Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

REFERENCE POINT-BASED EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION FOR INDUSTRIAL SYSTEMS SIMULATION

Florian Siegmund
Jacob Bernedixen
Leif Pehrsson
Amos H.C. Ng

Virtual Systems Research Center
University of Skövde
Högskolevägen, 541 28 Skövde, SWEDEN

Kalyanmoy Deb

Department of Mechanical Engineering
Indian Institute of Technology Kanpur
PIN 208016, INDIA

ABSTRACT

In Multi-objective Optimization the goal is to present a set of Pareto-optimal solutions to the decision maker (DM). One out of these solutions is then chosen according to the DM preferences. Given that the DM has some general idea of what type of solution is preferred, a more efficient optimization could be run. This can be accomplished by letting the optimization algorithm make use of this preference information and guide the search towards better solutions that correspond to the preferences. One example for such kind of algorithms is the Reference point-based NSGA-II algorithm (R-NSGA-II), by which user-specified reference points can be used to guide the search in the objective space and the diversity of the focused Pareto-set can be controlled. In this paper, the applicability of the R-NSGA-II algorithm in solving industrial-scale simulation-based optimization problems is illustrated through a case study for the improvement of a production line.

1 INTRODUCTION

Simulation-based multi-objective optimization problems are in many cases difficult to solve which is caused by the vast number of simulation runs that are needed in order to find a converged and diverse set of Pareto-optimal solutions. The standard goal usually pursued in multi-objective optimization is to find a converged set of solutions that is also featuring diversity. This goal requires to find a large number of solutions to cover the whole Pareto-front. Besides a complex problem structure the high number of simulation runs can also be caused by uncertainty in the input or output parameters of the model which has to be handled by simulating every solution multiple times and using the average objective values. Also a high number of problem objectives can make it necessary to execute many solution evaluations. This is because the Pareto-optimal front of a high-dimensional problem is a large subspace. In order to cover it sufficiently many Pareto-optimal solutions need to be found.

For complex real-world simulation models the execution time usually is high so that the overall number of simulation runs that can be executed is limited. With this limited simulation time budget in many cases only an inferior Pareto-optimal front can be found. For many optimization problems however the DM has domain knowledge about which solutions are most interesting. After the optimization has been run only a single solution is chosen and implemented. If the DM can provide preference information that indicates where the DM expects to choose a solution the algorithm can exploit this information and guide the search towards this area. In this way only a small part of the Pareto-optimal front needs to be covered

and the algorithm can find a focused solution set that is more converged in the interesting area than algorithms that aim at finding a solution set covering the whole Pareto-front.

Several algorithms have been proposed that can focus the search towards a certain area of the objective space. A comprehensive overview on guided search algorithms controlled by preferences is given in (Miettinen, 1999; Branke et al. 2008). Guided search algorithms can be distinguished by the type of information that is used to specify interesting solutions. For example the DM can make comparisons between different solutions. This information can be transformed into a value function that serves as a fitness function for the optimization algorithm (Deb et al. 2010). Another way of preference articulation is the specification of a preference direction or light beam along which the search is guided (Deb and Kumar, 2007; Jaszkiewicz and Slowinski, 1999). The way of preference specification that is used in this article requires the user to specify one or more reference points in interesting areas of the objective space where the search is guided to. An algorithm that implements this search is the R-NSGA-II algorithm which was proposed by (Deb et al. 2006). Extensions have been proposed by (Siegmund, Ng, and Deb, 2012).

In this paper we present the application of this algorithm in a real-world simulation optimization case study. This simulation optimization problem deals with the optimization of a production line that can be configured via binary parameters.

The paper is structured as follows. The R-NSGA-II algorithm is described in section 2. In section 3 the industrial case is introduced. Following, the experiments and results are presented in section 4. In section 5 we draw conclusions and present future research questions.

2 R-NSGA-II ALGORITHM

In this study the application of the R-NSGA-II algorithm on a real-world simulation optimization problem is performed and evaluated. In this section we describe the algorithm on a general level without going into details. This is to give the reader a general overview and understanding of the way the algorithm works.

The R-NSGA-II algorithm is a multi-objective evolutionary optimization algorithm that is based on the Non-dominated Sorting Genetic Algorithm II, NSGA-II (Deb et al. 2002). The NSGA-II algorithm is an elitist multi-objective evolutionary algorithm that uses two types of fitness functions. The primary fitness function is the Pareto-optimality. Solutions that cannot be compared by means of Pareto-optimality are distinguished by a secondary fitness criterion called crowding distance. To guide the search for new solutions the R-NSGA-II algorithm allows the specification of one or more reference points in the objective space where the search is attracted to. Solutions close to the reference points are preferred. This fitness criterion replaces the crowding distance measure of NSGA-II. In the following the context of where NSGA-II uses the crowding distance measure is described and how the distance measure to the reference points is used instead.

