Hybrid Dynamic Resampling for Guided Evolutionary Multi-Objective Optimization

Florian Siegmund[1], Amos H.C. Ng[1], and Kalyanmoy Deb[2]

[1] Virtual Systems Research Center, University of Skövde, Sweden
{florian.siegmund,amos.ng}@his.se
[2] Department of Electrical and Computer Engineering,
Michigan State University, East Lansing, USA
kdeb@egr.msu.edu

COIN Report Number 2014017

Abstract. In Guided Evolutionary Multi-objective Optimization the goal is to find a diverse, but locally focused non-dominated front in a decision maker's area of interest, as close as possible to the true Pareto-front. The optimization can focus its efforts towards the preferred area and achieve a better result [9, 17, 7, 13]. The modeled and simulated systems are often stochastic and a common method to handle the objective noise is Resampling. The given preference information allows to define better resampling strategies which further improve the optimization result. In this paper, resampling strategies are proposed that base the sampling allocation on multiple factors, and thereby combine multiple resampling strategies proposed by the authors in [15]. These factors are, for example, the Pareto-rank of a solution and its distance to the decision maker's area of interest. The proposed hybrid Dynamic Resampling Strategy DR2 is evaluated on the Reference point-guided NSGA-II optimization algorithm (R-NSGA-II) [9].

Keywords: evolutionary multi-objective optimization, guided search, reference point, dynamic resampling, budget allocation

1 Introduction

In Guided Evolutionary Multi-objective Optimization the decision maker is looking for a diverse, but locally focused non-dominated front in a preferred area of the objective space, as close as possible to the true Pareto-front. As solutions found outside of the area of interest are considered less important or even irrelevant, the optimization can focus its efforts towards the preferred area and find the solutions that the decision maker was looking for in a faster way, i.e. with less simulation runs. This is particularly important if the available time for optimization is limited, as for many real-world applications. Focusing the search effort sets time resources free which, if not needed elsewhere, can be used to achieve a better result. Multi-objective evolutionary algorithms that can perform guided search with preference information are, for example, the R-NSGA-II algorithm [9], Visual Steering [17], and interactive EMO based on progressively approximated value functions [7].

In Simulation-based Optimization the modeled and simulated systems are often stochastic. To obtain an as exact as possible simulation of the system behavior the stochastic characteristics are often built into the simulation models. When running the stochastic simulation this expresses itself in deviating result values. That means that if the simulation is run multiple times for a selected parameter setting the result value is slightly different for each simulation run. In the literature this phenomenon of stochastic evaluation functions is sometimes called Noise, respectively Noisy Optimization [1, 3].

If an evolutionary optimization algorithm is run without countermeasure on an optimization problem with a noisy evaluation function the performance will degrade in comparison with the case if the true mean objective values would be known. The algorithm will have wrong knowledge about the solutions' quality. Two cases of misjudgment will occur. The algorithm will see bad solutions as good and select them into the next generation. Good solutions might be assessed as inferior and might be discarded. The performance can therefore be improved by increasing the knowledge of the algorithm about the solution quality.

Resampling is a way to reduce the uncertainty of the knowledge the algorithm has about the solutions. Resampling algorithms evaluate solutions several times to obtain an approximation of the expected objective values. This allows EMO algorithms to make better selection decisions, but it comes with a cost. As the modeled systems are usually complex they require long simulation times, which limits the number of available solution evaluations. The additional solution evaluations needed to increase objective value knowledge are therefore not available for exploration of the objective space. This exploration vs. exploitation trade-off can be optimized, since the required knowledge about objective values varies between solutions. For example, in a dense, converged population it is important to know the objective values well, whereas an algorithm which is about to explore the objective space is not harmed much by noisy objective values. Therefore, a resampling strategy which samples the solution carefully according to their resampling need, can help an EMO algorithm to achieve better results than a static resampling allocation. Such a strategy is called Dynamic Resampling. This has been done previously for single-objective optimization problems by [11] and [4]. In this paper we study dynamic resampling algorithms that can handle multi-objective evaluation functions, based on a previous study [15].

The paper is structured as follows. In Section 2 background information to Dynamic Resampling and an introduction to the R-NSGA-II algorithm is given. In Section 3 different resampling techniques for EMO and for guided EMO are explained. A new resampling algorithm is proposed combining several resampling techniques in Section 4. In Section 5 numerical experiments on benchmark functions are performed. The test environment is explained and the experiment results are analyzed. In Section 6 conclusions are drawn and possible future work is pointed out.

2 Background

In this section, background information is given regarding Resampling as a noise handling method in Evolutionary Multi-objective Optimization and the preference-based multi-objective optimization algorithm R-NSGA-II [9], which are the basis for the proposed algorithms in this paper.

