Local Search Based Evolutionary Multi-Objective Optimization Algorithm for Constrained and Unconstrained Problems

Karthik Sindhya, Ankur Sinha, Kalyanmoy Deb and Kaisa Miettinen

***Abstract*— Evolutionary multi-objective optimization algorithms are commonly used to obtain a set of non-dominated solutions for over a decade. Recently, a lot of emphasis have been laid on hybridizing evolutionary algorithms with MCDM and mathematical programming algorithms to yield a computationally efficient and convergent procedure. In this paper, we test an augmented local search based EMO procedure rigorously on a test suite of constrained and unconstrained multi-objective optimization problems. The success of our approach on most of the test problems not only provides confidence but also stresses the importance of hybrid evolutionary algorithms in solving multi-objective optimization problems.**

I. Introduction

Evolutionary multi-objective optimization (EMO) algorithms are playing a dominant role in solving problems with multiple conflicting objectives and obtaining a set of non-dominated solutions which are close to the Pareto optimal front. They have a number of advantages such as, obtaining a set of non-dominated solutions in a single run, easy handling of problems with local Pareto fronts and discrete nature due to their population approach and flexible recombination operators [1]. Despite these advantages, EMO algorithms are often criticized for their lack of convergence proofs. Besides, in order to have a better diversity among non-dominated solutions, a Pareto optimal solution may be sacrificed to accept a non-Pareto optimal solution. This causes fluctuations, i.e. convergence to the Pareto optimal front followed by departure of some solutions out of the front [2].

In the case of solving single-objective optimization problems, the use of local search as a part of evolutionary algorithms has proved beneficial [3]. The implementation is easy as they both have the same goal to find the global optimum of a single function. But the use of local search in multi-objective scenarios is not straightforward, as local search usually deals with a single objective and it is not fair e.g to choose one particular objective function among the multiple conflicting ones. Multiple criteria decision making (MCDM) techniques are also commonly used to solve multi-objective optimization problems [9]. They constitute a collection of approaches with convergence proofs. Hence, incorporating MCDM techniques in EMO algorithms, can improve the convergence. MCDM approaches usually scalarize the multiple objectives into a single objective, which is later solved using any suitable mathematical programming technique. Thus, one way to amalgamate EMO with MCDM is to use some MCDM technique as a local search operator in EMO.

Hybridization of local search with EMO has enjoyed a lot of attention in recent past, to make EMO algorithms converge faster and accurately on to the Pareto optimal front. Hybrid EMO approaches galore in literature, such as multi-objective local search by Ishibuchi and Murata [4] and Jaszkiewicz [5], hybrid algorithms by Goel and Deb [6] etc.. For an extensive literature survey, see [7] and [8]. Based on these studies and others from the literature, we can conclude that, not much effort has been spent on borrowing more effective multiple criteria decision making (MCDM) [9] ideas for local search, as usually a naive neighborhood search procedure or a weighted sum of objectives is used. A weighted sum of objectives is known to fail in producing all Pareto optimal points when the problem is non-convex. Hence, in the previous paper [8], a more effective scalarizing function called achievement scalarizing function (ASF) was utilized (which has optimal points on Pareto optimal front only) in local search to propose a hybrid approach. Significant reduction in function evaluations on test problems derived from ZDT [1] and DTLZ [10] test suites was obtained. In this paper, we use the hybrid approach suggested in [8], with one change viz., a clustering technique is used instead of the crowding distance in NSGA-II, as the crowding distance is less effective in higher dimensions [1]. We present results obtained by tests conducted on constrained and unconstrained test problems reflecting complicated real-life problems for the CEC09 multi-objective algorithm contest.

In the remainder of this paper, we first briefly describe the ASF and local search based hybrid EMO procedure used in this study, in subsequent two sections. Thereafter, we present

Karthik Sindhya is a visiting research scholar at Department of Business Technology, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland and doctoral student at Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland (email: karthik.s.sindhya@jyu.fi)

Ankur Sinha is a research scholar at Department of Business Technology, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland (email: ankur.sinha@hse.fi)

Kalyanmoy Deb is a Finnish distinguished professor at Department of Business Technology, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland and Professor, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208016, India (email: kalyanmoy.deb@hse.fi)

Kaisa Miettinen is a professor at Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland (email: Kaisa.Miettinen@jyu.fi)

the simulation results on a number of unconstrained and constrained test problems. Conclusions are drawn at the end of the paper.

II. Achievement Scalarizing functions

We consider multi-objective optimization problems of the form:

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in \mathcal{S}, \end{array} \quad (1)$$

with $k \geqslant 2$ conflicting objective functions $f_i : \mathcal{S} \rightarrow \mathcal{R}$. We denote the vector of objective function values by $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x}))^T$ to be called an objective vector. The decision vectors $\mathbf{x} = (x_1, x_2, ..., x_k)^T$ belong to the feasible region S, which is a subset of the decision variable space R^n.

In multi-objective optimization, decision vector $\mathbf{x}^* \in \mathcal{S}$ is Pareto optimal if there does not exist another $\mathbf{x} \in \mathcal{S}$ such that $f_i(\mathbf{x}) \leqslant f_i(\mathbf{x}^*)$ for all $i = 1, 2, ..., k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one index j. An objective vector is Pareto optimal if the corresponding decision vector is Pareto optimal. A vector is $weakly\ Pareto\ optimal$ if there does not exist any other feasible vector for which all objective values are better.

