Finding a preferred diverse set of Pareto-optimal solutions for a limited number of function calls

Florian Siegmund, Amos H.C. Ng
Virtual Systems Research Center
University of Skövde
P.O. 408, 541 48 Skövde, Sweden
{florian.siegmund, amos.ng}@his.se

Kalyanmoy Deb, IEEE Fellow
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
PIN 208016, India
deb@iitk.ac.in

KanGAL Report Number 2012007

Abstract — Evolutionary Multi-objective Optimization aims at finding a diverse set of Pareto-optimal solutions whereof the decision maker can choose the solution that fits best to her or his preferences. In case of limited time (of function evaluations) for optimization this preference information may be used to speed up the search by making the algorithm focus directly on interesting areas of the objective space. The R-NSGA-II algorithm [1] uses reference points to which the search is guided specified according to the preferences of the user. In this paper, we propose an extension to R-NSGA-II that limits the Pareto-fitness to speed up the search for a limited number of function calls. It avoids to automatically select all solutions of the first front of the candidate set into the next population. In this way non-preferred Pareto-optimal solutions are not considered thereby accelerating the search process. With focusing comes the necessity to maintain diversity. In R-NSGA-II this is achieved with the help of a clustering algorithm which keeps the found solutions above a minimum distance ε. In this paper, we propose a self-adaptive ε approach that autonomously provides the decision maker with a more diverse solution set if the found Pareto-set is situated further away from a reference point. Similarly, the approach also varies the diversity inside of the Pareto-set. This helps the decision maker to get a better overview of the available solutions and supports decisions about how to adapt the reference points.

Keywords: Evolutionary multi-objective optimization; decision making; guided search, reference point, diversity, Pareto-optimal

I. INTRODUCTION

Evolutionary Multi-objective Optimization provides the decision maker with a diverse set of Pareto-optimal solutions. One of these optimal solutions must be chosen for implementation. The decision maker has domain knowledge on which she or he will base the decision which of the solutions will be implemented. Those preferences can be used to accelerate the search process that is crucial if the total number of function calls is limited. For real-world problems this is often the case since the function to optimize usually exhibits a long execution time. In many cases the decision maker initially needs to see some available alternative solutions to identify his or her preferences [2]. Optimization algorithms have been developed that allow to define and change preferences after the optimization process has started. As soon as the optimization algorithm holds preference information it can guide the search towards a preferred set of solutions. In this way the evaluation of only little interesting solutions can be avoided and the focused search algorithm will find the preferred set faster.

In order to be able to perform a focused evolutionary search the user has to articulate the preferences and they have to be transformed into a representation that can be accessed by the algorithm. Preferences can be articulated in different ways. One way is to make comparisons between known solutions and to identify which solution is more preferable than the other. This information can then be transformed into a value function that serves as a fitness function for the evolutionary algorithm [3]. Another way of preference articulation is to define a point in the objective space where the solutions in the population are attracted to. The fitness value of solutions that are nearer to this reference point is increased. This reference point method was published in [4] and adapted for evolutionary optimization in [1]. Moreover, preferences can be expressed through the specification of a reference direction along which the search is guided [5]. A preference-guided search algorithm that combines ideas of both the reference point- and reference direction methods is the light beam search approach. It was introduced in [6] and extended for evolutionary optimization in [7]. A comprehensive overview on guided search algorithms controlled by preferences is given in [8].

In this paper the R-NSGA-II algorithm [1] is examined. It makes use of reference points to guide the search and is based on NSGA-II [9]. In the present study extensions to make R-NSGA-II faster and to provide a population that features a non-linear diversity are proposed.

The R-NSGA-II algorithm [1] uses reference points to which the search is guided and which are specified according to the preferences of the user. The reference points can be defined and changed interactively, as the optimization is progressing. As the algorithm was implemented potential for improvement was discovered. The goal of this paper is to propose extensions for R-NSGA-II that will support it in two aspects:

1) Handling the limited number of function calls: Since the number of function calls is limited a focusing multi-objective evolutionary algorithm needs to be configured in a way that it finds the desired set of solutions fast. At the same time the algorithm must remain capable to find a desired set in any part of the objective space, given the case that the desired
set exists. For R-NSGA-II we have identified a speedup potential. R-NSGA-II first selects solutions based on the Pareto-fitness and then on their second fitness value, the distance to a reference point. Our experiments show that the R-NSGA-II quickly finds a diverse and converged set of solutions similar to the result of NSGA-II. After that the population moves towards the reference point requiring large search effort. In the extended R-NSGA-II, we propose compromises on Pareto-optimality and does not select all solutions from the first fronts as an automatism. Pareto-optimal solutions that are situated in a large distance from a reference point, near to the extremal solutions of a front, do not contribute to the search but instead mislead the algorithm. If R-NSGA-II can be extended with a mechanism that neglects those solutions it is expected that the algorithm can find the preferred set of solutions faster.