The primary fitness criterion Pareto-optimality means that solutions are preferred that are not worse in any objective than another solution but truly better in at least one objective. If this is true for two solutions then the better solution is said to be dominating the other solution. Solutions that are not dominated by any other solution are called non-dominated solutions. All non-dominated solutions form a set which is called a front. Assumed that those solutions are removed a new front of non-dominated solutions will be formed. In this way a solution set can be fully partitioned into front subsets. Inside of those subsets solutions are not comparable. NSGA-II selects solutions according to their front membership. It forms the union of population and offspring and partitions this set into fronts. All fronts that fit completely into the next population are selected. The front that only partially fits into the next population requires a special selection procedure. Here the secondary fitness type is applied. NSGA-II uses a diversity fitness measure called crowding distance which makes sure that a subset of this partially selected front with high diversity is selected. R-NSGA-II uses a different secondary fitness measure. It is based on the distance of the solutions to the reference points. Those who are closest to the reference points are selected into the next population. By means of this secondary selection criterion the algorithm can be guided towards the reference points. However, at the same time a certain degree of diversity must be maintained. For this purpose R-

NSGA-II uses a clustering algorithm that filters out solutions that are too close to each other. The remaining solutions are considered for selection. Without this clustering step the algorithm would find a population which is situated in very close proximity to the reference points in the objective space. The alternative solutions the DM could choose from would not be very different from each other.

2.1 ALGORITHM DESCRIPTION

The R-NSGA-II algorithm follows a scheme as of many evolutionary algorithms. A set of solutions, the population, is maintained whereof new offspring solutions are generated through mating, crossover and mutation (Deb, 2001). The best solutions among the old population and the offspring solutions are chosen to form the new population. The working scheme is depicted in Figure 1. In contrast to the NSGA-II algorithm which uses the Pareto-rank as fitness function in the tournament selection for mating the R-NSGA-II algorithm uses the reference point rank. The reference point rank is calculated by creating a distance ranking among all solutions for each reference point. For a certain reference point the closest solution is assigned rank 1, the second closest solution rank 2, and so on. The reference point rank for a solution is the minimum rank that has been assigned to it among all different rankings.

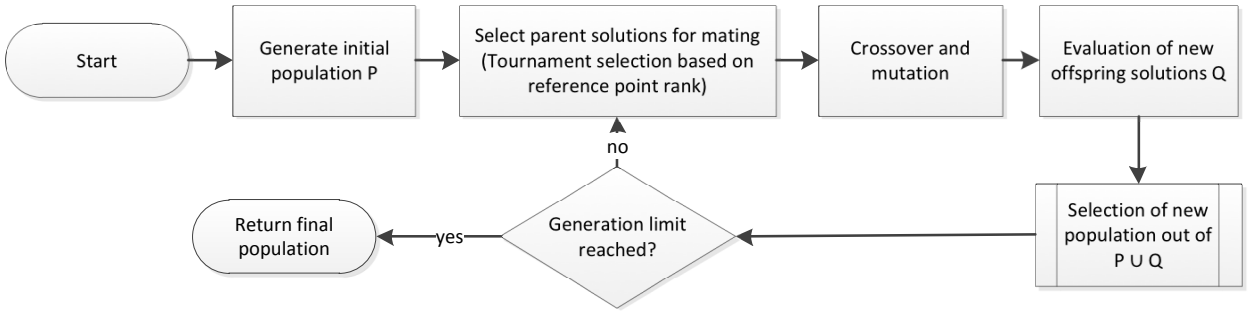

Figure 1: R-NSGA-II flowchart

The selection step of the R-NSGA-II algorithm that is marked as a sub process in Figure 1 is complex and is therefore displayed in Figure 2. After partitioning the set of population and offspring solutions into fronts on each front a clustering algorithm is performed. It chooses a representative solution out of the front which has the lowest reference point rank. All solutions within a radius epsilon are cleared and are added to the cluster of the representative. In the following they are not considered for selection, only the representative solution is considered. The clustering selects the next representative solution and clears out the neighboring solutions again. This is continued until all solutions have been assigned to clusters. This is performed for all fronts. The epsilon distance can be chosen as a parameter to allow the user to choose different amounts of diversity.

After the clustering the algorithm works similar as NSGA-II. It browses through the fronts and checks whether the current front, respectively its set of cluster representatives can be selected completely into the next front. In this case all representative solutions are selected into the next population and the algorithm continues with the next front. Otherwise, in case the algorithm can only use a subset of the cluster representatives, the representatives solutions with the lowest reference point ranks among the representatives are selected into the next population. The new population is complete and is returned to the algorithm main loop.

In case the algorithm reaches the last front and it was not possible to select enough solutions to fill the new population completely the algorithm starts over with the remaining solutions. The clustering is performed once more and the algorithm browses through the fronts starting with the first front.

For the full functionality of the algorithm the reader may confer to (Deb et al. 2006; Siegmund, Ng, and Deb, 2012).