2.1 Noise Compensation via Sequential Dynamic Resampling

To be able to assess the quality of a solution according to a stochastic evaluation function statistical measures like sample mean and sample standard deviation can be used. By executing the simulation model multiple times a more accurate value of the solution quality can be obtained. This process is called Resampling. We denote the sample mean value of objective function F_i for solution s as follows: $\mu_n(F_i(s)) = \frac{1}{n}\sum_{j=1}^{n} F_i^j(s)$, where $F_i^j(s)$ is the j-th sample of s, and the sample variance of objective function i: $\sigma_n^2(F_i(s)) = \frac{1}{n-1}\sum_{j=1}^{n}(F_i^j(s) - \mu_n(F_i(s)))^2$.

The general goal of resampling a stochastic objective function is to reduce the standard deviation of the mean of an objective value $\sigma(\mu(F_i(s)))$ which increases the knowledge about the objective value. With only a limited number of samples available the standard deviation of the mean can be estimated by the sample standard deviation of the mean which usually is called standard error of the mean. It is calculated as follows:

$$se_n(\mu_n(F_i(s))) = \frac{\sigma_n(F_i(s))}{\sqrt{n}}$$

By increasing the number of samples n of $F_i(s)$ the standard deviation of the mean and its estimate $se_n(\mu_n(F_i(s)))$ is reduced.

Dynamic resampling allocates a different sampling budget to each solution based on the evaluation characteristics of the solution. A basic dynamic resampling procedure would be to reduce the standard error until it gets below a certain threshold $se_n(\mu_n(F(s))) < se_{thr}$. The required sampling budget for the reduction can be calculated as $n > \left(\frac{\sigma_n(F(s))}{se_{thr}}\right)^2$. However, since the sample mean changes as new samples are added this one-shot sampling allocation might not be optimal. The number of fitness samples drawn might be too small for reaching the error threshold, in case the sample mean has shown to be larger than the initial estimate. On the other hand, a one-shot strategy might add too many samples if the initial estimate of the sample mean was too big. Therefore dynamic resampling is often done sequentially. Through this sequential approach the number of required samples can be determined more accurately. For Sequential Dynamic Resampling often the shorter term Sequential Sampling is used. Dynamic resampling techniques can therefore be classified in one-shot sampling strategies and sequential sampling strategies.

2.2 Reference point-guided NSGA-II

The resampling techniques described in this paper will be tested how well they can support the Reference point-Guided NSGA-II algorithm (R-NSGA-II) [9] as an example for a guided Evolutionary Multi-objective Optimization (EMO) Algorithm. It is particularly suitable for evaluation in this paper since it uses fitness functions which are used as resampling criteria in resampling algorithms. Therefore, the resampling algorithms can support R-NSGA-II particularly well.

R-NSGA-II is based on the Non-dominated Sorting Genetic Algorithm II [6] which is a widely-used and representative multi-objective evolutionary algorithm. NSGA-II sorts the solutions in population and offspring into different non-dominated fronts. Selected are all solutions in all fronts that fit into the next population. From the front that only fits partially those solutions are selected into the next population that have big distances to their neighbors and thereby guarantee that the result population will be diverse. As diversity measure the crowding distance is used. After selection is completed offspring solutions are generated by tournament selection, crossover, and mutation. The offspring are evaluated and the selection step is performed again. The R-NSGA-II algorithm replaces the crowding distance operator by the distance to reference points. Solutions that are closer to a reference point get a higher selection priority. The reference points are defined by the user in areas that are interesting and where solutions shall be found. As a diversity preservation mechanism R-NSGA-II uses clustering. The reference points can be created, adapted or deleted interactively during the optimization run. Since R-NSGA-II uses non-domination sorting it has a tendency to prioritize population diversity before convergence to the reference points. Therefore, extensions have been proposed which limit the influence of the Pareto-dominance and allow the algorithm to focus faster towards the reference points [14].

3 Resampling Algorithms

In this chapter several resampling algorithms are described which are used in this study to support the R-NSGA-II algorithm at stochastic simulation optimization problems. We denote: Sampling budget for solution s: b_s, minimum and maximum number of samples for an individual solution: b_{min} and b_{max}, acceleration parameter for the increase of the sampling budget: $a > 0$. Increasing a decreases the acceleration of the sampling budget, decreasing a increases the acceleration. The calculated normalized sampling need x_s is discretised in the following way which guarantees that b_{max} is assigned already for $x_s < 1$: $b_s = \min\{b_{max}, \lfloor x_s(b_{max} - b_{min} + 1)\rfloor + b_{min}\}$

3.1 Generally applicable resampling techniques

This section contains resampling strategies that can be used in all multi-objective optimization problems regardless of preference information is given by a decision maker or not. They can support the goal of finding the whole Pareto-front as well as the goal of exploring a limited, preferred area in the objective space.

Static Resampling samples each solution the same amount of times. The sampling budget is constant, $b_s = b_{min} = b_{max}$. If the objective noise is high enough this technique can lead to an improvement of the optimization result for $b_s > 1$. However, since many samples are wasted on less important solutions, this technique is inferior to more advanced resampling techniques. The reason for its popularity is the low implementation effort it requires.