2) Diversity management: A focusing multi-objective evolutionary algorithm must make special provisions to keep a diverse population. R-NSGA-II employs a clustering algorithm for this purpose. It makes sure that the selection of solutions is done in a way that aims at keeping a distance $\varepsilon$ between the solutions. In this paper we propose a self-adaptive epsilon approach that provides the decision maker with a more diverse solution set if the found Pareto-set is situated further away from a reference point. This gives the decision maker a better idea of the available solutions when the search result is still distant from the reference point. The reference point can then be adapted and the search is guided into a more interesting direction. Towards the end of the optimization the Pareto-set is near to the reference point and the diversity is reduced. Due to the same reason the self-adaptive approach also varies the diversity inside of the fronts in the unified set of population and offspring. Extremal solutions on the outer ends in the fronts feature longer distances to each other than the solutions in the fronts that are closest to the reference points.

The paper is structured as follows. In Section II the R-NSGA-II algorithm is described in detail and differences to the implementation used in this paper are pointed out. In Sections III and IV the extensions for R-NSGA-II are described and discussed in detail. In Section V we conduct numerical experiments to show the effectiveness and efficiency of the algorithm extensions. To conclude the paper we summarize our findings in Section VI.

II. REFERENCE-POINT-BASED NSGA-II

For the focused search in this paper the R-NSGA-II algorithm proposed in [1] is used, and extensions are suggested. In Section A our implementation of R-NSGA-II is described. In Section B differences to the original implementation are identified.

A. Algorithm listing and implementation description

Here, we list the R-NSGA-II proposed in [1] and describe implementation details and minor modifications. R-NSGA-II is based on the elitist Non-dominated Sorting Genetic Algorithm, (NSGA-II) which was introduced in [9].

1. Select a random population and evaluate it.
2. For all solutions in the population: Calculate the distances between the objective vector and all reference points. As distance measure an achievement scalarizing function is used that calculates the maximum normalized objective distance for an individual objective between the solution and a reference point.
3. For each reference point create a ranking for all solutions in the population. Assign each solution the minimum rank that was assigned to it among all rankings for all reference points. A solution that is closest to one of the reference points receives therefore rank 1.
4. Perform non-domination sorting and partition the population into fronts.
5. Mating: Select parent solutions by tournament selection. Choose two solutions randomly. Select the one that is a member of a better front as parent. If both solutions are member of the same front select the one with a higher rank.
6. Create offspring solutions through crossover and mutation. In our implementation we create as many offspring solutions as the solutions in the population.
7. Evaluate the offspring.
8. Perform steps 2., 3., and 4. on the unified set of population and offspring solutions.
9. The selection step which chooses the solutions for the next population out of the unified set of old population and offspring. Due to its complexity it is described below, following this algorithm listing.
10. If the number of generations has reached the limit, stop. Otherwise go to step 3.

Selection step:

9. The selection step requires a clustering algorithm that partitions every front of population and offspring in a set of clusters. The clustering starts with the solutions that each are nearest to a reference point. Each of them is used as a representative for a cluster. All solutions that are in the vicinity of the representative solution are added to its cluster and are disregarded temporarily. As distance criterion it is checked if the normalized Euclidean distance between the objective vector of the solution and the reference point is below the clustering parameter $\varepsilon$. After the clusters with the nearest representative solutions have been created the clustering is continued with random representatives. A random solution is chosen out of the solutions remaining in the front after clearing out solutions for the previous clusters. All solutions in proximity to this representative are added to its cluster. The process is continued until all solutions in the front have been assigned to a cluster.

After the clustering the selection browses through the fronts, starting from the Pareto-front, and selects as many solutions among the representatives as necessary. Only
the representatives are selected into the next population and are not considered any further. If the cluster set of a front contains more clusters and representative solutions than which are required to fill the next population a selection based on the rank (Step 7.) is performed. In case the algorithm has gone through all fronts but the next population could not be filled completely the procedure starts from the first front again. This may happen because by means of the clustering there are less solutions available for selection in each front. The clustering is performed once again on the remaining solutions and the selection procedure can browse through the hierarchy of fronts again. These steps, selection and reclustering are repeated until the next population has been completely selected.