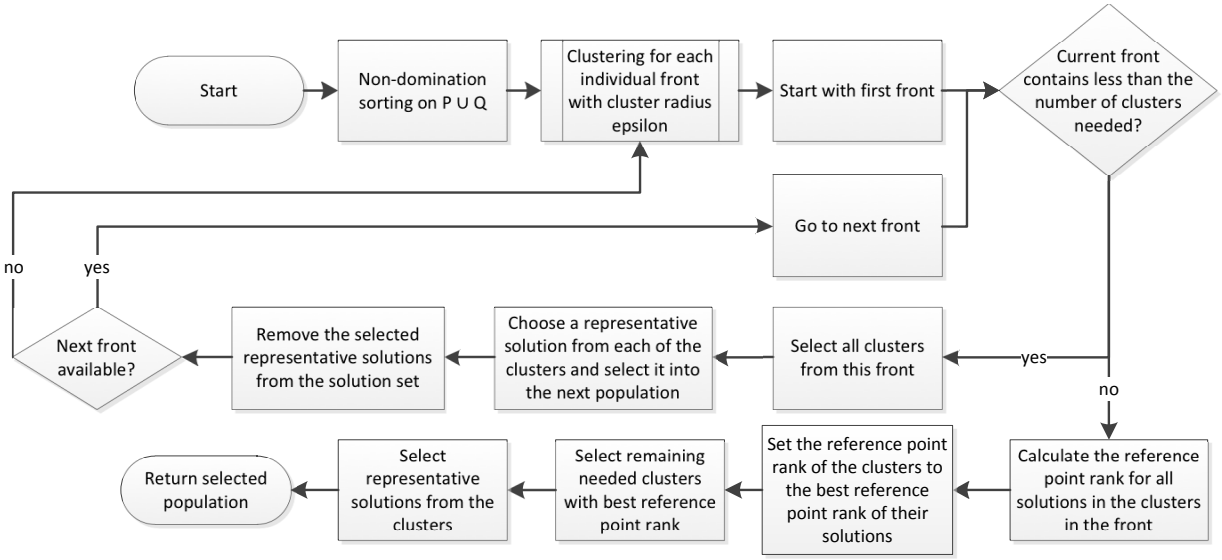

Figure 2: R-NSGA-II selection step sub flow

3 GUIDED SEARCH OPTIMIZATION CASE STUDY

This section describes the production line used in the case study. It is a large production line with around 90 stations and just over 70 buffers, mostly conveyor belts but also some larger warehouses. The line is very complex with several parallel sections, some assembly, several two-piece stations, automatic as well as manual testing with scrap or rework options, portal cranes, and some pallet loops. Figure 3 shows a simplified layout of the described production line.

The engineers in charge of the production line are faced with a rather tough, but not uncommon task. They are to make it capable of handling some new variants coming in and at the same time increase its capacity in order to meet customer demands. Based on the current condition of the line and a simulation study considering the new variants they are far from reaching their capacity goal. They are able to reach a throughput of 84 pieces per hour instead of the needed 95 pieces per hour, i.e. they will have to increase the capacity with at least 13%.

Given the size and complexity of the line it is not an easy task to decide where to make improvements in order to reach this goal, and it is even harder if at the same time it has to be made sure that as few improvements as possible are needed. Three types of improvements were considered as shown in Table 1 (due to limited amount of space, larger buffer capacities were excluded as a possible improvement).

Table 1: Improvement, values, and example of possible ways of achieving the improvement.

Improvement	**Values**	**Possible solutions**
Reduction of cycle time	Original value down to 26 seconds.	New faster machine, new tools, improvement machine sequence or code.
Improved availability	Original value up to 98%.	New more reliable machine or tools.
Reduction of mean time to repair	Original value down to 5 minutes.	More resources for support and maintenance.