Time-based Dynamic Resampling allocates a small sampling budget in the beginning of the optimization and a high sampling budget towards the end of the optimization [15]. The strategy of this resampling technique is to support the algorithm when the solutions in the population are close to the Pareto-front and to save sampling budget in the beginning of the optimization when the solutions are still far away from the Pareto-front.

Time-based Resampling is a dynamic resampling technique that is not considering variance and a one-shot allocation. We denote: B = maximum overall number of simulation runs, B_t = current overall number of simulation runs. The normalized time-based resampling need x_s^T is calculated as in Equation 1.

$$x_s^T = \left(\frac{B_t}{B}\right)^a \tag{1}$$

Rank-based Dynamic Resampling assigns more samples to solutions in the first few fronts and less samples to solutions in the last fronts, to save evaluations [15]. In a well-converged population most solutions will have Pareto-rank 1 and get the maximum number of samples. This technique is not effective in many-objective optimization, since there most solutions are non-dominated. Rank-based Resampling performs sequential sampling and is a comparative resampling technique. We denote: S = solution set of current population and offspring, R_s = Pareto-rank of solution s in S, $R_{max} = \max_{s \in S} R_s$. The normalized rank-based resampling need x_s^R is calculated as in Equation 2.

$$x_s^R = 1 - \left(\frac{R_s - 1}{R_{max} - 1}\right)^a \tag{2}$$

We propose a modification of Rank-based Resampling that allocates additional samples only to the first n fronts. This allows to concentrate the additional samples on the first fronts where they can be more beneficial. The allocation function of MaxN-Rank-based Resampling is $x_s^{Rn} = 1 - \left(\frac{\min\{n, R_s\} - 1}{\min\{n, R_{max}\} - 1}\right)^a$.

We propose a Hybrid Dynamic Resampling algorithm: Rank-based Resampling can be combined with Time-based Resampling to avoid allocating samples in the beginning of the optimization where the optimization algorithm has only little gain of knowing the accurate objective values. Similar to a logical conjunction x_s^R and x_s^T can be combined to form the Rank-Time-based Resampling allocation x_s^{RT} for solution s as in Equation 3.

$$x_s^{RT} = \min\{x_s^T, x_s^R\} \tag{3}$$

3.2 Preference-based resampling techniques

In this section two resampling algorithms are described that use the preference information given by a decision maker in form of a reference point r in the objective space for the R-NSGA-II algorithm [9].

Progress-based Dynamic Resampling allocates samples to solutions depending on the progress of the population towards a reference point r. If the population is more converged better knowledge of the objective values is required. The progress is defined as the average distance from the population members to r. Since the progress in Evolutionary Multi-objective optimization can be fluctuating the average progress $\overline{P}$ from the last n populations is used. Progress-based Resampling is a sequential sampling algorithm. We denote: P_{max} = the maximum progress threshold. If $\overline{P} > P_{max}$ then b_{min} is allocated. If a progress measurement is negative the absolute value of the progress will be used, multiplied by a penalty factor. This method also works for the case of a feasible reference point, where a population moving away from r is a wanted behavior. At convergence the absolute progress value becomes smaller and smaller, leading to higher sampling allocations. The normalized progress-based resampling need x_s^P is calculated as in Equation 4.

$$x_s^P = 1 - \left(\frac{\min\{\overline{P}, P_{max}\}}{P_{max}} \right)^a \tag{4}$$

The disadvantage of Progress-based resampling is that all solutions in the population are assigned the same budget. This means that solutions, population or offspring, which are dominated by many solutions or distant to r and thereby less relevant, will be assigned an unnecessary high number of samples. They might be discarded in the next selection step of the evolutionary algorithm which reduces the benefit of the assigned samples even more. Progress-based resampling without using the distance information to the reference point is often of little use. As soon as the optimization gets stuck in a local optimum far from r, too many samples are wasted, in a situation where less samples and more uncertainty would actually help to escape the local optimum. Progress-based Resampling has been evaluated in [16].

Distance-based Dynamic Resampling (DDR) The Hybrid Dynamic Resampling strategy Distance-based Dynamic Resampling DDR was proposed in [15]. A summary is given in the following. This paper focuses on the more important case of infeasible reference points, and only the description for this case is given. Infeasible reference points are outside the feasible objective space and cannot be attained by the optimization. Distance-based Resampling requires the use of the factors of progress and time. In the case of an infeasible reference point r the solution at the minimum possible distance to r should be assigned b_{max}, otherwise the decision maker would not be guaranteed full control over the sampling allocation. The time factor is required because in case the optimization gets stuck at an early point in time in a local optimum with a low progress, then no samples should be wasted during this stage. DDR is a sequential sampling algorithm.

