On The Use of Informed Initialization and Extreme Solutions Sub-population in Multiobjective Evolutionary Algorithms

Dipankar Dasgupta*, German Hernandez†, Andres Romero†, Deon Garrett*, Aishwarya Kaushal*, James Simien‡
* Department of Computer Science
University of Memphis
Memphis, TN, USA
Email: dasgupta@cs.memphis.edu, deong@acm.org, xxxxx@memphis.edu
† Department of Computer Science
National University of Columbia
Bogota, Columbia
Email: xxxxxxx@xxxxx.xxx, xxxxxxx@xxxxx.xxx
‡ Navy Personnel Research, Studies, and Technology
Millington, TN, USA
Email: james.simien@navy.mil

***Abstract*—This paper examines two strategies in order to improve the performance of multi-objective evolutionary algorithms when applied to problems with many objectives: informed initialization and extreme solutions sub-population. The informed initialization is the inclusion of approximations of extreme and internal points of the Pareto front in the initial population. These approximations, called informed initial solutions, are found using a high quality evolutionary or local search algorithm on single objective problems obtained by scalarizing the multiple goals into a single goal by the use of weight vectors. The extreme solutions sub-population is proposed here to keep the best approximations of the extreme points of the Pareto front at any point of the evolution, and the selection scheme is biased to give these solutions slightly higher chances of being selected. Experimental results applying these two strategies in continuous and combinatorial benchmark problems show that the diversity in the final solutions is improved, while preserving the proximity to the Pareto front. Some additional experiments that demonstrate how the number of initial informed solutions affects the performance are also presented.**

I. Introduction

Over the past ten years, a large amount of research has been conducted on the use of evolutionary techniques to solve multiobjective optimization problems. The inherent use of a population of candidate solutions provides benefits unattained by other techniques when applied to problems with two or three objectives. In recent years some attention has been devoted to the experimental study of the performance of these evolutionary techniques in many objective problems. These are optimization problems with more than three conflicting objectives. The experimental studies conducted thus lead to the conclusion that the performance of the evolutionary techniques is severely hindered when the dimension of the objective space grows. A recent study [3] shows that for problems with more than ten objectives a purely random search may perform favorably when compared with an evolutionary technique.

It is well-known that for single objective problems the performance of evolutionary algorithms can often be improved through the introduction of a local search operator. These combinations of evolutionary and local search are known by the names of *Memetic Algorithms* (MAs) [24], *Hybrid Evolutionary Algorithms* [10], [11], [17], and *Genetic Local Search* (GLS) [23]. These hybrids of evolutionary and local search algorithms have been shown to provide state-of-the-art performance on a wide range of hard single objective combinatorial optimization problems [23], [21], [25] and also have shown to provide a good performance on multiobjective optimization problems [15], [16], [20], [17], [19], [22].

This paper proposes two modifications to improve the performance of multiobjective evolutionary algorithms through the use of a particular form of hybridization. A high performance single objective search algorithm is used to seed the initial population with solutions to a scalarized version of the problem. In this work, we consider two such search methods: a linear programming algorithm for the Sailor Assignment Problem as well as a high performance evolutionary algorithm for general optimization. In addition, these initial solutions are maintained in an external population throughout the evolutionary process and given additional mating chances. We show that these modifications provided significant performance improvements over conventional multiobjective evolutionary algorithms.

This paper is organized as follows; Section II introduces the MOEAs used in this work, NSGA-II and SPEA2, and gives a brief overview of hybrid algorithms that combine MOEAs and other search strategies; Section III describes the benchmark problems used in this study and introduces the concept of informed initialization and extreme solutions sub-population.

Sections IV and V present the experiments and results used in this study, and Section VI briefly discusses the conclusions and future work.

II. MOEAs and Hybridized Algorithms

MOEAs have two primary goals: to find ”good” solutions (near the true Pareto optimal front), and to maintain diverse alternatives along the approximation to the Pareto front. Both NSGA-II and SPEA2, the algorithms considered in this work, achieve these goals by assigning fitness in such a way as to bias selection toward non-dominated solutions that do not resemble other known solutions.

A. NSGA-II

The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [5] uses the concept of a domination level to assign fitness. All non-dominated individuals are assigned level zero. Upon removing those individuals from the population, the newly non-dominated solutions are assigned level one. The process continues until all solutions have been assigned a non-domination level. Compared to SPEA, NSGA is able to maintain a better spread of solutions and convergence in the obtained non-dominated Pareto front. Like SPEA2, NSGA-II then employs a density estimation technique to estimate which solutions may be removed with the least impact on diversity.

B. SPEA2

SPEA2, the Strength Pareto Evolutionary Algorithm, was first published by Zitzler et al. in 2001 [28]. SPEA2 utilizes a concept called strength, a concept related to the number of other individuals a given solution dominates. The fitness of a dominated individual is computed based on the strengths of all other individuals that dominate it. The algorithm ensures that all non-dominated solutions have lower fitness values than any dominated solution, where minimization is the goal of the algorithm, thereby biasing selection toward non-dominated solutions. In addition, the distance to a solutions nearest neighbors is used to apply additional pressure toward individuals in sparsely populated regions of the Pareto front.

C. Hybrid Algorithms

Despite their popularity, pure evolutionary approaches often have some difficulty in finding truly optimal solutions, particularly in combinatorial optimization problems. The solutions obtained by these MOEAs are often clustered in a region of relatively low quality. Inclusion of a local search routine can help push solutions to improve both in their proximity to the Pareto front as well as improving the diversity of solutions maintained by the algorithm. In this work, we consider a linear programming algorithm as well as a high performance evolutionary algorithm, CHC, to generate the locally improved solutions that will participate in the evolutionary process.