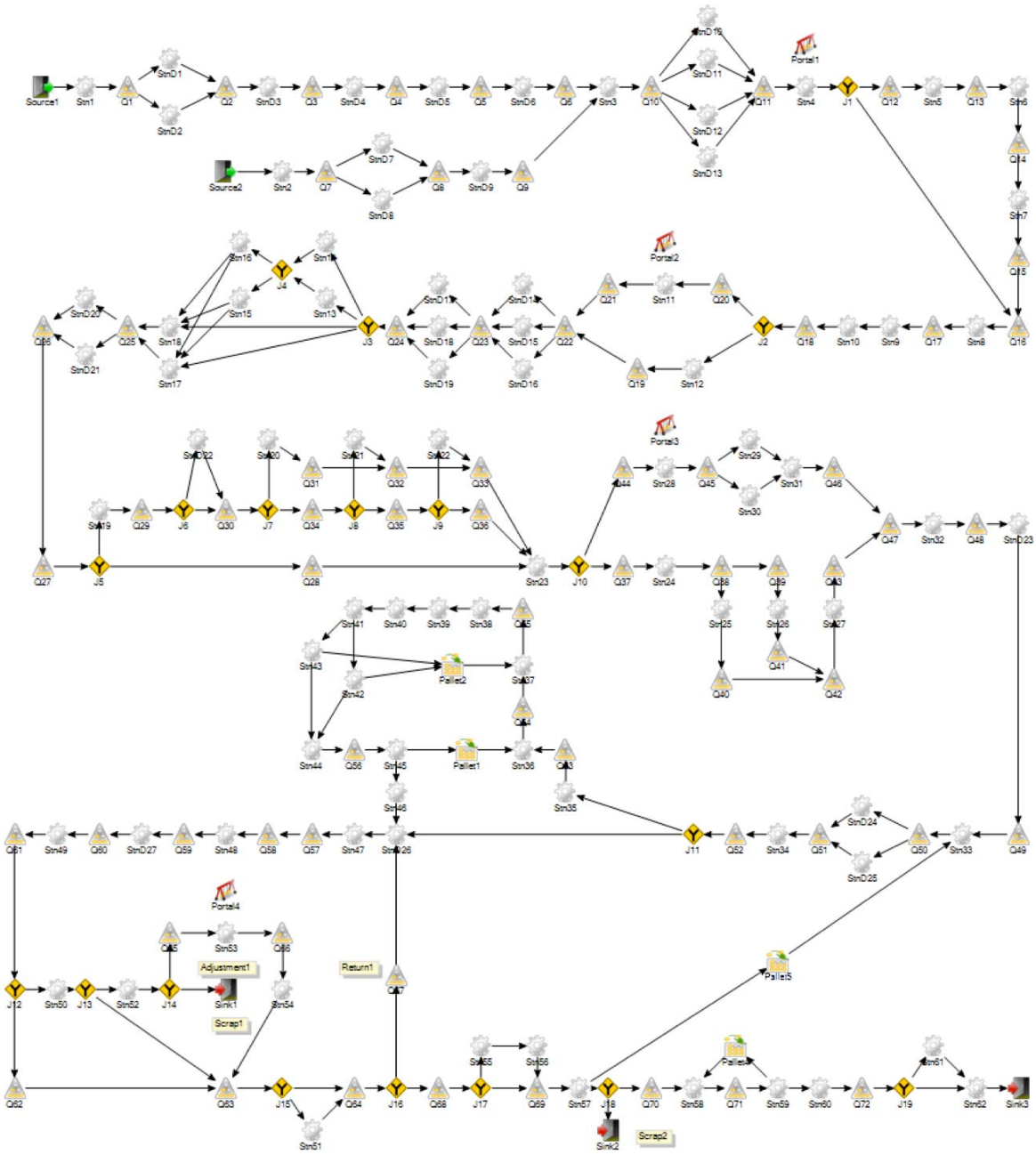

Figure 3: Simplified layout of the machining line.

The values for each improvement were provided by the engineers in charge of the production line. These improvements were considered on all stations if their original value was not worse than the proposed improvement value. This resulted in 128 potential improvements, which in turn represents $2^{128} \approx 3.4\text{E+}38$ possible combinations of improvements. Given this huge number of possible combinations it was decided to use simulation based multi-objective optimization to evaluate what improvements that should be implemented.

Using binary variables for representing the improvements (value = 0, original value / value=1, improvement value) the following multi-objective optimization problem was defined in which an implemented improvement is counted as one change.

The formal description of the optimization problem looks as follows:

$$\min Changes = \min\left(\sum_{i=1}^{58}\alpha_i + \sum_{j=1}^{24}\beta_j + \sum_{k=1}^{46}\gamma_k\right)$$
$$\max Throughput$$

where

$$\alpha_i = cycletime\ improvement \in \{0,1\}$$
$$\beta_j = availability\ improvent \in \{0,1\}$$
$$\gamma_k = mean\ time\ to\ repair\ improvement \in \{0,1\}$$

4 NUMERICAL EXPERIMENTS

In this section the experiment setup of the case study is described and the results are analyzed.

4.1 Case Study

The simulation horizon of the model was set to one month (32 days with 2 days of warm-up time) and evaluation was run with five replications. The time to run one evaluation was about 16 minutes. To speed up the search an initial population was used. The initial solution with zero changes (representing the current conditions of the line) was added to the otherwise randomly chosen initial population of the optimization. The optimizations in this study were allowed to run for 5000 evaluations. Given the stochastic nature of evolutionary optimization algorithms the optimizations were replicated five times in order to get reliable results. Results from these replicated optimizations are illustrated using median attainment surfaces (Bartz-Beielstein et al., 2010).

The NSGA-II algorithm was configured as follows. Since the problem is a binary decision problem the algorithm works on binary vectors. Uniform crossover is used which chooses randomly between the parameter values of two solutions, with uniform probability. Crossover probability is 0.8.

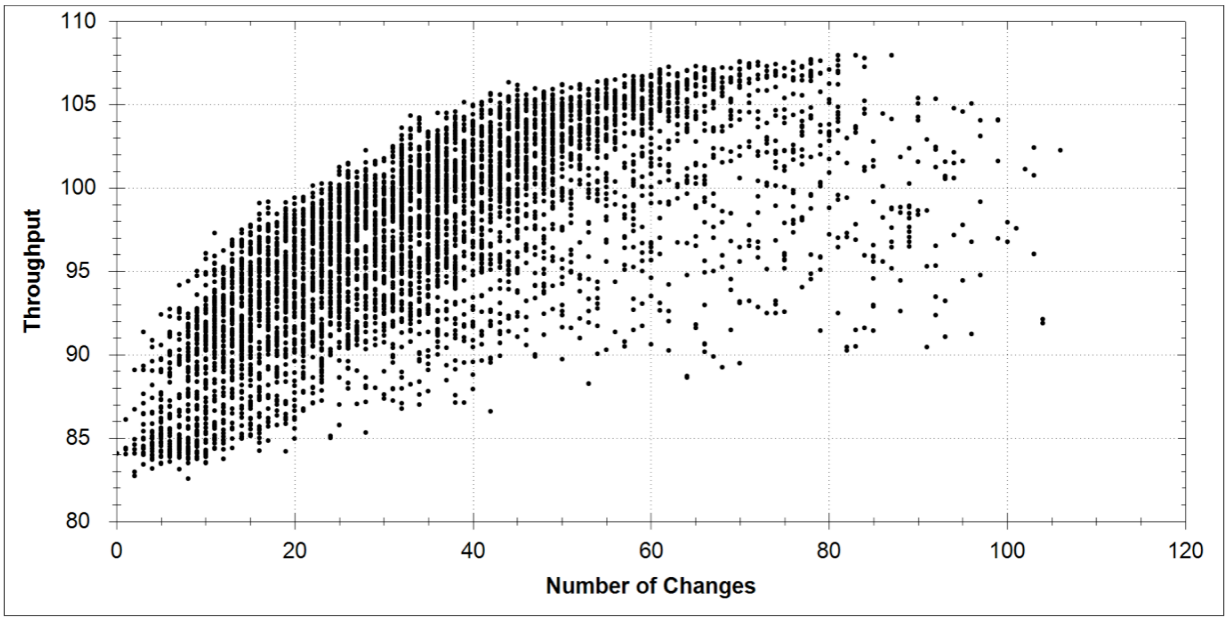

Figure 4: Results from a run without reference point (NSGA-II).

