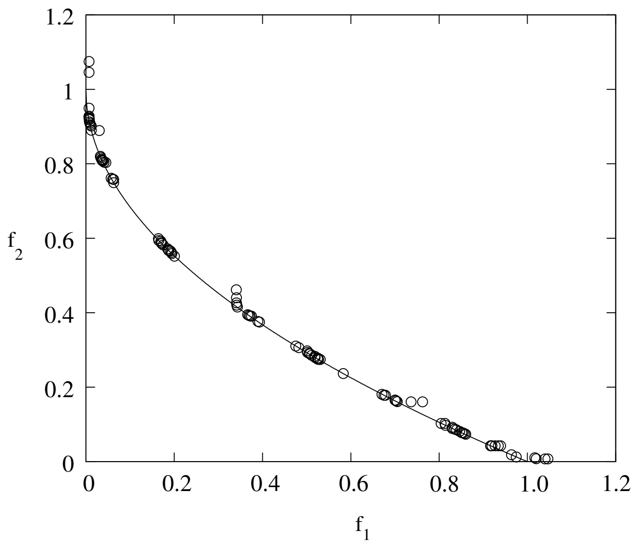

Fig. 3. Pareto front for problem UF1

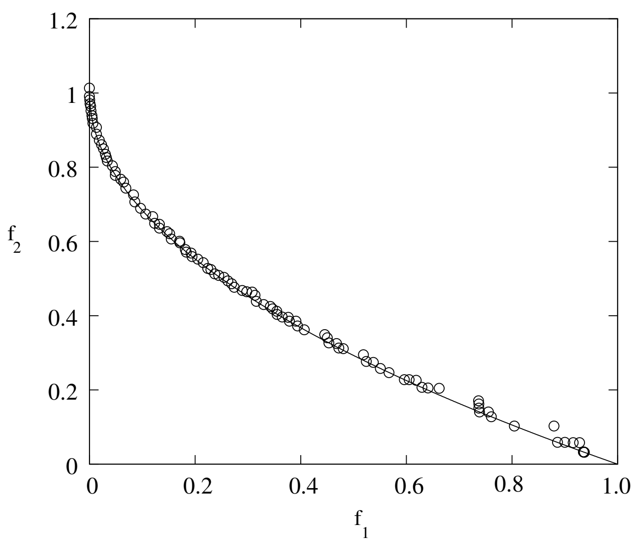

Fig. 4. Pareto front for problem UF2

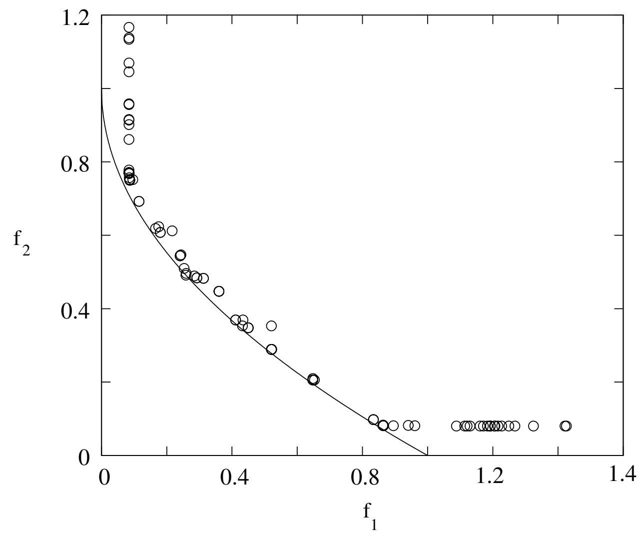

Fig. 5. Pareto front for problem UF3

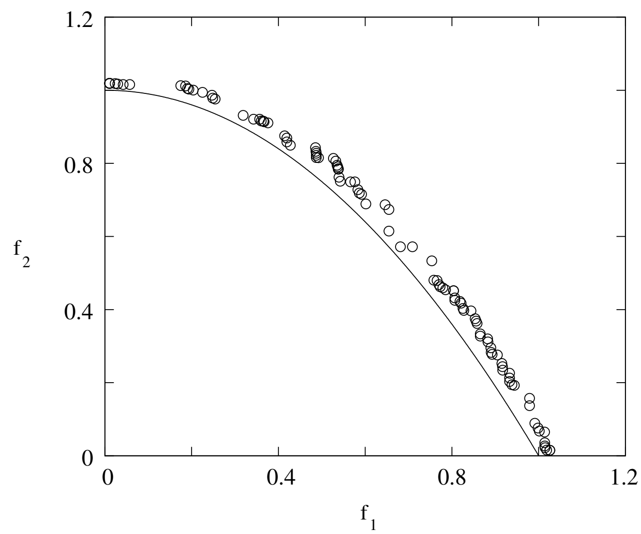

Fig. 6. Pareto front for problem UF4

global convergence is obtained but the complete Pareto-optimal frontier is not discovered by the hybrid AMGA. The primary cause of this behavior is the objective function profile which is multi-modal near the global Pareto-optimal frontier, and a slight perturbation in the optimization variables causes the solutions to become dominated. Also, the phenomenon of genetic drift causes the population to follow the good solutions which get discovered early in the search process. This genetic drift results in the clustering of the solutions around these points.

With the hybrid AMGA proposed in this paper, it is not possible to get the IGD metric at every function evaluation. It is therefore not possible to get the mean of the IGD metric at every function evaluation. Further, due to the inclusion of the SQP algorithm in the AMGA, the number of function evaluations exhausted at any generation is different, and therefore the IGD metric cannot be computed at the same number of function evaluations for different simulations starting with different random seeds. Hence, cubic spline interpolation was used to generate the plots of the mean value of the IGD metric. The cubic spline interpolates all the data points, and does not have overshoots. It thus gives an accurate interpolation of the data points. It is also evident from the convergence plots that the IGD metric does not always monotonically decrease with the increase in the number of function evaluations. This is due to way the AMGA algorithm works. The parent population is created from the archive using only the diversity in the variable space. Hence, instead of picking the best solutions, it picks