Like any hybrid algorithm, there are a number of choices for how to integrate the different search algorithms into a single method. In this work, we describe a two-part process by which CHC is used to generate a small number of initial solutions to the problem, which are then used to seed the initial population of a conventional multiobjective evolutionary algorithm. Then during the run of the MOEA, these “informed” solutions are maintained and given additional opportunities to participate in the evolutionary process. This process consists of adding to the initial population of a MOEA some individuals which are very good approximations to 1) the extreme solutions of the Pareto front and 2) some few internal points on the Pareto front trying to guide the remaining individuals in the population towards the Pareto front.

CHC [9] is an evolutionary algorithm combining strongly elitist environment selection driving the population toward uniformity with a novel restart mechanism allowing the algorithm to successfully continue searching after it has converged. In addition, CHC utilizes a crossover operator designed to ensure that offspring are maximally distant from their parents. This combination of features has led to CHC performing very well on a wide array of numeric and combinatorial optimization problems. Algorithm 1 below briefly describes the steps of CHC.

Algorithm 1 CHC Algorithm

$t \leftarrow 0$
Initialize $(Pa, ConvergenceCount, k)$ {Pa: Population}
while not EndingCondition (t, Pa) **do**
  $Parents \leftarrow$ SelectionParents$(Pa, ConvergenceCount)$
  $Offspring \leftarrow$ HUX$(Parents)$
  Evaluate$(Offspring)$
  $Pn \leftarrow$ ElitistSelection $(Offspring, Pa)$ {Pn: new population}
  if not modified (Pa, Pn) **then**
    $ConvergenceCount \leftarrow ConvergenceCount - 1$
    if $ConvergenceCount \leq -k$ **then**
      $Pn \leftarrow$ Restart(Pa)
      Initialize $ConvergenceCount$
    end if
  end if
  $t \leftarrow t + 1$
  $Pa \leftarrow Pn$
end while

In order to generate such initial solutions, the multiobjective problem is scalarized using particular weight vectors and solved using the CHC single objective genetic algorithm. The weight vectors are chosen so that each objective is considered in isolation, yield points at every extreme of the Pareto front. In addition, some number of additional points are chosen to push toward the interior regions of the front. The weight vectors for these interior points may be chosen randomly or systematically, e.g., each objective equally weighted. With these two types of vectors, it is possible to generate good approximations to be inserted into the initial population of NSGA-II; these approximations are generated by running the CHC algorithm using each one of the weight vectors.

The solutions generated using the CHC algorithm with the weight vectors are kept in a sub-population located in the main population of the NSGA-II algorithm; this sub-population is preserved along all the generations of the algorithm. This is because that sub-population contains very good approximations to the pareto front which must be preserved. Of course those solutions are not perfect and can be replaced from the sub-populations if in the remaining of the populations appears a better solution; that is, a solution that dominates one of the solutions in the sub-population will replace it.

The NSGA-II selection scheme was then modified to allow the solutions from the informed sub-population to recombine more often. The standard tournament selection procedure takes two solutions and first checks for dominance. If one of the solutions dominates the other, then that solution is selected; if the solutions are non-dominated, then the crowding distance is used to compare and select one of them. The modification proposed here adds an intermediate step, if the solutions are non-dominated then the one that is generated from the informed sub-population is selected. If no solution comes from the informed sub-population or if more than one solution is generated, then the crowding distance should be checked.

III. Benchmark Problems

A. DTLZ3

In [6], [7], Deb et. al. introduced several scalable multiobjective test problems. Among the most challenging, DTLZ3 contains an exponential number of false Pareto fronts which can trap an optimization algorithm. By scalable, it is meant that the problems may be defined with arbitrary numbers of objectives and decision variables without losing their basic character. For the purposes of this study, we chose DTLZ3 with both six and 12 objectives, using the recommendations of [6] for all other parameters.

B. QAP

The Quadratic Assignment Problem (QAP) is one of the fundamental and very popular combinatorial optimization problems. It was first introduced by Koopmans and Beckmann in 1957 [2] to model the problem of locating the departments with material flow between them.

The QAP problem can be stated as: A set of *n* facilities or departments and a set of *n* locations are given. For each pair of facilities weight or flow is given; and for locations, distance is given. The goal is to assign all facilities to different locations in such a way that the sum of the distances multiplied by the corresponding flows is minimized. The QAP can be described as follows: Given two $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$, find a permutation π that minimizes

$$\min_{\pi} \mathcal{F}(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} B_{\pi_i,\pi_j}. \quad (1)$$

Conventionally, the matrices $\mathbf{A}$ and $\mathbf{B}$ are called the *distance* and *flow* matrices, the terminology arising from the original formulation of QAP as a facilities layout problem. In a facility layout problem there are *n* number of departments or facilities to be assigned to *n* number of locations. It is for this reason that a QAP formulation requires an equal number of departments or facilities and locations. Despite the terminology, QAP is useful in modeling several disparate application areas, including backboard wiring, hospital layout, and keyboard design. This assignment problem is not only $\mathcal{NP}$-hard and $\mathcal{NP}$-hard to approximate [26], but it is also practically intractable problem. Due to the extreme difficulty of establishing an exact solution in large QAP instances, researchers have long turned to meta-heuristic approaches such as tabu search, simulated annealing, and evolutionary algorithms. QAP exhibits the useful property that the fitness of a new solution differing from the previous solution in only one swap can be calculated in constant time [27]. The impact of this property is that local search algorithms for the QAP can examine large numbers of candidate solutions very efficiently. As a result, local search algorithms, often based on Tabu Search, tend to dominate in sheer numbers and in performance to other approaches.