Mutation is performed by choosing uniformly either value 0 or 1 for the mutated parameters. Mutation probability is 1/15 which means that on average 9 of 128 parameters are mutated. The population size is chosen to be 50.

As a basis the optimization is run once by NSGA-II without guidance. The result is displayed in Figure 4. It is able to find a converged and diverse set of solutions that even includes several solutions that reach and exceed the capacity goal of 95 while at the same time requiring less than 10 changes. This was, according to the engineers, a reasonable number of changes. The following goal was set by the automation engineers: "Try to reach a throughput of 95 pieces per hour with 10 or fewer improvements on the line". This goal was added to the optimization algorithm as the reference point in order to find out if this additional information could help the optimization in providing even better solutions given the same budget.

R-NSGA-II was configured identically to NSGA-II. Additionally, the reference point was chosen to be (Number of Changes = 10, Throughput = 95) for all experiments. As distance measure an achievement scalarizing function was used that calculates the maximum normalized objective distance between a solution and the reference point, as defined in (1). For diversity control epsilon is configured to be 0.001 unless otherwise stated.

$$ASF(\overline{F(s)}, \bar{r}) = \max_i \left\{ abs(\overline{F_i(s)} - \bar{r}_i) \right\} \quad (1)$$

where

$\overline{F(s)}$ is the normalized vector of fitness values of solution s

$\bar{r}$ is the normalized reference point in the objective space

The result of R-NSGA-II with reference point (10, 95) is displayed in Figure 5. The optimization was able to find solutions that satisfy the capacity goal of 95 pieces per hour. The solution that met the capacity goal and had the fewest number of changes produced 96.7 pieces per hour with 8 changes. Compared to NSGA-II the solutions found by the optimization could achieve an improvement in both objectives.

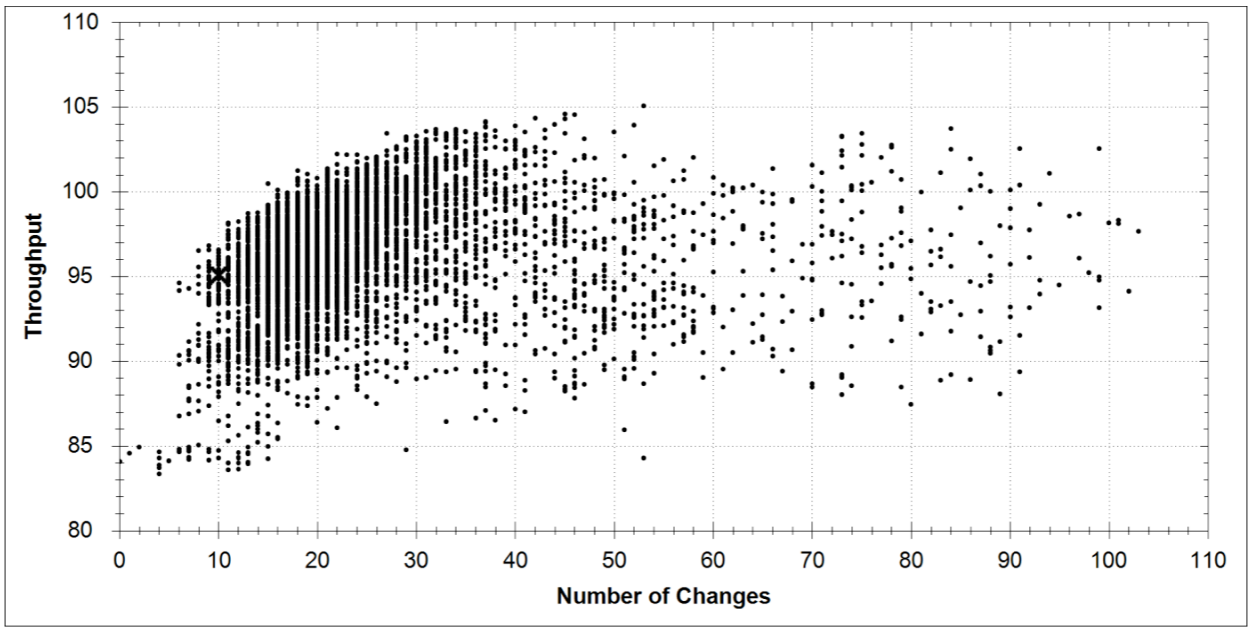

Figure 5: Results from a run with reference point (R-NSGA-II). Reference point (10,95).

To obtain reliable results both optimizations have been executed five times and their average performance was compared via a median attainment surface as in Figure 6. The performance advantage of R-

NSGA-II can clearly be observed. Also a tendency of R-NSGA-II to achieve the most improvement at parts of the front with higher throughput values than the reference point can be seen. It is assumed that this is an effect described in (Siegmund, Ng, and Deb, 2012). The 5000 evaluations do not allow the algorithm to converge fully to the true Pareto-front. In this state algorithms based on Pareto-optimality have a tendency to have a bias towards the ideal point (Deb, 2001) which lies approximately at (0, 115).