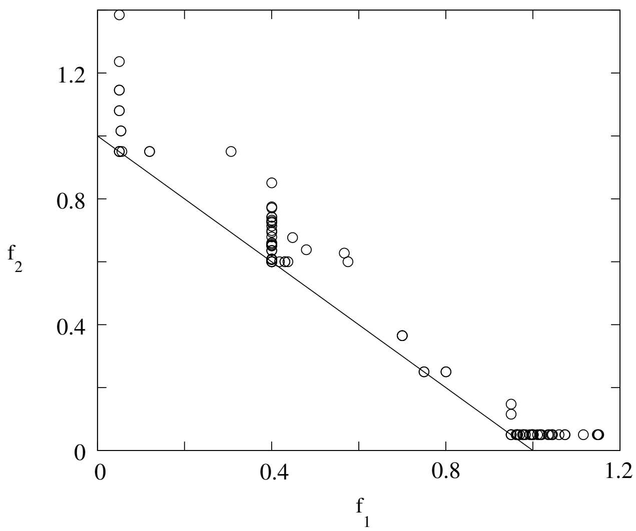

Fig. 7. Pareto front for problem UF5

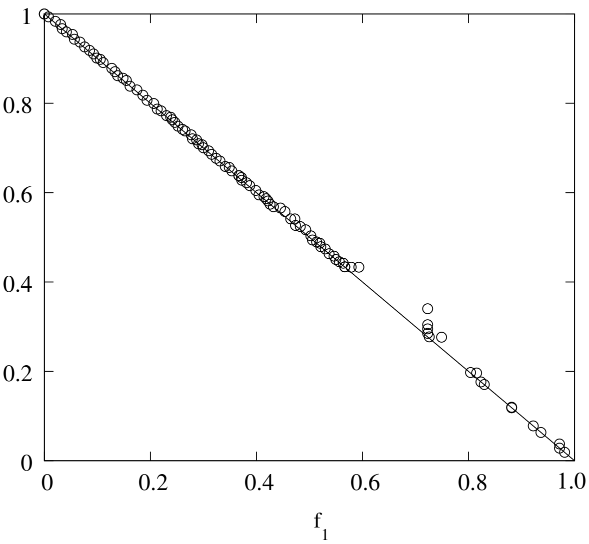

Fig. 9. Pareto front for problem UF7

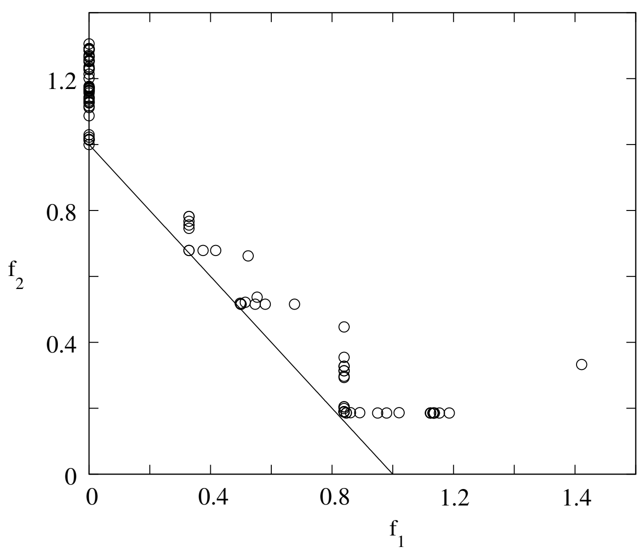

Fig. 8. Pareto front for problem UF6

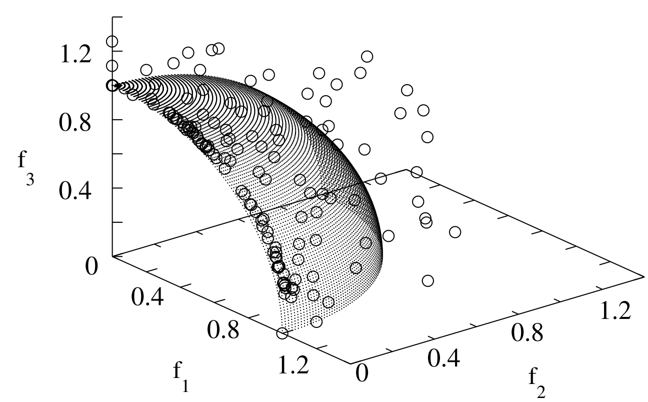

Fig. 10. Pareto front for problem UF8

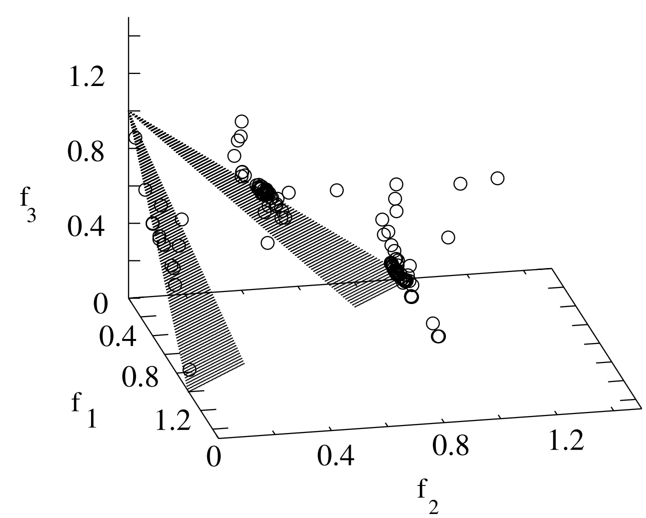

Fig. 11. Pareto front for problem UF9

the most diverse solutions. Such a strategy helps in the case of multi-modal problems, but does not always improve the convergence measure.

The plot of the mean of the IGD metric for the case of UF5 (Figure 17) shows that the smallest (best) value of the mean of the IGD metric is 0.12, however in the Table I, the mean value of the IGD metric is 0.094. This apparent discrepancy is due to the fact that, the smallest value of the IGD metric need not be obtained at the end of the simulation, nor is its behavior monotonic with the number of function evaluations for the case of hybrid AMGA. For every simulation, the best value of the IGD metric obtained at any stage of the simulation is reported as the IGD metric for that simulation. The mean value in the Table I represents the mean of the reported IGD metrics for each simulation.