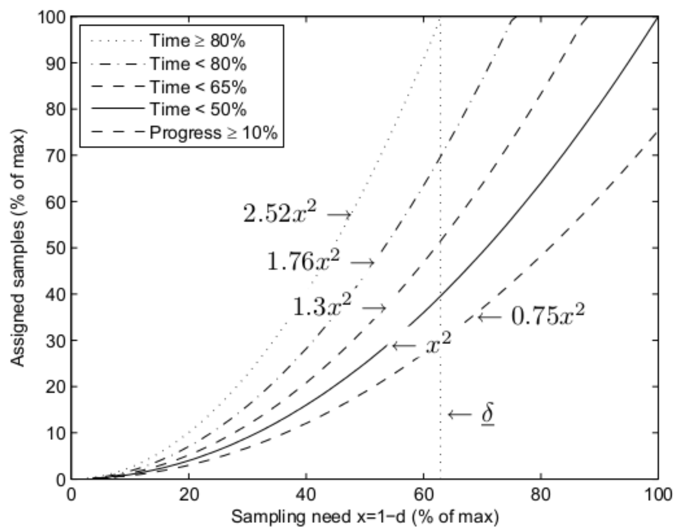

Fig. 1. Sampling allocation of the Distance-based Dynamic Resampling (DDR) algorithm [15]. Acceleration parameter $a = 2$.

If d_s is the distance of solution s to r then the sampling budget is assigned according to $x_s^{DDR} = (1-d_s)^a$. Distance-based Dynamic Resampling guarantees b_{max} samples for a certain percentage of the solutions in the current population, depending on the average progress $\overline{P}$ in the population. If $10\% > \overline{P} \geq 5\%$ then only the best (hypothetical) solution that is closest to r is allocated b_{max}. If $5\% > \overline{P} \geq 2.5\%$ then the 10% best solutions are allocated b_{max}. $2.5\% > \overline{P} \geq 1\%$ corresponds to 20% of the solutions and $\overline{P} < 1\%$ to 40%. For this purpose the maximum distance $\underline{\delta}$ to r of 40% of the population is calculated and the allocation function is increased by the factor $1/(1-\underline{\delta})$ (cf. Equation 5). If $\overline{P} > 10\%$ then the closest (hypothetical) solutions are allocated less than b_{max} to be prepared in case r is feasible and b_{max} should only be allocated to solutions dominating r. As mentioned above, the time criterion is used to slow down the allocation. In several steps during the optimization runtime the slowing effect is reduced. This is shown in Figure 1.

$$x_s^{DDR} = \min\left\{1, \left(\frac{1-d_s}{1-\underline{\delta}}\right)^a\right\} \tag{5}$$

4 Distance-Rank Dynamic Resampling (DR2)

As an attempt to use several different resampling criteria in a resampling algorithm and to create a hybrid resampling strategy for guided EMO, we propose to combine the Rank-based Dynamic Resampling and the Distance-based Dynamic Resampling (DDR) strategies described previously in Section 3. We call it Distance-Rank Dynamic Resampling (DR2). It uses four different factors to determine the resampling allocation for individual solutions: Pareto-rank, time, reference point distance, and progress. Whereas the elapsed optimization time can be combined with any other resampling criterion with little effort, as seen at Rank-Time-based Dynamic Resampling mentioned above, the Pareto-rank and the reference point distance are two truly different resampling criteria. Therefore we emphasize the term Hybrid for the DR2 algorithm in this paper.

Equation 6 describes the sampling allocation of DR2. Similarly to Rank-Time-based Resampling, the minimum of both the normalized sampling need of Distance-based Resampling and Rank-based Resampling is used to create a combined sampling allocation. However, the Distance-based sampling need is not calculated individually for each solution. Instead, DR2 identifies the solution s_m closest to r and the normalized sampling need for x_m^{DDR} is used in the formula as fixed value for all solutions in the current generation. This way of combining rank and distance information has shown the best optimization results.

$$x_s^{DR2} = \min\left\{x_m^{DDR}, x_s^R\right\} \tag{6}$$

5 Numerical Experiments

In this section the described resampling techniques in combination with R-NSGA-II are evaluated on two benchmark functions.Stepwise, more and more advanced techniques using multiple resampling criteria are compared in different configurations, showing superior results. To facilitate comparison, the experiments are grouped in experiments where no preference information is used for resampling and experiments where the resampling algorithm uses information about the distance to a reference point r defined for R-NSGA-II. For reasons of simplicity only experiments with one reference point are run, even though R-NSGA-II is capable of guiding multiple sub-populations to different reference points in the objective space. Also, the reference point is chosen to be infeasible in order to keep R-NSGA-II and the preference-based resampling algorithms as simple as possible. The combinations of R-NSGA-II with different resampling strategies are tested on two bi-objective benchmark functions ZDT1 and ZDT4 [18]. The ZDT1 function is used for evaluation due to its popularity in the literature. ZDT4 is more difficult to solve and features many local Pareto-fronts. This allows a more detailed analysis of the algorithm behavior. The ZDT benchmark functions are deterministic in their original version. In order to create noisy problems, a zero-mean normal distribution is added on both objective functions.