Our implementation allows that no reference point is defined. Then the algorithm works as NSGA-II. In this way the user can first let the algorithm find an initial set of solutions and choose a reference point based on these alternatives. It is also possible to delete or update reference points. A change on the set of reference points comes into effect when a new generation starts.

In those steps of the algorithm where normalization is done the objective vectors of the whole population (and offspring) are considered for calculating the objective ranges. Also all reference points are included in the calculation.

The original publication includes a possibility to define a weighting of the objectives when calculating distances [1]. This is not a weighted-sum approach but allows the decision maker to bias the multi-objective search towards certain objectives. This aspect of the algorithm has not been considered in this paper.

B. Modified implementation

The implementation listed in the last subsection, which was used for the experiments in this paper, features some modifications compared to the original publication [1].

1) Selection of complete fronts: In the original version of R-NSGA-II all solutions of the fronts that fit completely into the next population are selected automatically without any clustering. The selection procedure steps through the fronts, starting with the first front, and checks if the number of solutions is less than the number of solutions yet missing in the new population. If so, all solutions in the front are selected. Otherwise the front will be partially selected as described in Subsection 2).

2) Partially selected front: In case that a front does not fit completely into the next population the clustering is done and from the set of cluster representatives a subset is selected based on their rank (distance to the reference points). The selected subset contains at maximum the number of missing solutions in next population. Reference [1] leaves it open what happens in the case when the clustering finds less clusters than numbers of solutions missing in the next population. The implementation in subsection A continues in this case with clustering the next inferior front and selecting more solutions there, and so on. In reference [1] solutions that have been cleared out by clustering are assigned a penalty distance that makes them less favorable to be selected. It is also stated however that those solutions are not considered further.

3) Clustering: The original R-NSGA-II implementation in [1] starts the clustering with random solutions as representatives. The implementation in Section A first forms clusters around the solutions that are closest to a reference point. After that, the clustering is continued with picking random representative solutions out of the solutions that were not cleared out by clustering yet, as in [1]. Through this step we hope to achieve a slight advantage for the algorithm’s performance. This modification leads to a reduction of the probability that the solutions which are closest to a reference point are discarded in the selection step. The algorithm might start clustering with a random solution and clear out those closest solutions. The same could happen when the algorithm is going through the front hierarchy again. The loss of these solutions might be negative for the algorithm performance.

The proposed extensions for R-NSGA-II are described in the following two sections.

III. EXTENDED R-NSGA-II: SPEEDUP

During our experiments we observed that the R-NSGA-II algorithm converges quickly towards a diverse front. However, the following movement towards the reference point to find the preferred set is comparably slow. This is caused by the working principle of the algorithm. As in many other multi-objective optimization algorithms Pareto-optimal solutions are preserved, if possible. Generally, dominating solutions are preferred over dominated solutions. The solutions are selected front-wise, as the Pareto-fitness is the primary fitness criterion. First if a front cannot be selected completely a secondary fitness criterion is applied. In R-NSGA-II this working principle is slightly softened since dominating solutions might be cleared out by the clustering and dominated solutions might be preferred. (In R-NSGA-II cleared out solutions get a chance to be selected later.) Nevertheless, this prioritization of the Pareto-fitness results in a high spread of the population which usually is desirable to achieve a good diversity in the result population.

For the focused search however, the spread is not necessary to this extent. Instead, a high spread misleads and slows down the algorithm. Solutions far away from the reference point that have been found because of a high spread get an equal chance to influence coming generations through mating. Thereby the probability is increased that the algorithm maintains a diverse population that only slowly converges towards the reference points. In our experiments this influence showed to be so strong that R-NSGA-II in the beginning converges to almost the true Pareto-front, like NSGA-II, and first then starts to move towards the reference points. We therefore propose an approach that compromises on Pareto-optimality to speed up the search.