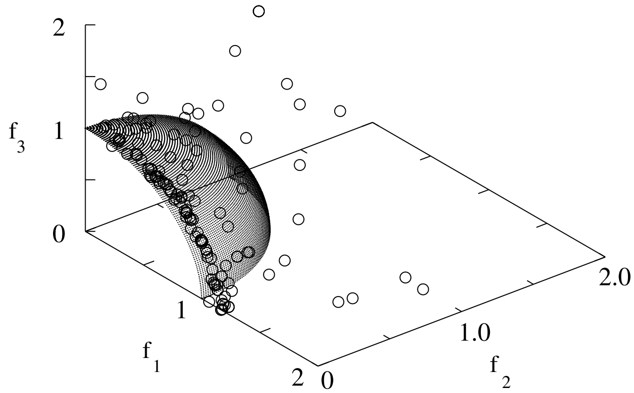

Fig. 12. Pareto front for problem UF10

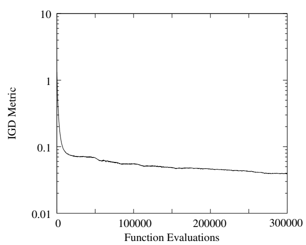

Fig. 13. Mean of the IGD Metric for problem UF1

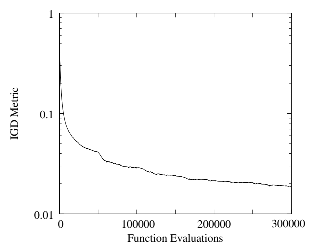

Fig. 14. Mean of the IGD Metric for problem UF2

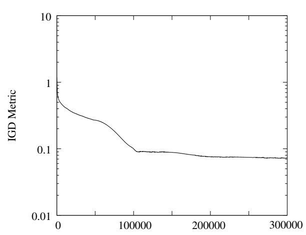

Fig. 15. Mean of the IGD Metric for problem UF3

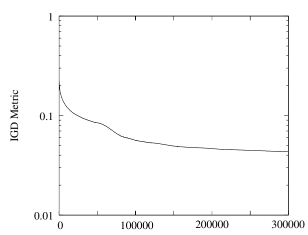

Fig. 16. Mean of the IGD Metric for problem UF4

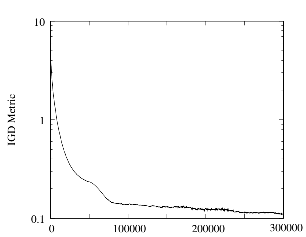

Fig. 17. Mean of the IGD Metric for problem UF5

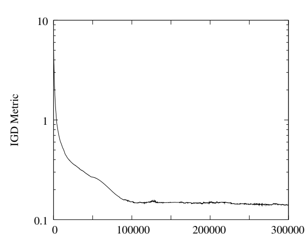

Fig. 18. Mean of the IGD Metric for problem UF6

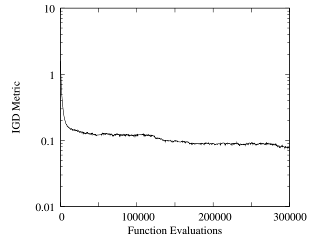

Fig. 19. Mean of the IGD Metric for problem UF7

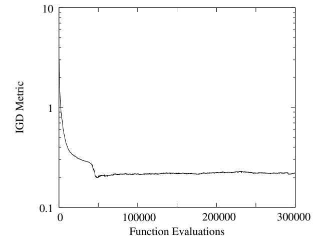

Fig. 20. Mean of the IGD Metric for problem UF8

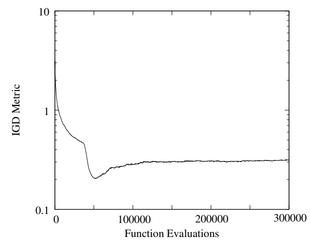

Fig. 21. Mean of the IGD Metric for problem UF9

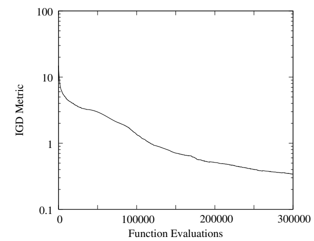

Fig. 22. Mean of the IGD Metric for problem UF10

For generating the plots of the mean of the IGD metric, the mean is computed by using the IGD metric at that function evaluation. Hence, the apparent discrepancy is due to the fact that different simulations starting with different random seeds do not achieve their respective best value of the IGD metric simultaneously.

It was also observed in many simulations, that a reasonably good convergence was obtained in under 100,000 function evaluations. And, the additional 200,000 function evaluations did not result in any significant improvement in the obtained solution set. This behavior is evident in test problems UF1, UF3, UF5, UF6, UF7, UF8, and UF9. Also, faster reduction in the value of the IGD metric can be observed at certain stages for different test problems. This behavior can be attributed to the SQP algorithm. The SQP in certain cases results in significantly improved objective function values which causes a sudden decrease in the value of the IGD metric.

Overall, the proposed hybrid AMGA performs reasonably well on the test problems used for this study. The inclusion of SQP speeds-up the search process and also helps in obtaining a fine-grained value for the objective functions. Some limitations of the proposed hybrid AMGA also became evident during the simulation process which are mentioned below.