5.1 Problem settings

The used benchmark functions are deterministic in their original version. Therefore zero-mean normal noise has been added to create noisy optimization problems. The ZDT1 objective functions are for example defined as $f_1(x) = x_1 + \mathcal{N}(0, \sigma_1)$ and $f_2(x) = 1 - \sqrt{x_1/(1 + 9\sum_{i=2}^{30} x_i/29)} + \mathcal{N}(0, \sigma_2)$. For the ZDT1 and ZDT4 functions the two objectives have different scales. Therefore the question arises if the added noise should be normalized according to the objective scales. We consider the case of noise strength relative to the objective scale as realistic which can occur in real-world problems, and therefore this type of noise is evaluated in this paper. For the ZDT1 function the relative added noise (5%) is $(\mathcal{N}(0, 0.05), \mathcal{N}(0, 0.5))$ (considering the relevant objective ranges of $[0, 1] \times [0, 10]$), and for ZDT4 it is $(\mathcal{N}(0, 0.05), \mathcal{N}(0, 5))$ (relevant objective ranges $[0, 1] \times [0, 100]$). In the following these problems are called ZDT1-(0.05,0.5) and ZDT4-(0.05, 5).

5.2 Algorithm parameters

The limited simulation budget is chosen as 2000 solution replications for ZDT1 and 5000 replications for ZDT4. This corresponds to a 1 day optimization runtime on a cluster with 50 computers and a 15 minutes function evaluation time, which could be a realistic real-world optimization scenario. R-NSGA-II is run with a crossover rate $p_c = 0.8$, SBX crossover operator with $\eta_c = 2$, Mutation probability $p_m = 0.07$ and Polynomial Mutation operator with $\eta_m = 5$. The Epsilon clustering parameter is chosen as $\epsilon = 0.001$. This corresponds to a 1 day optimization runtime on a cluster with 50 computers and a 15 minutes function evaluation time, which could be a realistic real-world optimization scenario. R-NSGA-II is run with a crossover rate $p_c = 0.8$, SBX crossover operator with $\eta_c = 2$, Mutation probability $p_m = 0.07$ and Polynomial Mutation operator with $\eta_m = 5$. The Epsilon clustering parameter is chosen as $\epsilon = 0.001$. For ZDT1 and ZDT4 the reference point $r = (0.05, 0.5)$ is used which is close to the Ideal point (0, 0).

Since there is no perfect parameter configuration for Dynamic Resampling algorithms that works well on all optimization problems, we chose one configuration and did not do any parameter optimization. We chose the parameter values that seemed most intuitive to us and used them for all experiments. For all resampling algorithms the minimum budget to be allocated is $b_{min} = 1$ and the maximum budget is $b_{max} = 5$. Static Resampling is run in two configurations with $b_s = 1$ and 2. Time-based Resampling uses uses a linear allocation, $a = 1$. Rank-based Resampling and Rank-Time-based Resampling are run as Max5 Rank-based Resampling and use linear allocation ($a = 1$) for both the rank-based and time-based criteria. Progress-based Dynamic Resampling is not evaluated due to the described disadvantages. Distance-Progress-Time-based Dynamic Resampling DDR uses delayed ($a = 2$) distance-based allocation. Distance-Rank-based Dynamic Resampling DR2 uses the same parameters as the constituting resampling techniques.

5.3 Evaluation, replication, and interpolation

In order to obtain a reliable performance measurement the accurate objective values for each evaluated solution are used. For the benchmark problems, either the added noise landscape can be removed and the solution is evaluated deterministically, or the solution can be sampled a high number of times to obtain accurate objective values, which was done in this study. 2500 samples on a benchmark problem solution reduce the uncertainty of the objective values by a factor of 50.

All experiments performed in this study are replicated 10 times and mean performance metric values are calculated. To be able to see the performance development over time a performance metric is evaluated after every generation of the optimization algorithm. This is shown in Figure 2. However, due to the resampling, each generation uses a different number of solution evaluations. Since it is assumed that solution evaluations are equally long and that the runtime of the optimization algorithm and resampling algorithms is negligible compared to the evaluation time, the number of solution evaluations corresponds to the optimization runtime. Therefore, an interpolation is required which calculates the performance metric values at equidistant evaluation number intervals where the mean performance measure values for all experiment replications can be calculated. Not only differ the number of solution evaluations per generation (measurement points) between experiment replications, but also between different experiments with different resampling algorithms of the same optimization problem, which shall be compared.