The MCDM literature as said in previous section has a number of techniques for solving multi-objective optimization problems. Among them reference point methods, a class of interative procedure are commonly used [9]. A reference point is a vector formed by the desirable values for each objective function by the decision maker (DM). The DM is a person who has prior knowledge on the problem under study and can take a decision to choose the most preferred solution from the supplied Pareto optimal set. Using this reference point, a scalarizing function like ASF [11] is optimized to find a solution that best satisfies the aspirations of the DM. The ASF has many advantages: an optimal solution of an ASF is always Pareto optimal and any Pareto optimal solution can be obtained by just changing the reference point.

Assuming we have a reference point $\bar{\mathbf{z}} \in R^k$, an example of an achievement scalarizing function is given by:

$$\begin{array}{ll} \text{minimize} & \max_{i=1}^{k} [w_i(f_i(\mathbf{x}) - \bar{\mathbf{z}}_i)], \\ \text{subject to} & \mathbf{x} \in \mathcal{S}, \end{array} \quad (2)$$

where $w_i = \frac{1}{\mathbf{z}_i^{nadir} - \mathbf{z}_i^{ideal}}$ is a weight factor assigned to each objective function f_i and is utilized for normalizing each of the objective functions. The nadir and ideal vectors are represented as $\mathbf{z}_i^{nadir}$ and $\mathbf{z}_i^{ideal}$ respectively, reflecting the worst and best objective function values in the Pareto optimal front. In the context of EMO and successive studies we replace $\mathbf{z}_i^{nadir}$ and $\mathbf{z}_i^{ideal}$ with currently available worst and best function values $\mathbf{z}_i^{max}$ and $\mathbf{z}_i^{min}$ respectively, as nadir and ideal vector values may not be available during an EMO algorithm run time.

One possible drawback with the ASF is the presence of a non-differentiable function, which inhibits the use of gradient-based mathematical programming algorithms for solving it. The deficiency can be overcome with an extra real-valued variable (α), new constraints and utilizing an equivalent differentiable formulation [9]:

$$\begin{array}{ll} \text{minimize} & \alpha, \\ \text{subject to} & [w_i(f_i(\mathbf{x}) - \bar{\mathbf{z}}_i)] \leqslant \alpha \text{ for all } i\text{=}1,\ldots,k, \\ \mathbf{x} \in \mathcal{S}, & \alpha \in R. \end{array} \quad (3)$$

The above formulation of an ASF may produce a weakly Pareto optimal solution and this can be avoided by adding an augmentation term. The augmented ASF is written as:

$$\begin{array}{ll} \text{minimize} & \max_{i=1}^{k} \frac{f_i(\mathbf{x}) - \bar{\mathbf{z}}_i}{\mathbf{z}_i^{max} - \mathbf{z}_i^{min}} + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x}) - \bar{\mathbf{z}}_i}{\mathbf{z}_i^{max} - \mathbf{z}_i^{min}}, \\ \text{subject to} & \mathbf{x} \in \mathcal{S}, \end{array} \quad (4)$$

where $\rho > 0$, binds the trade-offs called an augmentation coefficient. The above problem produces (properly) Pareto optimal solutions with bounded trade-offs only.

In the next section we briefly summarize the hybrid approach presented in [8].

III. Local Search Based EMO

We present a hybrid approach where we use the NSGA-II method [12] as the EMO algorithm and hybridize it with an ASF which is solved with any appropriate local search method. The local search is started from an offspring solution, which is considered as a reference point. The local search utilizes this reference point and minimizes the augmented ASF to obtain at least a locally Pareto optimal solution closest to the reference point.

The hybrid approach is as follows: In the t-th generation of the NSGA-II algorithm, a parent population P_t is subjected to selection, recombination and mutation operators and children Q_t are created. Later, each individual in the child population Q_t is evaluated and sent for local search with a probability p_l, which will lead to a new child population Q'_t. Thereafter, the parent and child populations are combined and a non-dominated sorting is performed. All the non-dominated individuals are then copied in P_{t+1}. If the size of P_{t+1} exceeds/is less than the population size, we reduce/increase the size of the population with a clustering procedure. Thereafter, the NSGA-II procedure continues as usual.

There are two significant changes that are made in the original NSGA-II procedure, clustering by k-means [13] replacing the crowding distance and the use of p_l for local search. During non-dominated sorting, the new population P_{t+1} is filled by solutions from different non-dominated fronts, one at a time (1 is the best level or rank) until the population size is met or exceeded. When the number of individuals in a particular rank is more than that needed to fix the population to a given size, we use clustering to cluster all the individuals of a particular rank in the objective space and then choose one from each of the clusters. If the cardinality of a cluster is more than one, we chose the individual closest to the centroid of the cluster. The hybrid approach uses a probability of local search p_l tracing a saw-tooth function, which periodically increases and drops linearly with generations. For example, starting from zero at the initial generation, the probability rises to 0.01 in

$(0.5N - 1)$ generations (where N is the population size) and drops to zero in $t = 0.5N$ generations. This means that when $N = 100$ and generation $= (0.5N - 1)$, on an average one solution in the entire population gets modified by the local search. The initial generations have a smaller local search probability, as typically the population is far from the Pareto-optimal front and the local search may mostly produce extreme Pareto-optimal solutions. The probability increases linearly as more solutions may need to be modified using the local search procedure to ensure convergence to the Pareto-optimal front. Probability p_l goes to zero after each period to prevent loss in diversity both during these initial phases and when the population approaches the Pareto-optimal front.

In the next section, we present the simulation results with parameter settings employed in the hybrid algorithm for testing the given set of multi-objective test problems.