- In most cases, the global Pareto-optimal front was found, and the extreme solutions were discovered, however the diversity of the obtained solutions was not good. Better diversity preservation operators are needed to approximate and represent the entire Pareto-optimal front.
- The scalarization scheme proposed to convert the multi-objective problem to a single-objective problem cannot

handle constrained test problems.

- In highly multi-modal problems, the search process often gets stuck in a locally optimal basin, and further improvement is not observed.

These limitations provide an opportunity for the further improvement of the proposed hybrid AMGA which shall be the focus of the future research.

The hybridized AMGA algorithm and related resources can be downloaded from http://people.clemson.edu/∼stiwari/amga.html

References

[1] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Proceedings of the EUROGEN2001 Conference*, 2001, pp. 95–100.

[2] D. W. Corne, J. D. Knowles, and M. J. Oates, "The pareto envelope-based selection algorithm for multi-objective optimization," in *Parallel Problem Solving from Nature*. Berlin: Springer, 2000, pp. 839–848.

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] S. Watanabe, T. Hiroyasu, and M. Miki, "NCGA: Neighborhood cultivation genetic algorithm for multi-objective optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2002*, 2002, pp. 458–465.

[5] K. Deb and S. Tiwari, "Omni-optimizer, a generic evolutionary algorithm for single and multi-objective optimization," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1062–1087, 2008.

[6] H. Eskandari, C. D. Geiger, and G. B. Lamont, "Fastpga: A dynamic population sizing approach for solving expensive multiobjective optimization problems," in *Evolutionary Multiobjective Optimization Conference EMO-2007*. LNCS 4403 Springer-Verlag Berlin Heidelberg, 2007, pp. 141–155.

[7] L. Marti, J. Garcia, A. Berlanga, and J. M. Molina, "Introducing moneda: Scalable multiobjective optimization with a neural estimation of distribution algorithm," in *Genetic and Evolutionary Computation Conference*. ACM New York, NY, USA, 2008, pp. 689–696.

[8] O. Schuetze, G. Sanchez, and C. A. C. Coello, "A new memetic strategy for the numerical treatment of multi-objective optimization problems," in *Genetic and Evolutionary Computation Conference*. ACM New York, NY, USA, 2008, pp. 705–712.

[9] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multi-objective optimization test instances for the cec 2009 special session and competition," IEEE Congress on Evolutionary Computation, Tech. Rep., 2008.

[10] S. Tiwari, G. Fadel, P. Koch, and K. Deb, "Amga: An archive-based micro genetic algorithm for multi-objective optimization," in *Genetic and Evolutionary Computation Conference*. ACM New York, NY, USA, 2008, pp. 729–736.

[11] M. Soleymani and S. Morgera, "An efficient nearest neighbor search," *IEEE Transaction on Communications*, vol. 35, no. 6, pp. 677–679, 1987.

[12] K. Schittkowski, "Nlpql: A fortran subroutine solving constrained nonlinear programming problems," *Annals of Operations Research*, vol. 5, no. 4, pp. 485–500, 1986.

[13] ——, "Test problems for nonlinear programming - user's guide," University of Bayreuth, Tech. Rep., 2002.

[14] P. A. N. Bosman and E. D. Jong, "Exploiting gradient information in numerical multi-objective optimization," in *Genetic and Evolutionary Computation Conference*. ACM New York, NY, USA, 2005, pp. 755–762.

[15] X. Hu, Z. Huang, and Z. Wang, "Hybridization of the multi-objective evolutionary algorithms and the gradient-based algorithms," in *IEEE Congress on Evolutionary Computation*, 2003, pp. 870–877.

[16] D. Sharma, A. Kumar, K. Deb, and K. Sindhya, "Hybridization of SBX based NSGA-II and sequential quadratic programming for solving multi-objective optimization problems," in *IEEE Congress on Evolutionary Computation*, 2007, pp. 3003–3010.

[17] S. Z. Martinez and C. A. C. Coello, "Hybridizing an evolutionary algorithm with mathematical programming techniques for multi-objective optimization," in *Genetic and Evolutionary Computation Conference*. ACM New York, NY, USA, 2008, pp. 769–770.

[18] A. Messac and C. A. Mattson, "Normal constraint method with guarantee of even representation of complete pareto frontier," in press: AIAA Journal.

[19] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

[20] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[21] W. L. Loh, "On latin hypercube sampling," *Annals of Statistics*, vol. 33, no. 6, pp. 2058–2080, 2005.

[22] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.