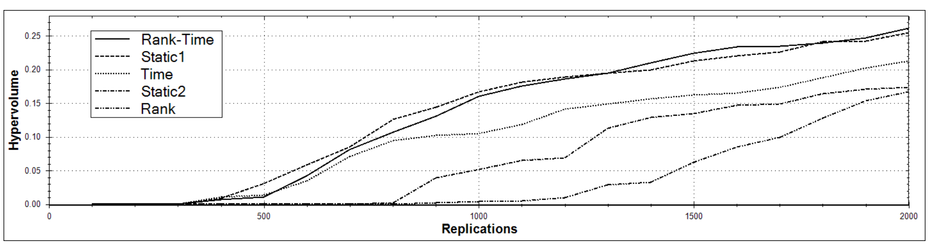

Fig. 2. Focused Hypervolume chart showing the R-NSGA-II progress development over time on ZDT1-(0.05,0.5) for resampling methods that do not rely on preference information. Reference point (0.05, 0.5).

5.4 Focused Hypervolume

To measure and compare the results of the different resampling algorithms together with R-NSGA-II the Focused Hypervolume performance metric for ϵ-dominance based EMOs (F-HV) is used [16]. The F-HV allows to measure the convergence and diversity of a population limited to a preferred area in the objective space. The limits are defined based on the intended diversity and the population size of the optimization algorithm. This allows to measure to which degree the optimization algorithm can achieve the intended diversity. For the R-NSGA-II algorithm the intended diversity is controlled by the user parameter epsilon. F-HV is based on the Hypervolume metric (HV) [19]. In F-HV the population is filtered before it is judged by the HV. The filter is a cylindrical subspace of the objective space retaining only solutions close to the reference point r dominating the HV-reference point. The cylinder axis is defined by r and a second point determining the direction, approximately orthogonal to the potentially not completely known true Pareto-front. For R-NSGA-II and a bi-objective problem, solutions within the distance $d = \epsilon \frac{N}{2}$ from the cylinder axis, with N being the population size, are passed on to the standard HV, the rest is discarded. In this way, within the cylinder there is enough space for N non-dominated solutions, each with the distance ϵ to its neighbors. The cylinder filter must be applied before the non-domination sorting is performed. Otherwise, dominated solutions are filtered out during non-domination sorting which would be non-dominated after the application of the cylinder filter. The F-HV is similar to the R-Metric (R-HV) [8], which is a metric to assess the quality of converged, focused populations on the Pareto-front. It filters the solutions with a box around a representative solution of the population and then projects the remaining solutions on an axis defined by r and HV reference point. Due to the limited optimization time in our experiments, the population will never fully converge towards the Pareto-front, or r. Therefore, the R-HV representative point filter and the shifting operation cannot be applied. Instead, the F-HV is used which filters the solutions close to r, regardless of the population position in the objective space. This can lead to zero metric values in the beginning of the optimization.

For ZDT1 with the reference point $r = (0.05, 0.5)$ the HV reference point is chosen as (0.1, 1.1) and a base point for normalization as (0, 0.68). For the ZDT4 function with $r = (0.05, 0.5)$ the HV reference point is chosen as (0.1, 30) and a base point for normalization as (0, 0).

In all cases, the population size 50 together with R-NSGA-II epsilon 0.001 leads to a cylinder diameter of 0.05. The cylinder axis is defined by r and the direction point. For ZDT1 and this direction point is defined as (0.06, 1.1) and for ZDT4 as (0.06, 30).

5.5 Resampling without preference information

In this section the resampling algorithms from Section 3.1 are evaluated and compared: Static Resampling, Time-based Dynamic Resampling, Rank-based and Rank-Time-based Dynamic Resampling.

In Figure 2 the results of the different resampling techniques together with R-NSGA-II are evaluated on the ZDT1-(0.05,0.5) problem with reference point (0.05, 0.5) and 2000 function evaluations. The results show that Static Resampling with 1 sample is both better than Time-based and Rank-based Dynamic Resampling. Static2-Resampling is worse than Static1-Resampling, which shows that Dynamic Resampling is required to achieve a performance gain over the

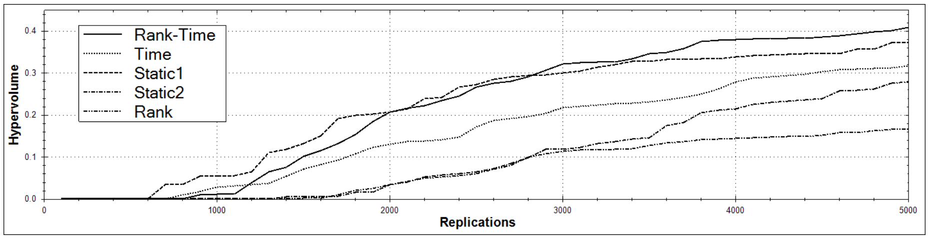

Fig. 3. Focused Hypervolume chart showing the R-NSGA-II progress development over time on ZDT4-(0.05,5) for resampling methods that do not rely on preference information. Reference point (0.05, 0.5).

Static1 strategy. This is achieved by the hybrid strategy Rank-Time-based Resampling which outperforms all others.