IV. Numerical Simulation Results and Discussions

Our hybrid algorithm was tested using test problems suggested by Zhang et. al [14]. The test suite is a collection of thirteen unconstrained and ten constrained multi-objective problems. Each of the test problems was run thirty times independently, with different seeds, by pre-fixing the maximum number of function evaluations to be 300,000. Local search which was incorporated as an extra operator in our hybrid approach, uses KNITRO solver [15] with a sequential quadratic programming (SQP) as a local solver. The finite difference method was used to calculate the derivatives for SQP. IGD metric is used to measure the performance of the algorithm. For a detailed explanation, see [14].

All the tests have been executed on MACBOOK 2.1, Intel core 2 Duo 2.16 GHz processor, with 1GB RAM. The algorithm is coded using C programming language.

Any EMO usually involves a number of parameters and their setting greatly influences the efficacy of the algorithm. An hybrid algorithm usually involves two main types of settings, NSGA-II and local search specific parameter settings. Here, we briefly summarize them.

1) **NSGA-II parameters**:
   a) Population size: 100 for bi-objective problems, 150 for three objective problems and 300 for five objective problems.
   b) Crossover probability: 0.9.
   c) SBX distribution index: 5 for two and three objective problems. 10 for five objective problems.
   d) Mutation probability:
      i) Unconstrained: 0.01 for two and three objective problems. 0.033 for five objective problems.
      ii) Constrained: 0.1 for all test problems.
   e) Mutation distribution index: 15 for two and three objective problems. 20 for five objective problems.
2) **Local search parameters**:
   a) Probability of local search:
      i) Unconstrained: Probability follows a sawtooth function and reaches peak every 50 generations to 0.01 for two and three objective problems and every 25 generations to 0.01 for five objective problems.
      ii) Constrained: Probability follows a sawtooth function and reaches peak every 25 generations to 0.01.
   b) Maximum number of iterations in local search is fixed to be 50.
   c) The final relative stopping tolerance for the KKT (optimality) error is 10^{-4}.

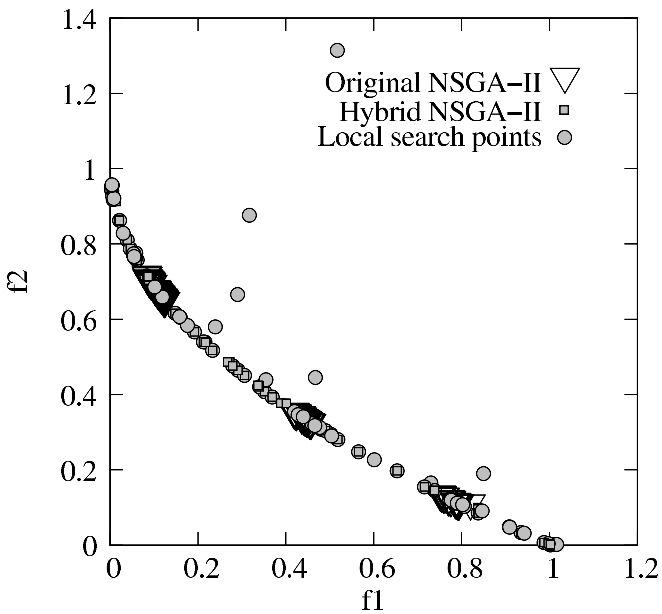

Fig. 1. Pareto front of UF1 obtained by hybrid approach and original NSGA-II.

In the next subsection, we present the results obtained with our hybrid algorithm on given test problems.

A. Simulation Results

In Tables I and II, the best and worst obtained IGD values with their corresponding time taken for all the unconstrained and constrained test problems respectively, are presented with their mean and standard deviation. The algorithm performances well on most of the test problems. Our hybrid approach involves a local search algorithm, which is derivative based. Hence, when the functions are non-differentiable, another optimization method should have been used. The five objective problems are rotated functions and the SBX recombination operator, which is a variable-wise recombination operator, does not perform well in such problems [16] and we suspect this to be a reason for high IGD values on such problems. The degraded performance on some of the constrained problems may be due to the constraint handling strategy, which is argued [17] to have problems to maintain diversity.

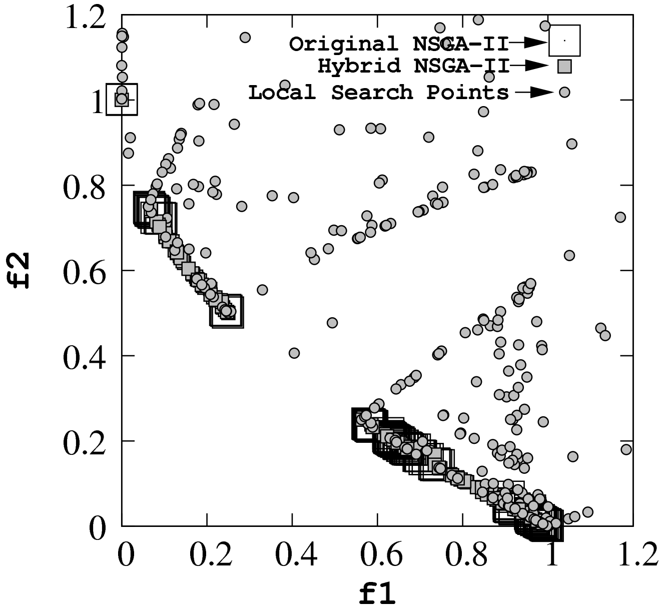

Fig. 2. Pareto front of CF2 obtained by an hybrid approach and an original NSGA-II.