In this paper the selection operator of R-NSGA-II is modified to speed up the convergence of the optimization
algorithm towards a set of solutions near to the reference point with the desired diversity. We control the number of solutions that are selected from each individual front. In the following the population size is referred to as $N$ and the number of fronts in the population (and offspring) is denoted as $n$. In the selection step in total $N$ solutions need to be chosen out of $2N$ solutions. We need to constrain the number of solutions from the first fronts which have the highest influence on the optimization process through mating. However, it is still important to select a large part of the next population from those fronts. Otherwise a strong emphasis on solutions from inferior fronts would render the algorithm ineffective. We therefore propose a decreasing assignment of solutions for selection. The number of solutions that should be selected out of front $i$ is calculated as

$$\frac{N}{2^i}, \ i = 1, \ldots, n$$  \hspace{1cm} (1)

Since the assigned numbers might not be integer we use the floor function to calculate the number of selected solutions for each front. The total sum of selected solutions will be

$$\sum_{i=1}^{n} \left\lfloor \frac{N}{2^i} \right\rfloor \ N$$  \hspace{1cm} (2)

Ideally, without rounding and with infinite number of fronts $n$, Equation (2) would give around $N$ solutions as a sum value. This is because it contains a geometric series with limit $\frac{1/2}{1-1/2}$. However since the number of fronts is finite and since the floor function is used the sum can never reach $N$. This selection gap will be compensated in a later step.

It may happen that a front contains less solutions than the assigned budget. In this case the missing budget will be transferred to the preferred budget of the next front.

Another case that has to be taken care of is when some fronts contain many more solutions than the preferred budget. This might lead to that it is not possible to shift the budget to other fronts. In this situation our implementation goes through the fronts and adds the missing budget to the first front where yet more solutions can be selected. To emphasize the first fronts this search starts with the Pareto-optimal front. This provision also covers the selection gap caused by the floor function as described above.

IV. EXTENDED R-NSGA-II: DISTANCE-BASED DIVERSITY CONTROL

The extended R-NSGA-II features a diversity control that automatically adapts the size of the $\epsilon$ parameter according to the distance between the solutions in the set of population and offspring and the reference points.

In the original version of the algorithm [1] the population is influenced during the optimization process through a clustering algorithm towards the desired minimum distance $\epsilon$ between the solutions. The decision maker might be interested in finding a compact set of solutions near to the reference point and will choose a small $\epsilon$ value. However with the small $\epsilon$ the population might not exhibit sufficient spread in the first generations. In order to be able to make good decisions how to adapt the reference points it is necessary to present a widespread population to the decision maker in the beginning of the optimization. If the decision maker mostly sees close-by alternative solutions that are not clearly different she or he might miss other important areas in the objective space and might make suboptimal decisions where to guide the search.

Here our extension of R-NSGA-II comes in. It calculates the distances of the Pareto-front to the reference points and adapts $\epsilon$ proportional to those distances. Additionally, the distance of each solution inside of a front to the solution that is nearest to the reference point is taken into account when adapting $\epsilon$.

We have to mention that with the extended R-NSGA-II the minimum spread is not guaranteed because the algorithm allows solutions to be selected that are situated in a distance smaller than $\epsilon$. This happens when the selection procedure has finished going through all fronts and still some solutions are missing in the next population. Then it has to start browsing through the fronts again, starting with the first front, and select the missing solutions. They might be chosen out of the vicinity of previously selected solutions. This must be possible for the optimization algorithm since a widespread distribution of solutions might not even exist. If it exists, we expect that after a certain number of generations a population will be found that offers the desired spread.

Towards the end of the optimization the Pareto-set is near to the reference points and the diversity is reduced. The adaptive approach should be designed in the way that at this closest distance of the Pareto-front to a reference point the original $\epsilon$ value should be assigned to the $\epsilon$ parameter. In our approach this is achieved by a minimum $\epsilon$ limit parameter that bounds the proportional adaption of $\epsilon$. The approach also offers a maximum limit parameter for $\epsilon$, through a maximum distance the user can set. If the Pareto-set is further away from the reference point $\epsilon$ is not further increased.

Since the R-NSGA-II algorithm allows several reference points to be specified our adaptive approach will lead to a different $\epsilon$ value for each solution. In the unified set of population and offspring subsets are formed that consist of solutions that are closest to one of the reference points. An $\epsilon$ basis value for this subset is determined, dependent on the minimal distance of the set to the reference point. The solution that is nearest to the reference point is then the basis for further adaptation of $\epsilon$ for all other solutions in the set. Each solution is assigned a different $\epsilon$ value based on its distance to the solution that is nearest to the reference point. This $\epsilon$ value is then used if the solution is chosen as a representative in the cluster algorithm of the extended R-NSGA-II.