In Figure 3 the results of the different resampling techniques together with R-NSGA-II are evaluated on the ZDT4-(0.05,5) problem with reference point (0.05, 0.5) and 5000 function evaluations. The results confirm the results shown in Figure 2, however, the differences between the different resampling algorithms become more clear, since the ZDT4-(0.05,5) problem is more difficult (many local Pareto-fronts) and needs more time to converge to the reference point.

5.6 Resampling with reference points

In this section the resampling algorithms from Section 3.2 are evaluated and compared: Distance-Progress-Time-based Dynamic Resampling DDR, and Distance-Rank(-Progress-Time)-based Dynamic Resampling DR2. The results for Rank-Time-based Resampling are included for comparison purposes.

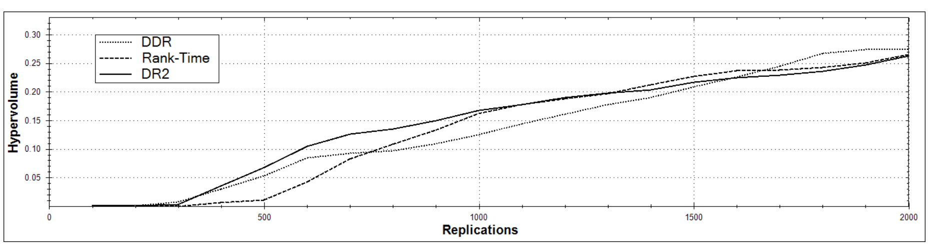

Fig. 4. Focused Hypervolume chart showing the R-NSGA-II progress development over time on ZDT1-(0.05,0.5) for resampling methods that use preference information. Reference point (0.05, 0.5). For comparison with the non-preference methods, the curve for Rank-Time-based resampling from Figure 2 is included.

In Figure 4 the results of the different resampling techniques together with R-NSGA-II are evaluated on the ZDT1-(0.05,0.5) problem with reference point

(0.05, 0.5) and 2000 function evaluations. The results show that DDR is slightly better than Rank-Time-based Resampling. However, DR2 performs slightly worse than Rank-Time-based Resampling. As a reason we can see that the ZDT1-(0.05,0.5) problem is not sufficiently complex (short convergence time) and does not allow DR2 to develop its full potential.

In Figure 5 the results of the different resampling techniques together with R-NSGA-II are evaluated on the ZDT4-(0.05,5) problem with reference point (0.05, 0.5) and 5000 function evaluations. Since the the ZDT4-(0.05,5) problem is more difficult it allows for a more clear evaluation. Here, it can be seen very clearly that DR2 outperforms Rank-Time-based Resampling, and thereby all other resampling algorithms evaluated in Figure 3. DDR however, shows not to be very powerful on this problem. Yet, combined with the Pareto-rank criterion as DR2, it is superior to all others.

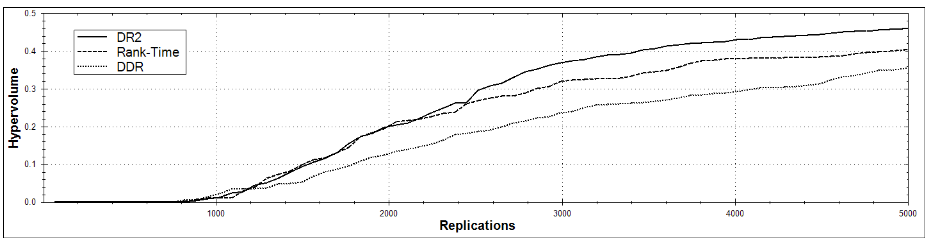

Fig. 5. Focused Hypervolume chart showing the R-NSGA-II progress development over time on ZDT4-(0.05,5) for resampling methods that use preference information. Reference point (0.05, 0.5). For comparison purposes, the curve for Rank-Time-based resampling from Figure 3 is included.

6 Conclusions and future work

We have proposed and evaluated Hybrid Dynamic Resampling strategies that use multiple resampling criteria on the guided EMO algorithm R-NSGA-II. Examples are Rank-Time-based Dynamic Resampling which uses the Pareto-rank and elapsed optimization runtime for sampling allocation, or Distance-Progress-Time-based Dynamic Resampling (DDR) [15]. They are compared with resampling techniques that base their sampling allocation on a single criterion, like Time-based Dynamic Resampling or Rank-based Dynamic Resampling. The results on benchmark functions and a reference point close to the Ideal point show that Hybrid Dynamic Resampling techniques are superior to single-criterion techniques and Static Resampling, given that the optimization problem is sufficiently complex. Furthermore, we proposed and evaluated a resampling algorithm that uses both the Pareto-rank and Reference point distance as a basis for sampling allocation. Both these criteria are used by the R-NSGA-II algorithm as fitness functions. Thus, we expected that the Distance-Rank Dynamic Resampling algorithm (DR2) is able to support the R-NSGA-II algorithm better than previous resampling algorithms that only consider one of the criteria, which we could prove in numerical benchmark experiments.