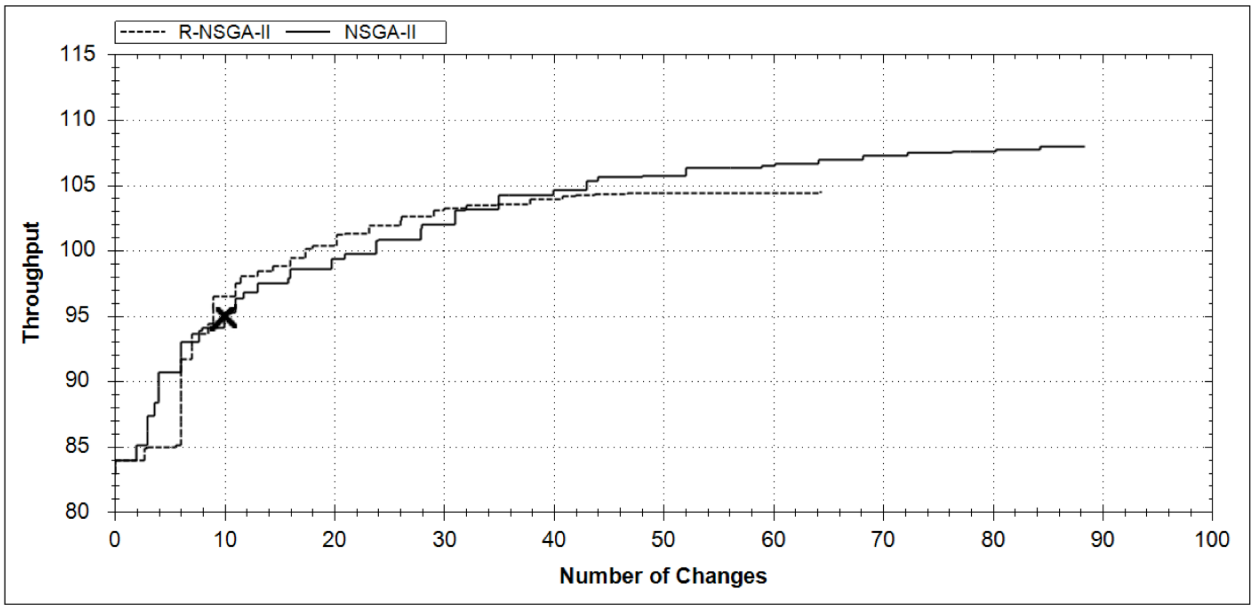

Figure 6: Attainment surface combining the experiments of R-NSGA-II (dotted line) and NSGA-II (solid line). Reference point (10, 95).

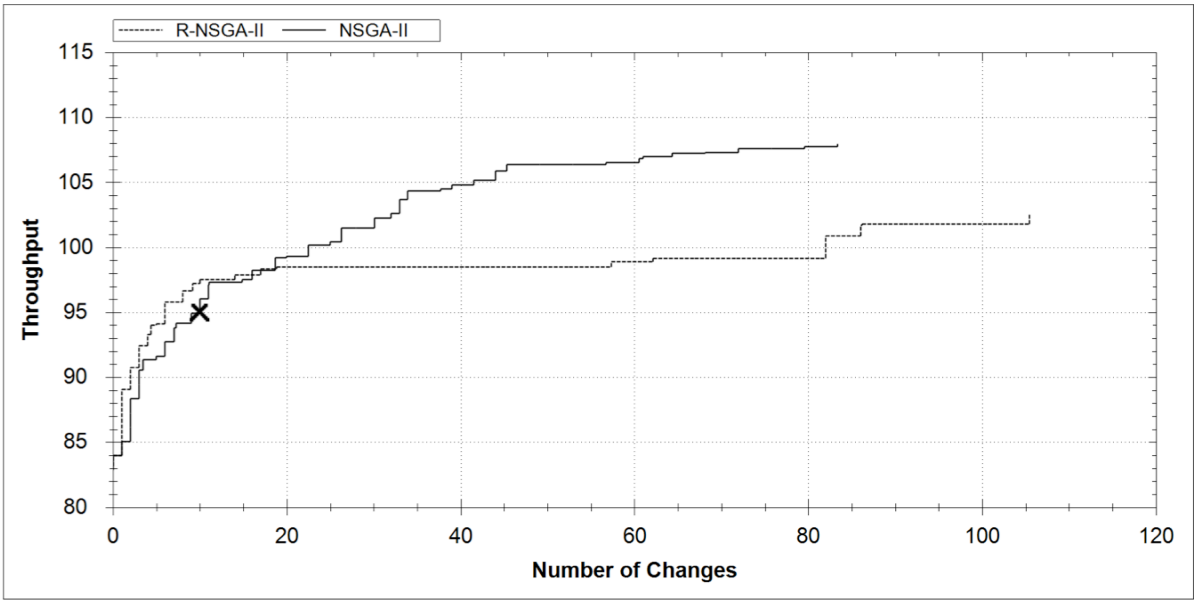

Figure 7: Attainment surface combining the experiments of R-NSGA-II (dotted line) and NSGA-II (solid line), using biased crossover and mutation operators favoring a lower number of changes. Reference point (10, 95).

4.2 Biased Variation Operators

Considering the long run time and the tremendously large search space (3.4E+38 possible combinations) biased crossover and mutation operators promoting low number of changes were implemented in the optimization algorithm in order to get even better results. In detail, continuous variation operators are used that generate crossover and mutation values that lie in the interval [0,1]. All values in [0, 1) are rounded downwards to 0.