The multiobjective QAP (mQAP) was proposed by Knowles and Corne [17], [18] as a benchmark problem for multiobjective metaheuristics such as evolutionary algorithms. The mQAP consists of a single $n \times n$ distance matrix, and k distinct $n \times n$ flow matrices. There exist then k different pairings of the distance matrix with one flow matrix, yielding k independent single objective QAP problems. The objective function value of a permutation π is thus a k-dimensional vector with

$$\mathcal{F}^m(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} B^m_{\pi_i,\pi_j} \quad \forall m : 1 \leq m \leq k. \quad (2)$$

The mQAP models any sort of facilities layout problem in which the minimization of multiple simultaneous flows is required.

C. SAP

According to the United States Navy's personnel policies, roughly every three years sailors serving on active duty are reassigned to a different job. As a result, at any given time there exists a sizable population of sailors to be reassigned to available jobs. Currently, more than 300,000 sailors serve in the Navy and more than 120,000 are reassigned each year [14]. Sailors must be reassigned in such a way as to satisfy their individual preferences and the needs of the Navy. Many factors go into determining a good set of assignments. The Navy must ensure that not only are the sailors and commanders satisfied with the assignment, but also the match must be implemented within a fixed budget. If the resulting cost is too expensive, then the Navy cannot adequately maintain its other priorities. However, if the assignment is overly focused on costs, it may be that the sailors are unhappy with their assigned jobs. This could lead to a decrease in morale, followed by decreased retention rates and other problems. Thus, the Navy's goal is to identify sailor and job matches that maximize some overall criterion of desirability and is referred to as the Sailor Assignment Problem (SAP). In particular, the SAP involves finding a balanced range of solutions that maximize the total training score, the sailor preference score, and the commander preference score, and minimize the PCS cost.

In [13], genetic algorithms were tested on single objective versions of the problem and compared against results of the Gale-Shapley algorithm that solves in $O(n^2)$ time the stable marriage problem. In [12] it was shown that when NSGA-II [5] or SPEA2 [28] are applied to multiobjective instances of the sailor assignment problem (discussed later), it results in solution that lack adequate diversity and that combining the evolutionary algorithms with a rudimentary local search operator, i.e. an hybrid algorithm, achieves greatly improved diversity.

In [4] a different and more efficient hybrid algorithm was introduced for SAP combining NSGA-II with the Kuhn-Munkres algorithm. The Kuhn-Munkres algorithm solves linear assignment problems in $O(n^3)$ time and it was chosen due to fact that the single objective versions of SAP with small modifications are linear assignment problems. Because the Kuhn-Munkres algorithm does fully solve the SAP, one option is to simply apply the algorithm to a wide range of scalarizations of the problem yielding a good approximation to the true Pareto front. This approach, while yielding very good solutions, suffers from very long run times as the problem sizes are increased. Thus, one useful goal is to use metaheuristics to reduce the computation time required, while maintaining a reasonably good approximation to the front as found by the linear programming algorithm.

IV. Experiments

In order to validate the performance of the proposed approach, experiments were carried out utilizing the jMetal platform [8] and using the scalable multiobjective test problems defined in [7]; specifically, the DTLZ3 problem has been tested in 6 and 12 dimensions. In addition, the approach was tested on the SAP with the linear programming initialization as well as the QAP using CHC for the local improvement. To measure the performance of the algorithms, the *coverage* and *proximity* metrics [1] were used. The metrics are briefly described below.

A. Proximity

The distance between the approximation set S and the pareto front P_F is defined as the average of the minimum distance between a solution and the pareto optimal front over each solution in S, as in [1]:

$$D_{S\to P_F}(S) = \frac{1}{|S|}\sum_{Z^0\in S} min_{Z^1\in P_F} d(Z^0, Z^1), \qquad (3)$$

where $d(Z^0, Z^1)$ is the euclidean distance between the two solutions. A small value for this indicator means that all the points in the approximation set are, on average, close to the true pareto front. An ideal value of 0 is obtained when all the points are actually in the pareto front.

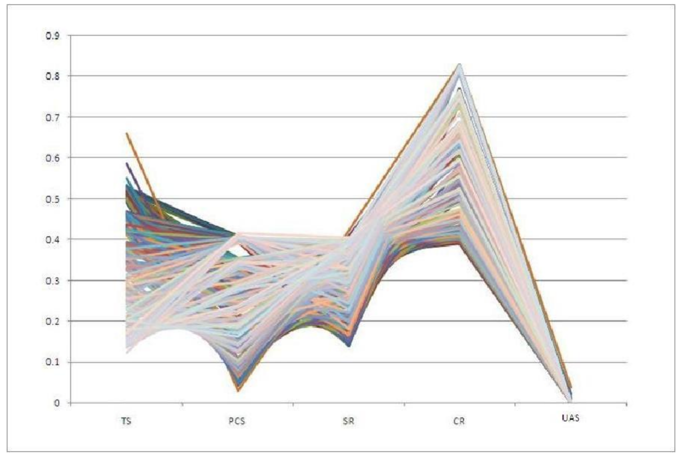


Fig. 1. **Kuhn-Munkres complete sample of the Pareto front**

B. Diversity

To measure the diversity of the approximation set the reverse of the proximity metric is used, as defined in [1]:

$$D_{P_F\to S}(S) = \frac{1}{|P_F|}\sum_{Z^1\in P_F} min_{Z^0\in S} d(Z^0, Z^1). \qquad (4)$$

In this indicator, for each solution in the pareto front the distance to the closest solution is calculated, and the average is taken as the value for the indicator. A small value for this indicator means that all the points in the true pareto front have, at least, one point in the approximation set which is very close. The ideal value of 0 is obtained when all the points in the pareto front are also contained in the approximation set.