Future work will cover the following studies:

- A future task will be to study the combination of a resampling algorithm that uses the objective variance and Distance-based Dynamic Resampling. Such a resampling strategy based on variance is Multi-objective Standard Error Dynamic Resampling [15].
- The resampling algorithms in this paper that are based on the Pareto-rank base their sampling allocation on a comparison of solutions. They have thereby an advantage over resampling algorithms that treat each solution individually. Slightly modified, the comparison approach could support the evolutionary optimization algorithm in comparing solutions for selection decisions, also called Selection Sampling. A study investigating the effect of Selection Sampling on guided EMO of stochastic systems will be performed.
- A parametric study will be performed that identifies guidelines for parameter configuration for different problems characteristics.
- A worthwhile future task will be to extend and evaluate the resampling and optimization algorithms for scenarios with feasible reference points.
- Extensions for existing EMO algorithms for guided search need to be proposed that allow for faster convergence to the preferred objective space area.

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

References

1. Bartz-Beielstein, T., Blum, D., and Branke, J.: Particle swarm optimization and sequential sampling in noisy environments. Metaheuristics – Progress in Complex Systems Optimization, 261–273 (2007)
2. Branke, J., and Gamer, J.: Efficient sampling in interactive multi-criteria selection. Proceedings of the 2007 INFORMS Simulation Society Research Workshop, 42–46 (2007)
3. Branke, J., and Schmidt, C.: Sequential Sampling in Noisy Environments. Parallel Problem Solving from Nature - PPSN VIII - LNCS, vol. 3242, 202–211 (2004)
4. Chen, C. H., He, D., Fu, M., and Lee, L. H.: Efficient Simulation Budget Allocation for Selecting an Optimal Subset. Informs Journal on Computing, vol. 20, no. 4, pp. 579–595 (2008)
5. Deb, K.: Multi-Objective Opimization using Evolutionary Algorithms. John Wiley & Sons (2001)
6. Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, 182–197 (2002)

7. Deb, K., Sinha, A., Korhonen, P. J., and Wallenius, J.: An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. IEEE Transactions on Evolutionary Computation, vol. 14, no. 5, pp. 723-739, (2010)
8. Deb, K., Siegmund, F., and Ng, A. H.C.: R-HV : A Metric for Computing Hypervolume for Reference Point Based EMOs. Accepted for publication at the International Conference on Swarm, Evolutionary, and Memetic Computing 2014, Bhubaneswar, Odisha, India (2014)
9. Deb, K., Sundar, J., Bhaskara Rao N., U., and Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research, vol. 2, no. 3, 273–286 (2006)
10. Di Pietro, A.: Optimizing Evolutionary Strategies for Problems with Varying Noise Strength. PhD-thesis, University of Western Australia, Perth (2007)
11. Di Pietro, A., While, L., and Barone, L.: Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. Congress on Evolutionary Computation 2004, vol. 2, 1254–1261 (2004)
12. Jin, Y., and Branke, J.: Evolutionary optimization in uncertain environments - a survey. IEEE Transactions on Evolutionary Computation, vol. 9, no. 3, pp. 303–317, ISSN 1089-778X (2005)
13. Siegmund, F., Bernedixen, J., Pehrsson, L., Ng, A. H.C., and Deb, K.: Reference point-based Evolutionary Multi-objective Optimization for Industrial Systems Simulation. Proceedings of the Winter Simulation Conference 2012, Berlin, Germany, ISBN 978-1-4673-4781-5 (2012)
14. Siegmund, F., Ng, A. H.C., and Deb, K.: Finding a preferred diverse set of Pareto-optimal solutions for a limited number of function calls. Proceedings of the IEEE Congress on Evolutionary Computation 2012, Brisbane, Australia, pp. 2417–2424, ISBN 978-1-4673-1508-1 (2012)
15. Siegmund, F., Ng, A. H.C., and Deb, K.: A Comparative Study of Dynamic Resampling Strategies for Guided Evolutionary Multi-Objective Optimization. Proceedings of the IEEE Congress on Evolutionary Computation 2013, Cancún, Mexico, pp. 1826–1835, ISBN 978-1-4799-0454-9 (2013)
16. Siegmund, F., Ng, A. H.C., and Deb, K.: Dynamic Resampling Strategies for Guided EMO of Stochastic Systems - Part1. European Journal of Operational Research - EJOR, in preparation for submission (2015)
17. Stump, G. and Simpson, T. W. and Donndelinger, J. A. and Lego, S. and Yukish, M.: Visual Steering Commands for Trade Space Exploration: User-Guided Sampling With Example. Journal of Computing and Information Science in Engineering, vol. 9, no. 4, pp. 044501:1–10, ISSN 1530-9827 (2009)
18. Zitzler, E., Deb, K., and Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, vol. 8, no. 2, 173–195 (2000)
19. Zitzler, E., and Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. Parallel Problem Solving from Nature V, 292–301 (1998)