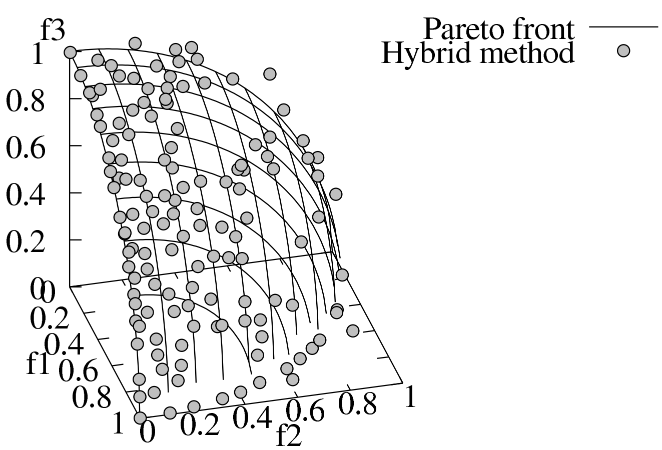

Fig. 3. Pareto front of UF8 obtained by hybrid approach.

Figures 1 and 2 show the non-dominated fronts obtained by original and hybrid approach for UF1 and CF2 respectively. It can be seen that, hybrid approach, was able to generate the entire non-dominated front as against the original NSGA-II. To investigate further the reason for the superior performance of our hybrid algorithm, we also mark all the points generated by the local search operator during a single run. Since most of the points on the front were a result of the local search, it clearly proves the ASF to be an effective scalarizing function, but it should not be concluded that local search was the sole reason for the good convergence. The ASF needs reference points which in turn were supplied by NSGA-II. Hence an effective integration of local search and an EMO algorithm, is the reason for a better performance of the hybrid algorithm. Figure 3 shows the non-dominated front for the three objective unconstrained UF8 problem and the reasons for its convergence can be similarly argued.

TABLE I
IGD VALUES OF UNCONSTRAINED MULTI-OBJECTIVE TEST PROBLEMS AND THEIR CORRESPONDING TIME TAKEN.

Test Problem	IGD values (Time in seconds)			
	Best	Worst	Mean	Standard deviation
UF 1	0.009851 (4)	0.04734 (4)	0.01153	0.0073
UF2	0.006025(8)	0.05455 (9)	0.01237	0.009108
UF3	0.03435 (6)	0.26207 (5)	0.10603	0.06864
UF4	0.04823 (7)	0.06975 (7)	0.0584	0.005116
UF5	0.29106 (5)	1.0498 (5)	0.5657	0.1827
UF6	0.08202 (5)	0.71745 (5)	0.31032	0.19133
UF7	0.007631 (5)	0.08801 (5)	0.02132	0.01946
UF8	0.06762 (16)	0.10911 (15)	0.0863	0.01243
UF9	0.03873 (9)	0.19140 (10)	0.0719	0.04504
UF10	0.5339 (7)	1.1266 (7)	0.84468	0.1626
UF11	0.1642 (95)	0.1836 (102)	0.1752	0.007103
UF12	78.16 (24)	207.834 (22)	158.05	40.437
UF13	2.7286 (780)	3.3937 (874)	3.2323	0.2273

Non-dominated fronts obtained for unconstrained problems are shown in Figures 4 to 13. Figures 14 to 26 show the non-dominated fronts for constrained problems. In addition, in Figures 22, 24 and 26, we also show a magnified version of the non-dominated set near the Pareto optimal front, as the final non-dominated set in these problems also has un-converged solutions due to insufficient number of function evaluations.

TABLE II
IGD VALUES OF CONSTRAINED MULTI-OBJECTIVE TEST PROBLEMS AND THEIR CORRESPONDING TIME TAKEN.

Test Problem	IGD values (Time in seconds)			
	Best	Worst	Mean	Standard deviation
CF 1	0.002665 (6)	0.01153 (6)	0.00692	0.0025062
CF 2	0.003381 (12)	0.0599 (11)	0.011836	0.01296
CF 3	0.11817 (5)	0.49155 (5)	0.23994	0.0858
CF 4	0.00912 (5)	0.02644 (6)	0.01576	0.00453
CF 5	0.1228 (6)	0.3447 (8)	0.1842	0.06077
CF 6	0.007197 (8)	0.09784 (8)	0.02013	0.01735
CF 7	0.08837 (4)	0.4751(5)	0.23345	0.08693
CF 8	0.08228 (25)	0.2309 (25)	0.11093	0.03682
CF 9	0.068661 (22)	0.1947 (20)	0.1056	0.02928
CF 10	0.2333 (14)	0.5010 (11)	0.3592	0.07503

V. CONCLUSIONS

In this paper, we have presented an efficient implementation of a local search procedure with an EMO algorithm. To take advantage of fast and accurate convergence to Pareto optimal solutions, EMO algorithms must use a directed and provable local search procedure. In this study, we have used an augmented achievement scalarizing function to be solved with an appropriate local search method. The local search procedure has been implemented as an additional operator and applied to EMO populations with a varying

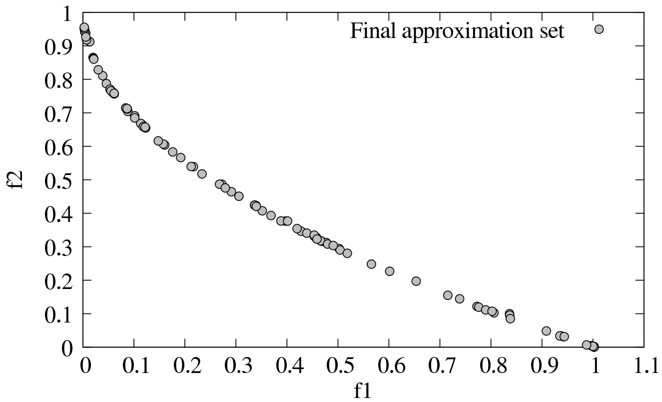

Fig. 4. Final approximation set - Unconstrained Problem 1.