V. Results

A. NSGA-II with Kuhn Munkres

As described above, in [4], NSGA-II and SPEA2, with informed initialization were applied on large instances of SAP problem. To illustrate how the diversity is improved with informed initialization, figures 1, 2 and 3 show the experimental results for a small instance of SAP with 1,000 sailors and 1,100 jobs . Fig.1 shows a parallel plot for the objective values of the assignments found a complete sample of obtained solving the single objective versions corresponding to the 287 weight vectors imposed by discretizing the simplex by 0.1 in each dimension using the Kuhn-Munkres algorithm. In the parallel plots, each objective is mapped onto one "bar" of the graph, and a single Pareto set approximation consists of a number of solutions, drawn as lines connecting the solution's values for each objective function respectively.

Obviously, including the informed initialization into NSGA-II provides significantly increased performance, and begins to approach the performance of the full Kuhn-Munkres solution while using dramatically fewer computational resources. In preliminary testing, the evolutionary methods produced a five to ten fold speedup over running the full linear programming algorithm for every weight vector [4]. However, this performance does come at the cost of solution quality in the interior

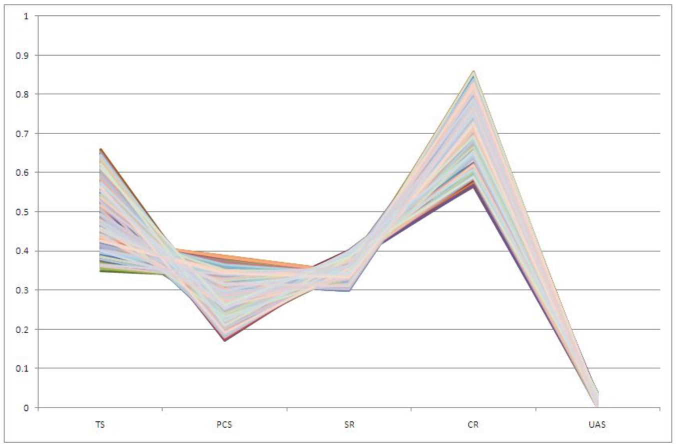


Fig. 2. **NSGA-II solutions**

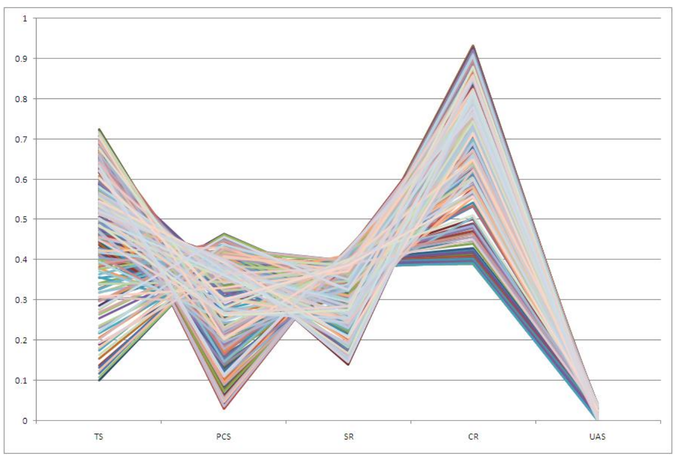
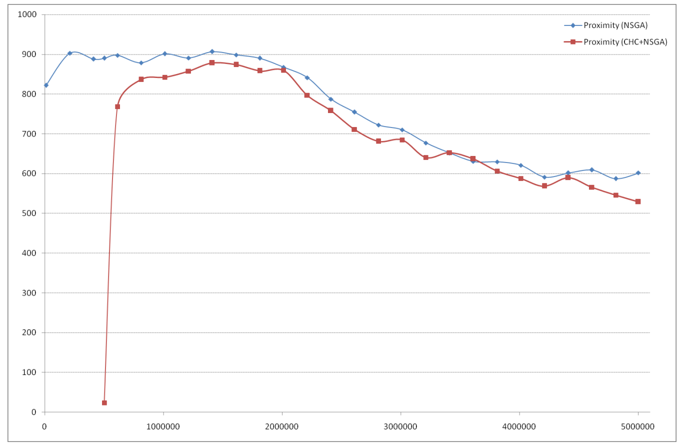


Fig. 3. **NSGA-II with Kuhn-Munkres Informed Initialization (5 initial solutions)**