In Figure 7 the attainment surface of the experiments with the biased variation operators is displayed. The dotted line represents the result of R-NSGA-II. The performance advantage in the preferred region of R-NSGA-II over NSGA-II is even more clear compared to the previous experiment (Figure 6).

4.3 Diversity Control

The R-NSGA-II algorithm allows to control the diversity of the found solutions via a parameter called epsilon. Since R-NSGA-II prioritizes solutions with high reference point rank before Pareto-optimal solutions R-NSGA-II uses clustering to maintain diversity. The parameter epsilon is used in the clustering algorithm as the minimum distance between two selected solutions as in (1). Step by step a representative solution is chosen and all other solutions within radius epsilon around this solution are discarded. In this chapter we show the effect of choosing a different epsilon parameter value. Epsilon is increased from the original value 0.001 to 0.005. The objective vectors of the Pareto-front in the original experiment with epsilon 0.001 are shown in Figure 8. The results of the increased diversity experiment with epsilon 0.005 are displayed in Figure 9. The results show that by increasing epsilon the Pareto-front is not more widespread. However, a more evenly spread Pareto-front can be found. During the optimization process more solutions in dense areas of the current non-dominated front of the population are cleared out than in the original experiment. This increases the probability for the creation of Pareto-solutions in more sparse areas of the Pareto-front. If the budget for simulation evaluations would not be limited the final population would be more widespread for epsilon 0.005 than for epsilon 0.001 as shown in the experiments in (Deb et al. 2006).

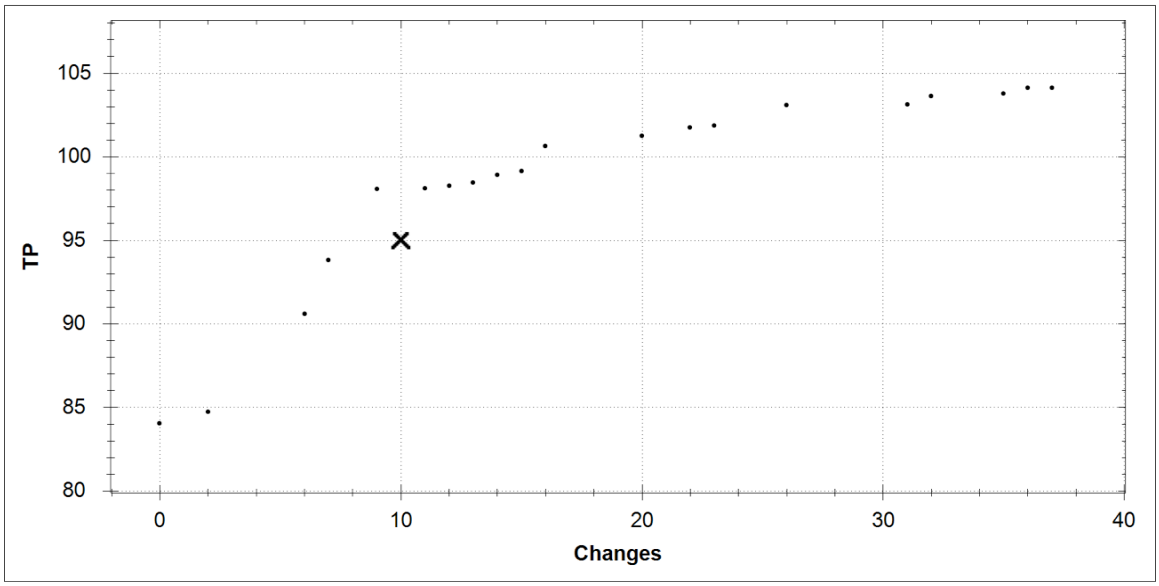

Figure 8: Found Pareto-front for experiment with diversity parameter value epsilon=0.001. Reference point (10, 95).