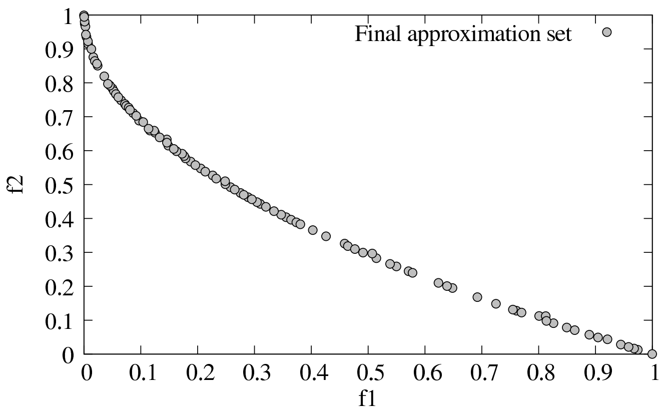

Fig. 5. Final approximation set - Unconstrained Problem 2.

probability. On a number of test problems provided in the test suite involving two to five objectives, we have observed that our proposed hybrid approach with NSGA-II is able to overcome different vagaries of landscapes, converging near to the Pareto optimal front.

ACKNOWLEDGMENTS

The support by the Academy of Finland (grant 118319) is highly appreciated.

REFERENCES

[1] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, UK, 2001.

[2] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multi-objective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.

[3] C. Grosan, A. Abraham, and H. Ishibuchi, *Hybrid evolutionary algorithms*, ser. Studies in Computational Intelligence. Springer, 2007, vol. 75.

[4] 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, 1998.

[5] A. Jaszkiewicz, "Genetic local search for multiple objective combinatorial optimization," *European Journal of Operational research*, vol. 137, no. 1, pp. 50–71, 2002.

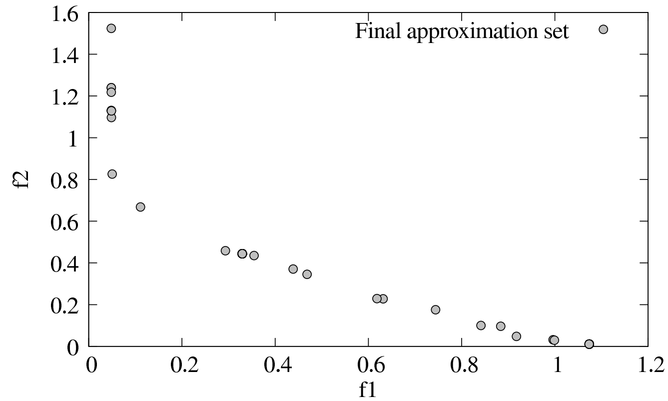

Fig. 6. Final approximation set - Unconstrained Problem 3.

[6] T. Goel and K. Deb, "Hybrid methods for multi-objective evolutionary algorithms," in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, L. Wang, K. C. Tan, T. Furuhashi, J. H. Kim, and X. Yao, Eds., vol. 1. Nanyang Technical University, 2002, pp. 188–192.

[7] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for solving Multi-Objective Problems*, 2nd ed., ser. Genetic and Evolutionary Computation, D. E. Goldberg and J. R. Koza, Eds. Springer, 2007.

[8] K. Sindhya, K. Deb, and K. Miettinen, "A local search based evolutionary multiobjective approach for fast and accurate convergence," in *Parallel Problem Solving from Nature – PPSN X*, ser. Lecture Notes in Computer Science, vol. 5199/2008. Springer Berlin/Heidelberg, 2008, pp. 815–824.

[9] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.

[10] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Congress on Evolutionary Computation (CEC 2002)*, vol. 1. IEEE Press, 2002, pp. 825–830.

[11] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *MCDM theory and Application*, ser. Lecture Notes in Economics and Mathematical Systems, G. Fandel and T. Gal, Eds., vol. 177. Springer-Verlag, 1980, pp. 468–486.

[12] K. Deb, S. Agarwal, 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.

[13] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[14] Q. Zhang, A. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC09 special session and competition," University of Essex and Nanyang Technological University, Technical Report CES-487, 2008.

[15] R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*, G. di Pillo and M. Roma, Eds. Springer-Verlag, 2006, pp. 35–59.

[16] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *GECCO '06: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO - 2006)*. ACM, 2006, pp. 1141–1148.

[17] C. A. C. Coello, "Theoritical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 1245–1287, 2002.

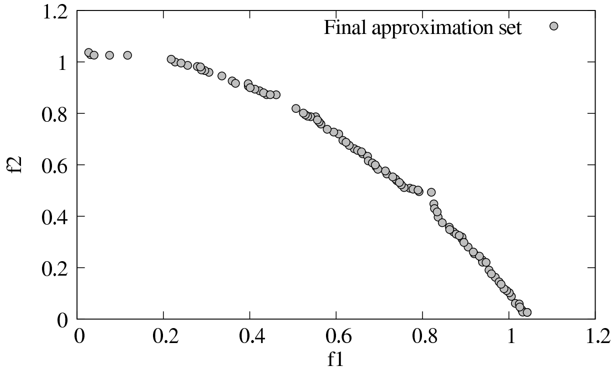

Fig. 7. Final approximation set - Unconstrained Problem 4.