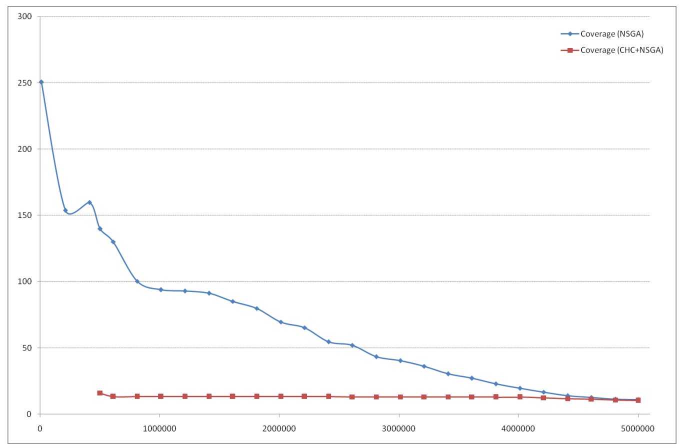
regions of the Pareto front. The full Kuhn-Munkres results show a marked curve along the bottom edge of the parallel plots which is not present in the results of NSGA-II with the Kuhn-Munkres initialization. This curve is created by the increase in solution quality of the average interior solution in the full Kuhn-Munkres results. Essentially, many of the interior "lines" in the parallel plot are lower in several dimensions simultaneously, leading to an increase in the number of lines near the bottom edge of the curve. Considering the hypervolume, shown in Table I provides another view of this effect.

TABLE I
HYPERVOLUME OF NSGA-II WITH INFORMED INITIALIZATION VERSUS THE FULL LINEAR PROGRAMMING SOLUTION ON THE SAP. COLUMN HEADINGS REFER TO THE NUMBER OF SAILORS/JOBS IN THE PROBLEM INSTANCE.

Algorithm	1000/1100	2000/2100	4000/4500
LP	0.3393	0.2394	0.2928
NSGA-II	0.1362	0.0806	0.0696
NSGA-II+LP	0.2331	0.1137	0.1840



(a) DTLZ3, M=6, Proximity



(b) DTLZ3, M=6, Coverage

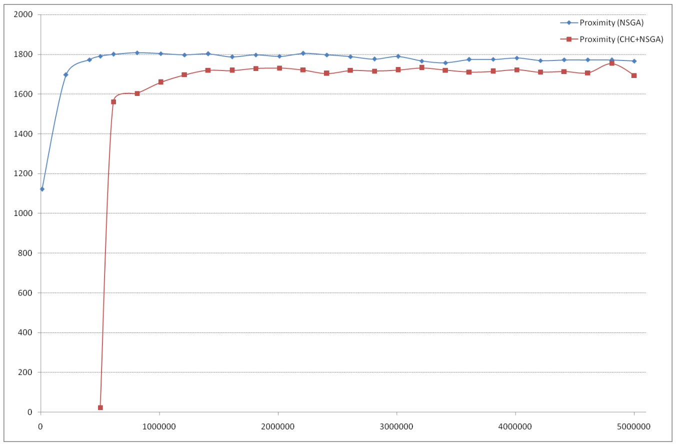
Fig. 4. DTLZ3 problem on 6 objectives with NSGA-II with informed initialization of 7 solutions

B. NSGA-II with CHC

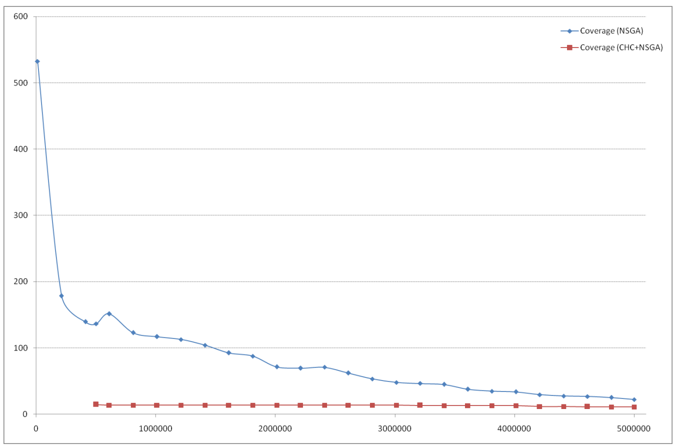
The proposed approach was tested using the DTLZ3 problem, on 6 and 12 objectives, performing 5 million function evaluations; Fig.4 and Fig.5 show proximity and diversity of the populations of solutions for the NSGA-II standard and NSGA-II with informed initialization of 7 solutions.

In Fig.4(a), the proximity indicator for the two algorithms (NSGA-II standard and NSGA-II with informed initialization) shows a decreasing value, which means that the two algorithms are moving all the non-dominated points toward the actual pareto front. Note the small value found at the beginning of the NSGA-II with informed initialization algorithm. This is because in the first generations, the set of non-dominated solutions consists only of the approximations obtained with CHC, which are solutions closely resembling the actual pareto front. Then, the indicator gets a higher value as the number of points in the non-dominated set gets higher and, on average, those points are farther from the pareto front than those of the initial solutions.