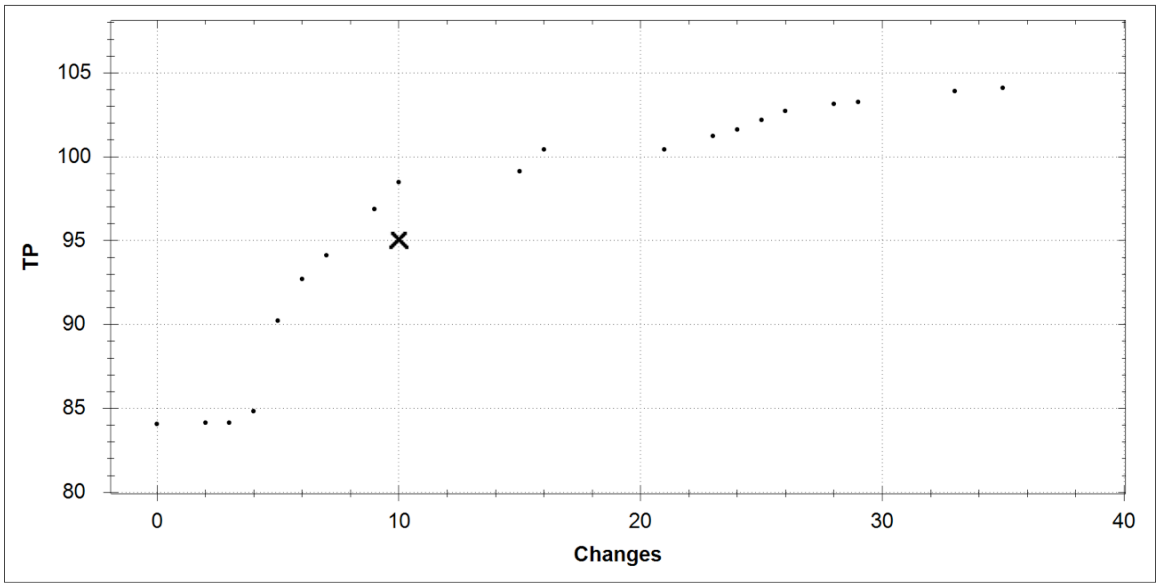

Figure 9: Found Pareto-front for experiment with diversity parameter value epsilon=0.005. Reference point (10,95).

5 CONCLUSIONS

The following conditions are often present in industrial cases, (1) limited time for analysis (e.g. optimizations), (2) very large and complex problems, and (3) knowledge of what represents a desirable and somewhat realistic performance of the studied system exists. This paper has shown, at least for one such case, a successful use of a reference point guided optimization algorithm, R-NSGA-II, that outperformed its standard counterpart, NSGA-II.

In future studies we will test the use of multiple reference points. Another ability of R-NSGA-II will be demonstrated in future work. In multi-objective simulation optimization problems it can happen that parts of the Pareto-front are not fully explored. This occurs often for parts of the Pareto-front where a small improvement of one objectives leads to large losses regarding another objective. Here it is difficult to find a diverse set of Pareto-optimal solutions. With R-NSGA-II those areas can be explored and the gaps can be filled. One such case is described in (Aslam, Karlsson, and Ng, 2012).

Several extensions to R-NSGA-II have been made regarding diversity control and compromised Pareto-optimality (Siegmund, Ng, and Deb, 2012). There will be follow up studies testing these extensions of R-NSGA-II on real-world problems.

ACKNOWLEDGMENTS

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

REFERENCES

Aslam T., I. Karlsson, A.H.C. Ng. 2012 ”Integrating system dynamics and multi-objective optimization for manufacturing supply chain analysis” Submitted for publication in the *Proceedings of the Winter Simulation Conference 2012*.

Branke J., K. Deb, K. Miettinen, and R. Slowinski. 2008 “Multiobjective optimization - Interactive and evolutionary approaches – LNCS,” Springer-Verlag, Berlin.

Bartz-Beielstein T., M. Chiarandini, L. Paquete, M. Preuss. 2012 “Experimental Methods for the Analysis of Optimization Algorithms,” Springer-Verlag, Berlin.

Deb K. 2001 "Multi-Objective Opimization using Evolutionary Algorithms," John Wiley & Sons.

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

Deb K., A. Kumar, 2007 "Light beam search based multi-objective optimization using evolutionary algorithms," *IEEE Congress on Evolutionary Computation 2007*, pp. 2125–2132.

Deb K., A. Sinha, P.J. Korhonen, and J. Wallenius. 2010 "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.

Deb K., J. Sundar, U. Bhaskara Rao N., and S. Chaudhuri. 2006 "Reference point based multi-objective optimization using evolutionary algorithms," *Int. Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286.

Jaszkiewicz A. and R. Slowinski. 1999 "The light beam search approach – An overview of methodology and applications," *European Journal of Operation Research*, vol. 133, no.2, pp. 300–314.

Miettinen K., 1999 "Nonlinear Multiobjective Optimization," *Kluwer's International Series in Operations Research & Management Science*, vol. 12, Kluwer Academic Publishers.

Siegmund F., A. H.C. Ng, K. Deb. 2012 "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,* in press.

AUTHOR BIOGRAPHIES

FLORIAN SIEGMUND studied Computer Science and Operations Research at Karlsruhe Institute of Technology, Germany. In 2009 and 2010 he was a visiting research student at University of Skövde, Sweden and National University of Singapore. Since 2011 he has been working as a Ph.D. student at University of Skövde, Sweden. His research interests are in heuristic optimization algorithms for simulation-based optimization. His email address is florian.siegmund@his.se.

JACOB BERNEDIXEN is a system developer and PhD student at University of Skövde. He received his MSc degree from University of Linköping, Sweden. His email address is jacob.bernedixen@his.se.

LEIF PEHRSSON is an industrial PhD student at University of Skövde. His email address is leif.pehrsson@his.se.

AMOS H.C. NG is an Associate Professor at the University of Skövde, Sweden. He holds a B.Eng. degree and a M.Phil. degree, both in Manufacturing Engineering from the City University of Hong Kong and a Ph.D. degree in Computing Sciences and Engineering from De Montfort University, Leicester, U.K. He is a member of the IEE and a Chartered Engineer in the U.K. His research interests include virtual engineering for manufacturing machinery and machine systems as well as simulation-based optimization. His email address is amos.ng@his.se.

KALYANMOY DEB is currently the Deva Raj Chair Professor of Mechanical Engineering with the Indian Institute of Technology Kanpur, India. He obtained his PhD from the University of Alabama, USA. His main research interests are in the area of computational optimization, modeling and design, and evolutionary algorithms. He is a Fellow of the Indian National Academy of Engineering, the Indian Academy of Sciences, and of the Institute of Electrical and Electronics Engineers IEEE. His email address is deb@iitk.ac.in.