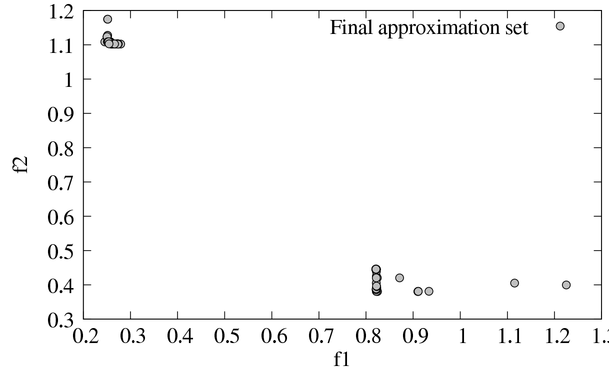

Fig. 8. Final approximation set - Unconstrained Problem 5.

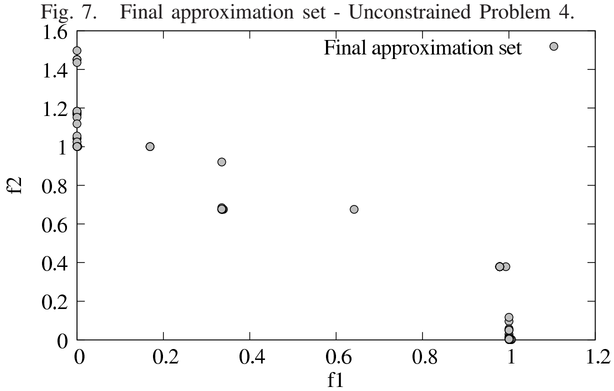

Fig. 9. Final approximation set - Unconstrained Problem 6.

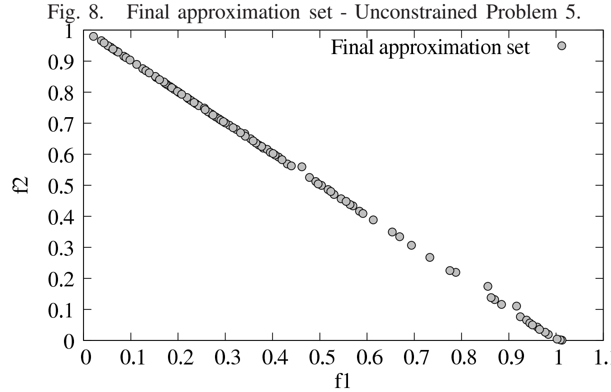

Fig. 10. Final approximation set - Unconstrained Problem 7.

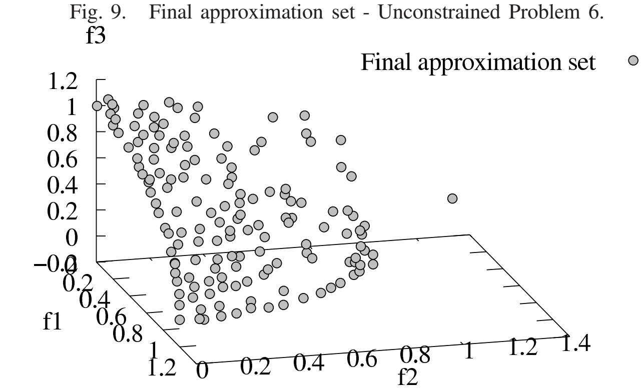

Fig. 11. Final approximation set - Unconstrained Problem 8.

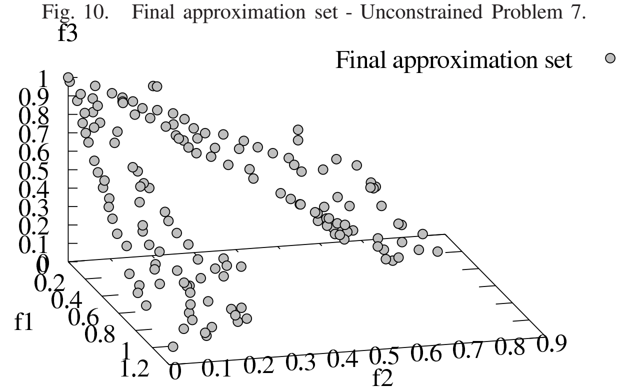

Fig. 12. Final approximation set - Unconstrained Problem 9.

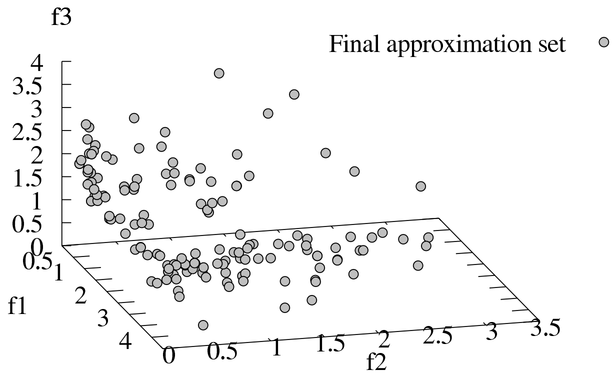

Fig. 13. Final approximation set - Unconstrained Problem 10.