On the other hand, in Fig.4(b) the value for the coverage indicator is very good from the beginning in the NSGA-II

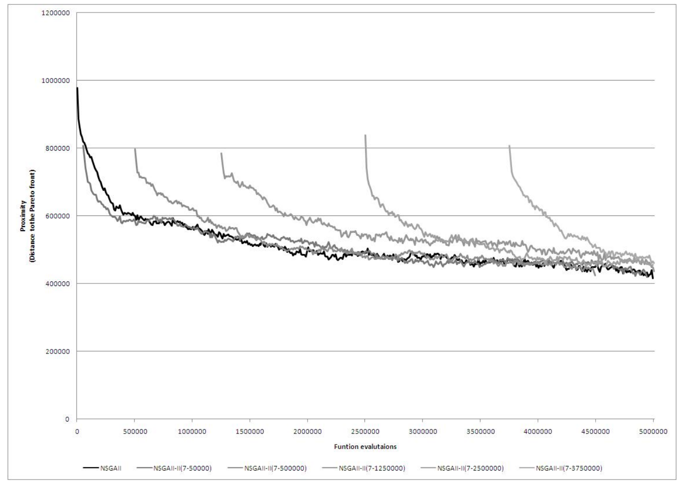


(a) DTLZ3, M=12, Proximity

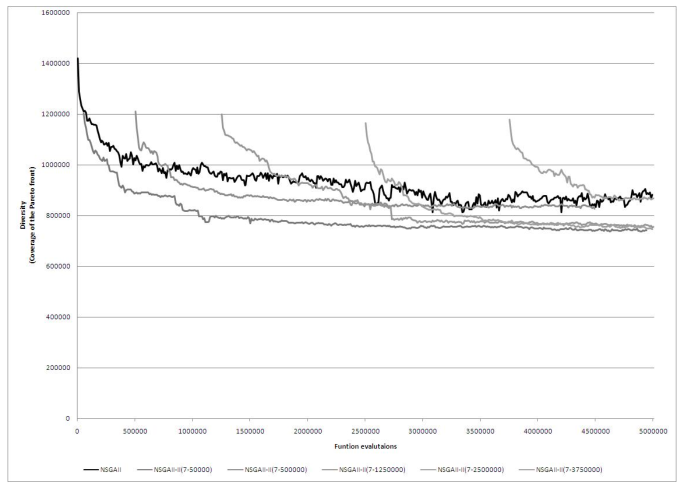


(b) DTLZ3, M=12, Coverage

Fig. 5. DTLZ3 problem on 12 objectives with NSGA-II with informed initialization of 13 solutions



(a) mQAP with six objectives, Proximity



(b) mQAP with six objectives, Coverage

Fig. 6. Performance on mQAP with 6 objectives with NSGA-II-informed initialization of seven solutions and increasing number of evaluations to CHC

with informed initialization algorithm, and, for the NSGA-II standard algorithm. This indicator is very high and gets a good value only after several generations. To understand this behavior, this indicator should be seen along with the proximity indicator: a low coverage with a high proximity value mean that the approximation set is on average far from the pareto front, but there are some feasible solutions very good which are close to the front. When the proximity value decreases, this signifies that the solutions are moving toward the pareto front and are covering it more effectively. This performance is obtained consistently in the NSGA-II with informed initialization algorithm, but is only reached after several (in this case almost 5,000) generations in the NSGA-II standard algorithm.

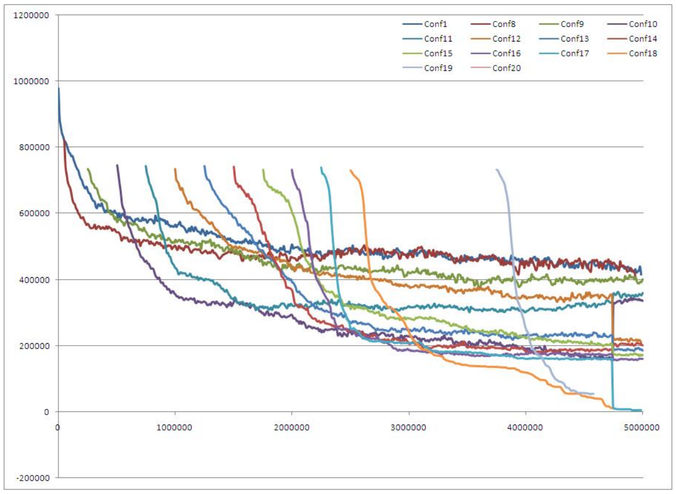
The proposed approach was also tested using the mQAP problem, performing 5 million function evaluations. Fig.6 shows the proximity and diversity of the populations of solutions for the NSGA-II standard and NSGA-II with informed initialization of seven solutions for increasing numbers of CHC evaluations. Table II shows the average final proximity and coverage for 30 runs, showing that the informed ini-

TABLE II
MQAP WITH SIX OBJECTIVES

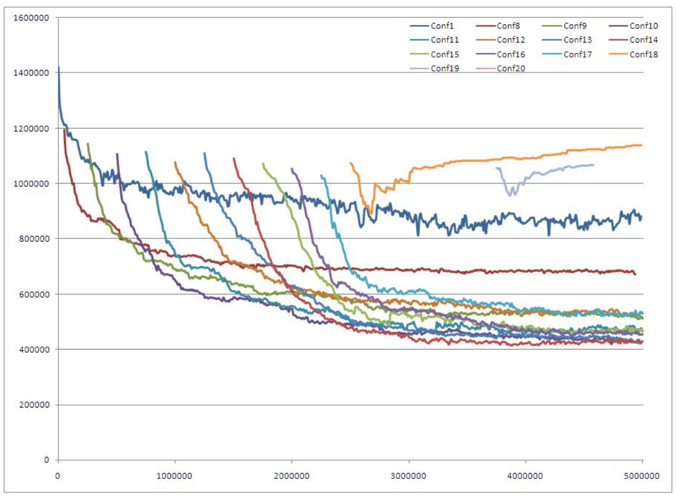
CHC Evaluations	NSGA-II + 7 solutions of CHC	
	Proximity	Coverage
0	415738.4	881203.7
50000	419072.5	2.741834.2
500000	424375.4	826684.2
1250000	435831.0	755655.4
2500000	455340.3	745563.7
2500000	461200.3	745563.0

tialization has a positive impact increasing the diversity and proximity, but the extra time devoted with CHC to find this initial informed solutions show no additional impact in the performance.