Two aspects of distance are considered: the distance between the Pareto-front and reference point and the distance between the solutions inside of a front. How our approach handles the first aspect is described in the following subsection A. The second aspect and an adaptive procedure that combines both aspects are described in subsection B.
A. $\varepsilon$ proportional to front distance

The extended R-NSGA-II algorithm adapts the diversity parameter $\varepsilon$ according to the distance between a reference point and the nearest solution in the Pareto-set. In the unified set of population and offspring subsets are formed. Every subset contains all solutions that are closest to one specific reference point. One of those solutions is situated nearest to the reference point in the objective space. Thereby it determines through its distance the epsilon parameter $\varepsilon_i$ for this subset in the front.

We introduce the following notation:

$r_i$ Reference point $i$

$N$ Population size

$2\beta N$ Number of non-dominated solutions in the set of population and offspring

$K$ Number of reference points

$s_i$ Desired spread of the solutions assigned to $r_i$

On average, the number of solutions that are assigned to $r_i$ are

$$\frac{2\beta N}{K}. \tag{3}$$

To calculate $\varepsilon_i$, we equally divide the spread for $r_i$ among the solutions assigned to $r_i$ and choose $\varepsilon_i$ proportional to this value:

$$\varepsilon_i \propto \frac{s_i}{(2\beta N / K) - 1} = \frac{s_i K}{2\beta N} \tag{4}$$

We would want that the spread $s_i$ is proportional to the minimum distance between $r_i$ and the solutions that are assigned to it.

$$s_i = \alpha d_i \tag{5}$$

As distance $d_i$ we choose the Euclidean distance between $r_i$ and the closest solution out of the solutions assigned to $r_i$. We choose $\alpha = \beta$.

$$\varepsilon_i \propto \frac{\alpha d_i K}{2\beta N} = \frac{d_i K}{2N} \tag{6}$$

If the Pareto-set is close to $r_i$ the adapted $\varepsilon_i$ will be small. To make sure that $\varepsilon_i$ cannot get smaller than the $\varepsilon$ parameter the user would have specified in the original R-NSGA-II we introduce a lower bound for $\varepsilon_i$ as parameter, which we denote $\underline{\varepsilon}$. $\varepsilon_i$ is also upper bounded by a parameter $\overline{d}$ that limits the distance value $d_i$ to avoid unreasonably high distances between the solutions. The possible values for $\varepsilon_i$ are therefore included in the interval

$$\varepsilon_i \in \left[ \underline{\varepsilon}, \frac{\overline{d} K}{2N} \right] \tag{7}$$

For the distance between the solutions and the reference points the normalized Euclidean distance is used. The normalization includes all solutions in population and offspring and all reference points.

B. Horizontal adaption of $\varepsilon$

In addition to a diversity control that is dependent on the distance of the Pareto-front to the reference points a variation of $\varepsilon$ inside of the fronts is introduced. This approach can however be employed independently from the approach in A. By the same argument as before we want to provide the decision maker with a compact crowded set of solutions near to the reference points and a more spread set of solutions towards the outer ends of the fronts.

For this purpose subsets are formed inside of the set of population and offspring based on the reference point that the solutions are closest to. We then calculate the distance of each solution to the solution that is nearest to the cluster’s reference point $r_i$. Each solution $j$ is assigned its own clustering parameter $\varepsilon_{ij}$ which dependent on this distance. We define:

$$\delta_j$$ Distance of solution $j$ to the solution in its cluster that is nearest to $r_i$

$$\overline{\delta}_j$$ $\delta_j$ normalized by all solutions assigned to $r_i$

We make $\varepsilon_{ij}$ dependent on $\overline{\delta}_j$ and the $\varepsilon$ parameter specified by the user via a parameter $k$ as in (8).

$$\varepsilon_{ij} = \left( 1 + k \overline{\delta}_j \right) \varepsilon \tag{8}$$

For the distance between the solutions the normalized Euclidean distance is used. The normalization includes all solutions in population and offspring and all reference points.

It is possible to combine both aspects of $\varepsilon$ adaption in one adaption procedure. Thereto first the $\varepsilon_i$ value has to be calculated for each reference point and subsequently the $\varepsilon_{ij}$ values can be calculated for each solution as in (8).

V. NUMERICAL EXPERIMENTS

In this section the experiments carried out with the extended R-NSGA-II algorithm are presented. They prove the efficiency of the speedup. The performance of the algorithm
on several benchmark functions is assessed. The functions are chosen out of the ZDT benchmark suite [10] and the DTLZ benchmark suite [11].

For the conducted experiments the extended R-NSGA-II has been configured as follows. Population size and offspring size are 50. Blend crossover is used with a crossover probability of 0.8. Mutation probability is $5/(\text{number of parameters})$.