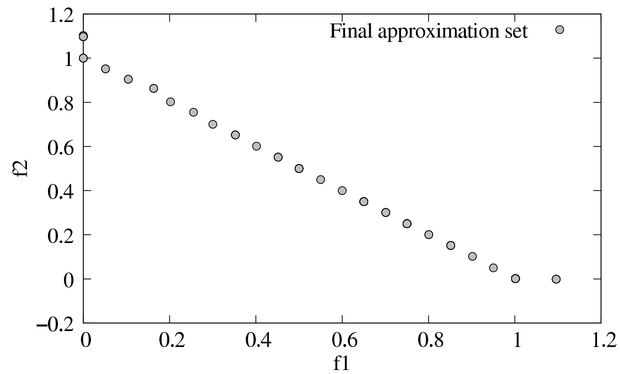

Fig. 14. Final approximation set - Costrained Problem 1.

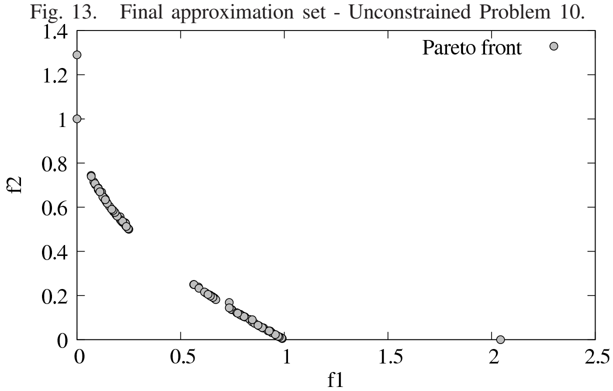

Fig. 15. Final approximation set - Constrained Problem 2.

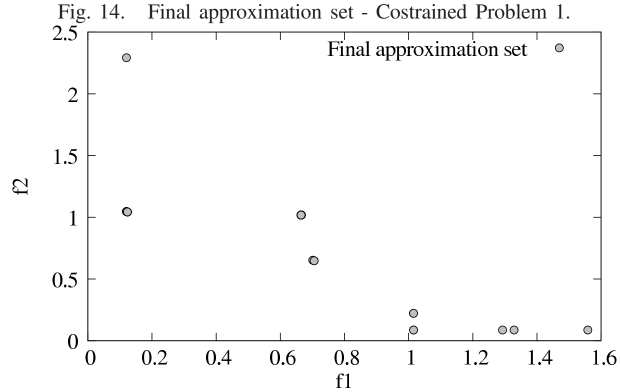

Fig. 16. Final approximation set - Constrained Problem 3.

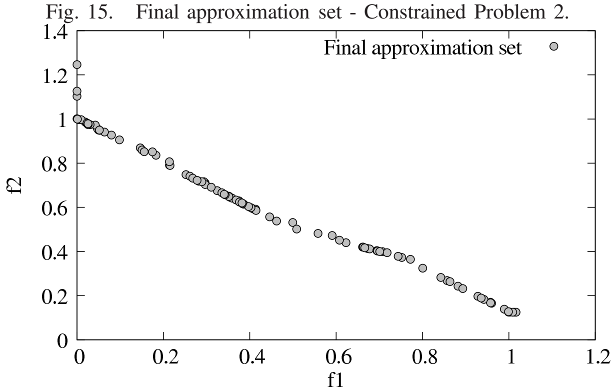

Fig. 17. Final approximation set - Constrained Problem 4.

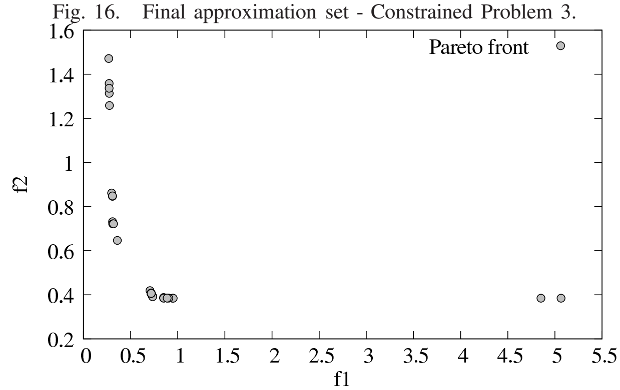

Fig. 18. Final approximation set - Constrained Problem 5.

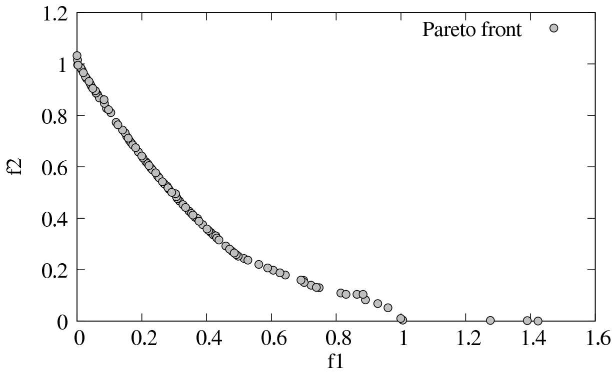

Fig. 19. Final approximation set - Constrained Problem 6.