Finally, Fig.7 shows proximity and diversity of the populations of solutions for the NSGA-II standard and NSGA-II with informed initialization as the relative search effort allocated to CHC is increased. There are two ways of incrementing the number of evaluations performed by CHC: one is running



(a) mQAP, m=6, Proximity



(b) mQAP,m=6, Coverage

Fig. 7. mQAP with NSGA-II with informed initialization with increasing number of informed solutions

TABLE III
MQAP WITH SIX OBJECTIVES

CHC		NSGA-II + 7 solutions of CHC	
Informed solutions	Evaluations	Proximity	Coverage
0	0	415738.4	881203.7288
7	35000	410897.4	671413.2048
50	250000	400758.4	512040.2724
100	500000	356457.3	462133.6121
200	1000000	339632.9	452533.6798
300	1500000	209061.7	429534.2537
400	2000000	185116.0	421772.8846
500	2500000	171862.1	423019.2604
600	3000000	145711.3	529289.8992
700	3500000	17595.9	530252.3769
800	4000000	17375.0	1137618.673
900	4500000	23875.7	1067919.391

TABLE IV
DIFFERENT CONFIGURATIONS TO CHANGE THE % OF EVALUATIONS PERFORMED BY CHC

Config	CHC Evals	CHC Runs	CHC_{frac}	NSGA-II Evals
1	0	18	0	5000000
2	50000	18	0.01	4950000
3	500000	18	0.1	4500000
4	1250000	18	0.25	3750000
5	2500000	18	0.5	2500000
6	3750000	18	0.75	1250000
7	5000000	18	1	0
8	50000	20	0.01	4950000
9	250000	100	0.05	4750000
10	500000	200	0.1	4500000
11	750000	300	0.15	4250000
12	1000000	400	0.2	4000000
13	1250000	500	0.25	3750000
14	1500000	600	0.3	3500000
15	1750000	700	0.35	3250000
16	2000000	800	0.4	3000000
17	2250000	900	0.45	2750000
18	2500000	1000	0.5	2500000
19	3750000	1500	0.75	1250000
20	5000000	2000	1	0

CHC more times which gives more informed solutions in the initial populations of NSGA-II; the second way is to perform larger runs of CHC which gives the same number of informed solutions, but those solutions are optimistically closer to the actual pareto front. Table IV shows the different configurations tested. Table III shows the average final proximity and coverage for 30 runs, showing that if the number of initial informed solutions is increased the performance increases, meaning better proximity and diversity, but only if the number of total function evaluations does not exceed 50% of the total number of evaluations. Here, in Fig.7 the different configurations are used to change the $\%$ of evaluations performed by CHC with a basis of 5 million evaluations in total.

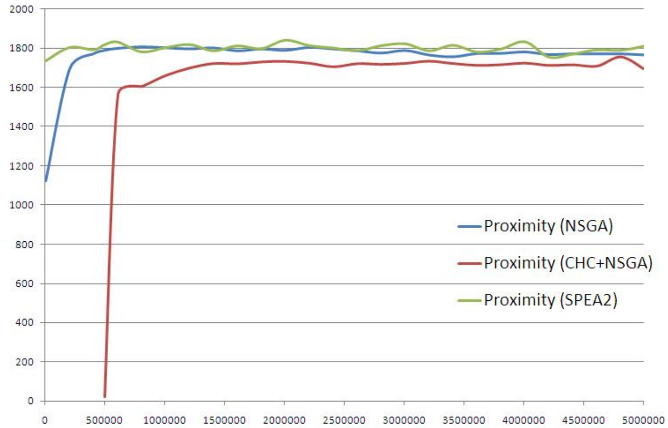
VI. CONCLUSIONS

In this paper, we have shown that the performance in terms of proximity and coverage of multiobjective algorithms can be improved through the injection of solutions generated by a high performance single objective search algorithm on various test problems. In addition, the selection scheme for a conventional multiobjective evolutionary algorithm was modified to increase the likelihood of selecting the injected solutions. We show that the best results in terms of proximity and coverage are obtained from NSGA-II with informed initialization when compared to the standard NSGA-II and SPEA2 algorithms for DTLZ3 test problem on 12 objectives.

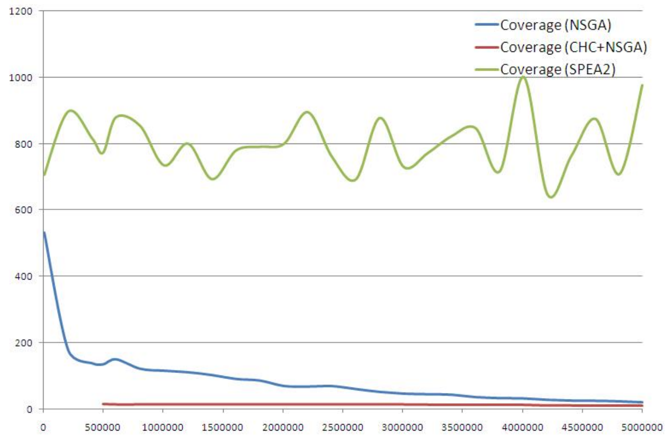
Many hybrid MOEAs operate by subjecting the individuals to local search throughout the evolutionary process. In this work, we consider the ability of a multiobjective evolutionary algorithm to achieve good results through the injection of good solutions into the initial population. While performance was certainly improved, there remain several open questions regarding the relative merits of various forms of hybridization. This work provides a preliminary study from which many of these questions may be addressed in future work.