A. ZDT1

The ZDT1 function is a two dimensional benchmark function with 30 input parameters. Both conflicting objectives shall be minimized. Its Pareto-front can be analytically described and is convex.

We investigate the performance of R-NSGA-II given a limited number of function calls: The original publication of R-NSGA-II [1] did examine its optimization process under the precondition that no computation budget limitation needs to be considered. It shows the result population after 50000 evaluations which is a high number when compared to computation time and budgets available in real-world problems where costly simulation models have to be run for each function call. In this paper it is therefore investigated how the R-NSGA-II behaves given a limited budget of function calls.

a) General performance: The following experiment examines the performance of the implementation used in this paper on ZDT1 given a limited number of function calls of 5000. Several optimization runs with different $\varepsilon$ values are performed ($\varepsilon=0.0001, 0.001, 0.005, 0.01$). Figure 1 shows the results for a reference point of (0.5, 0). In addition the result of NSGA-II is given for comparison purposes which represents the case when no reference point is defined for R-NSGA-II.

The results show that all runs have converged to the optimal Pareto-front. NSGA-II achieves a complete spread along the Pareto-front. The spread of the result sets of the different runs increases with a bigger $\varepsilon$. The implementation behaves as expected. The results show that R-NSGA-II converges to the optimal front and delivers a focused set even with a low budget of 5000 evaluations. If compared with the results after 50000 evaluations [1] the achieved spread is not as good. More evaluations have to be evaluated. We find that the algorithm works in two phases. First, the convergence to the Pareto-front which is completed after only few evaluations and then the second phase where the focused set is narrowed to the preferred set.

b) Speedup: In this experiment it is investigated if by the constrained Pareto-fitness approach proposed in section III the time can be reduced that is needed to find the preferred set. The extended R-NSGA-II is run on ZDT1 with 10000 evaluations with an $\varepsilon$ value of 0.001 and reference point (0.5, 0). The results are displayed in Figure 2.

After certain time steps the current non-dominated front is recorded. In the development of the front in Figure 2 it can be seen that solutions in the preferred set have been found after 5000 evaluations. After this point the found set only gets narrower to finally reach the end width. If we compare with [1] this width is not reached after 10000 evaluations.
If we compare the results with the results of the original R-NSGA-II in the same situation as shown in Figure 3 we see a similar behavior except for that the fronts here are wider and contain more solutions.

To find a difference in the behavior the development of the populations before 2000 evaluations have to be examined. This is done in Figure 4 and Figure 5. Here it can be seen that the extended R-NSGA-II moves its fronts earlier towards the reference point whereas the version without constrained Pareto-fitness explores first the left part of the Pareto-front before it moves to the center and the reference point. The latter behavior was observed often during our experiments. This is due to the fact that R-NSGA-II favors non-dominated solutions over solutions close to the reference point. This leads to that the population first moves towards the area where both objectives are minimal and then the search continues along the front.

This can be observed in another type of diagram in Figure 6 and Figure 7. The constrained version moves faster towards the preferred set which has an average x1 value of around 0.55. This is due to the tilted Pareto-front which causes that the solutions nearest to the reference point are situated slightly to the right of it.

For the constrained Pareto-fitness the point where no big decrease in the distance of new found solutions occurs is reached at about 2000 evaluations (Figure 8). The original version of R-NSGA-II reaches this point at 4500 evaluations. This is a speedup of 2.5. The results show that the extended R-NSGA-II has advantages in the range below 2000 evaluations. There the algorithms are diverted towards low values of x1 due to their prioritized use of the Pareto-fitness. The extended version can avoid this to a certain extent by not keeping all Pareto-optimal solutions in the population. This causes less attraction to low x1 values.
For future work we have the following ideas:

- Find an optimal extent of constraining the Pareto-optimality. Run experiments on problems with different numbers of objectives.
- Give solutions near to the reference point a higher chance to be selected for mating. Currently, in the tournament selection, solutions with a large distance to a reference point, yet Pareto-optimal, are preferred over solutions close to the reference point.
- In every generation present the current non-dominated front to the decision maker, including the Pareto-optimal solutions that were discarded in the previous generation due to constrained Pareto-optimality. In this way, with the extended R-NSGA-II, the decision maker can observe the development of non-dominated fronts that contain as many solutions as the non-dominated fronts of R-NSGA-II.

ACKNOWLEDGMENT

This study was partially funded by VINNOVA, Sweden, through the FFI-HSO project and Volvo Car Corporation. The authors gratefully acknowledge their provision of research funding.

REFERENCES