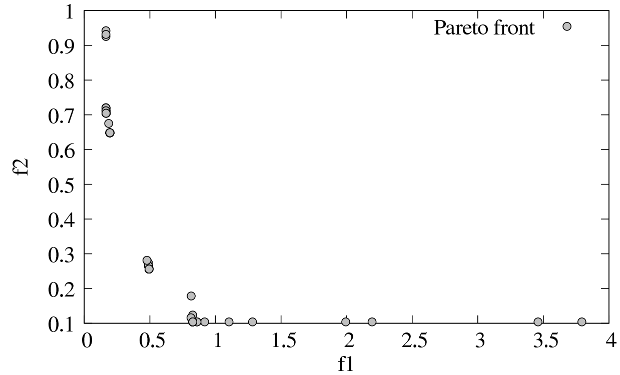

Fig. 20. Final approximation set - Constrained Problem 7.

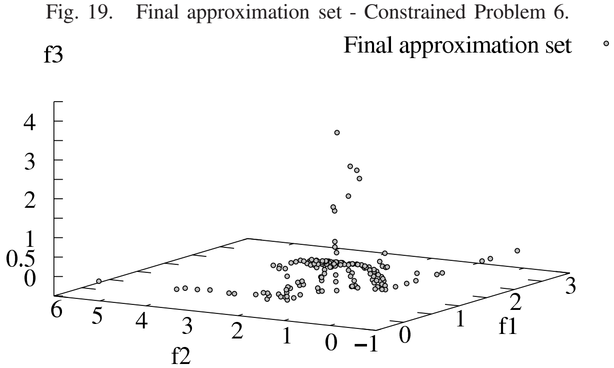

Fig. 21. Final approximation set - Constrained Problem 8.

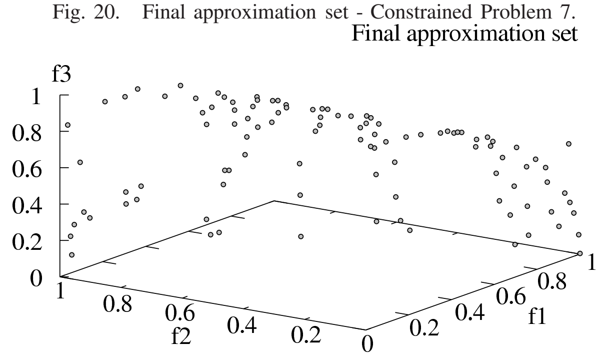

Fig. 22. Final approximation set (Magnified)- Constrained Problem 8.

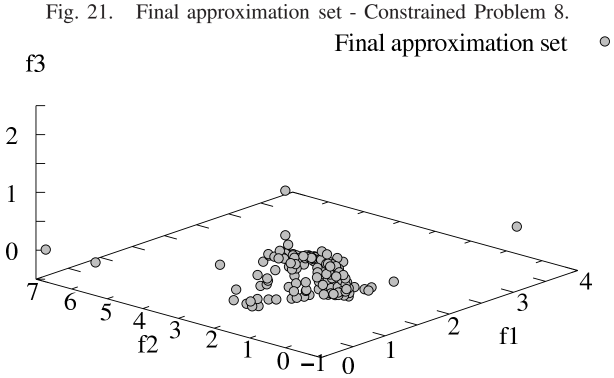

Fig. 23. Final approximation set - Constrained Problem 9.

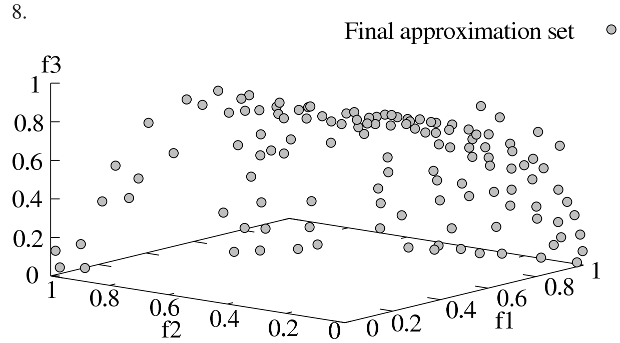

Fig. 24. Final approximation set (Magnified)- Constrained Problem 9.

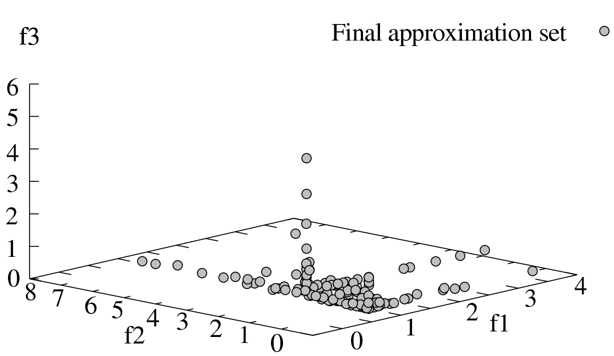

Fig. 25. Final approximation set - Constrained Problem 10.

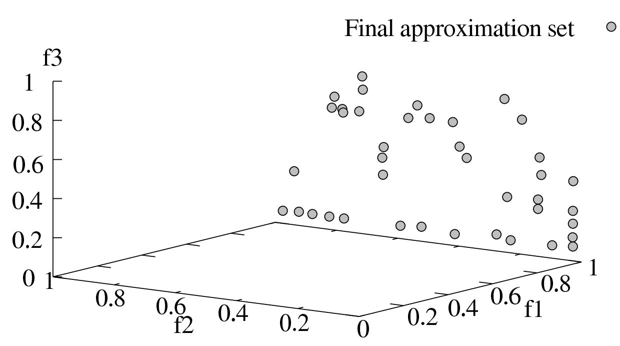

Fig. 26. Final approximation set (Magnified) - Constrained Problem 10.