REFERENCES

[1] P. A. N. Bosman. *Design and Application of Iterated Density-Estimation Evolutionary Algorithms*. PhD thesis, Institute of Information and Computing Sciences, Universiteit Utrecht, Utrecht, The Netherlands, 2003.



(a) DTLZ3 on 12 objective, Proximity



(b) DTLZ3 on 12 objective, Coverage

Fig. 8. Comparison of the proximity and coverage of NSGA-II, SPEA2 with NSGA-II with informed initialization

[2] T. C.Koopmans and M. Beckmann. Assignment problems and location of economic activities. *Econometrica*, 25:53–76, 1957.

[3] D. W. Corne and J. D. Knowles. Techniques for highly multiobjective optimization: Some nondominated points are better than others. In *Proceedings of the 2007 Genetic and Evolutionary Computation Conference (GECCO '07)*, 2007.

[4] D. Dasgupta, G. Hernandez, D. Garrett, P. Vejandla, A. Kaushal, R. Yerneni, and J. Simien. A comparison of multiobjective evolutionary algorithms with informedinitialization and kuhn-munkres algorithm for the sailor assignment problem. In *Proceedings of the 2008 Genetic and Evolutionary Computation Conference (GECCO-08)*. ACM press, 2008.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multiobjective test problems. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 825–830, 2002.

[7] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[8] J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba. jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics. Technical Report ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos, December 2006.

[9] L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G. J. E. Rawlings, editor, *Foundations of Genetic Algorithms*, pages 265–283. Morgan Kaufmann, 1991.

[10] H. Feltl and G. Raidl. An improved hybrid genetic algorithm for the generalized assignment problem. In *2004 ACM Symposium on Applied Computing*, pages 990–995, 2004.

[11] C. Fleurent and J. Ferland. Genetic hybrids for the quadratic assignment problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16, Quadratic Assignment and Related Problems:173–187, 1994.

[12] J. D. Garrett, J. Vannucci, D. Dasgupta, and J. Simien. Applying hybrid multiobjective evolutionary algorithms to the sailor assignment problem. In L. Jain, V. Palade, and D. Srinivasan, editors, *Advances in Evolutionary Computing for System Design*, pages 269–301. Springer, 2007.

[13] J. D. Garrett, J. Vannucci, R. Silva, D. Dasgupta, and J. Simien. Genetic algorithms for the sailor assignment problem. In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO '05)*, 2005.

[14] A. Holder. Navy personnel planning and optimal partition. *Operations research*, 53(1):77–89, January-February 2005.

[15] H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In T. Fukuda and T. Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation*, pages 119–124, 1996.

[16] A. Jaszkiewicz. On the performance of multiple objective genetic local search on the 0/1 knapsack problem: A comparative experiment. Technical Report RA-002/2000, Institute of Computing Science, Poznan University of Technology, July 2000.

[17] J. Knowles and D. Corne. Towards Landscape Analyses to Inform the Design of Hybrid Local Search for the Multiobjective Quadratic Assignment Problem. In A. Abraham, J. R. del Solar, and M. Koppen, editors, *Soft Computing Systems: Design, Management and Applications*, pages 271–279. IOS Press, Amsterdam, 2002.

[18] J. Knowles and D. Corne. Instance generators and test suites for the multiobjective quadratic assignment problem. In C. Fonseca, P. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Proceedings Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 2003*, pages 265–283. LNCS 2632, Springer, 2003.

[19] J. D. Knowles and D. Corne. Memetic algorithms for multiobjective optimization: Issues, methods, and prospects. In N. Krasnogor, J. E. Smith, and W. E. Hart, editors, *Recent Advances in Memetic Algorithms*. Springer, 2004.

[20] J. D. Knowles and D. W. Corne. M-PAES: A memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC-2000)*, pages 325–332. IEEE Press, 2000.

[21] N. Krasnogor. Towards robust memetic algorithms. In W. E. Hart, N. Krasnogor, and J. E. Smith, editors, *Recent Advances in Memetic Algorithms*, pages 185–208. Springer, 2004.

[22] M. López-Ibáñez, L. Paquete, and T. Stützle. Hybrid population-based algorithms for the bi-objective quadratic assignment problem. Technical Report 1, FG Intellektik, FB Informatik, TU Darmstadt, 2004. Journal of Mathematical Modelling and Algorithms.

[23] P. Merz and B. Freisleben. A genetic local search approach to the quadratic assignment problem. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 465–472, 1997.

[24] P. Moscato. On evolution, search, optimization, genetic algorithms, and martial arts: Towards memetic algorithms. Technical Report 826, Caltech Concurrent Computation Program (C3P), 1989.

[25] P. Moscato. Memetic algorithms: A short introduction. In D. Corne, F. Glover, and M. Dorigo, editors, *New Ideas in Optimization*, pages 219–234. McGraw-Hill, 1999.

[26] S. Sahni and T. Gonzalez. P- complete approximation problems. *Journal of the Association of Computing Machinery*, 23:555–565, 1976.

[27] E. D. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.

[28] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report Technical Report 103, Computer Engineering and Networks Laboratory(TIK), Swiss Federal Institute of Technology (ETH), Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.