Postprint

Parameter Tuning of MOEAs using a Bilevel Optimization Approach

Martin Andersson, Sunith Bandaru, Amos Ng, and Anna Syberfeldt

Virtual Systems Research Centre, University of Skövde, Skövde, Sweden,
`{martin.andersson,sunith.bandaru,amos.ng,anna.syberfeldt}@his.se`

Abstract. The performance of an Evolutionary Algorithm (EA) can be greatly influenced by its parameters. The optimal parameter settings are also not necessarily the same across different problems. Finding the optimal set of parameters is therefore a difficult and often time-consuming task. This paper presents results of parameter tuning experiments on the NSGA-II and NSGA-III algorithms using the ZDT test problems. The aim is to gain new insights on the characteristics of the optimal parameter settings and to study if the parameters impose the same effect on both NSGA-II and NSGA-III. The experiments also aim at testing if the rule of thumb that the mutation probability should be set to one divided by the number of decision variables is a good heuristic on the ZDT problems. A comparison of the performance of NSGA-II and NSGA-III on the ZDT problems is also made.

Keywords: parameter tuning, NSGA-II, NSGA-III, ZDT, bilevel optimization, multi-objective problems

1 Introduction

Real-world optimization problems are often formulated with multiple objectives and are therefore preferably solved using multi-objective evolutionary algorithms (MOEAs). Metaheuristics such as EAs involve a set of user-defined parameters that control various aspects of the algorithm. It is well-known [1, 10] that these settings can greatly affect the search process and the overall performance of the algorithm. However, setting them for a particular problem is not always intuitive. A strategy that is often used is to choose parameter values that have been shown to be effective on similar problems. Some metaheuristics, such as evolutionary strategies (ES), come with their own heuristics or recommendations for choosing the parameters. Neither method guarantees maximal performance from the algorithm. This paper addresses this issue by using the idea of *optimal* parameters, similar in principle to the one proposed in [9]. The parameter-setting problem can itself be viewed as an optimization problem in which the objective is to maximize the performance of the algorithm used on a particular problem. For single-objective problems, this performance indicator could be directly related to the best function value attained by the algorithm. Since this work considers

multi-objective problems, a commonly used performance indicator is the hyper-volume [12]. Thus, our formulation contains a multi-objective problem nested within a single-objective problem and resembles the following,

$$
\begin{aligned}
&\underset{\mathbf{p}}{\text{Maximize}} \quad HV(\mathbf{p}) \\
&\text{where,} \qquad HV(\mathbf{p}) \text{ is the hypervolume of the non-dominated set} \\
&\qquad\qquad \text{obtained by solving the following problem with parameters } \mathbf{p} \\
&\qquad\qquad \underset{\mathbf{x}}{\text{Minimize}} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\} \\
&\qquad\qquad \text{Subject to} \quad g_j(\mathbf{x}) \geq 0 \ \forall \ j \in \{1, 2, \ldots, J\} \\
&\qquad\qquad\qquad\qquad h_k(\mathbf{x}) = 0 \ \forall \ k \in \{1, 2, \ldots, K\} \\
&\qquad\qquad\qquad\qquad \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u
\end{aligned} \tag{1}
$$

The algorithmic parameters of the lower-level optimization problem become the variables for the upper-level optimization problem.

Many real-world optimization problems are also designed to be scalable with respect to the variables. For example, consider a production line involving several machines in which their processing times have to be optimized for maximizing the overall throughput and minimizing the work in process of the line. Adding additional operations (machines) to such a line is equivalent to scaling the original optimization problem since the objectives remain the same. In such a situation, it is beneficial to study how the optimal parameter values for the algorithm change with respect to the problem size. Another important aspect is the computational cost. Objective functions in the real world are rarely analytical. In other words, evaluation of the objective functions may involve computationally expensive simulations. Studying the impact of the available computational budget on the optimal parameter values can lead to considerable savings in time and cost.

In order to illustrate the above ideas, two MOEAs, namely NSGA-II [4] and NSGA-III [3], are chosen with the ZDT test suite [13] to experimentally study the effects of problem size and available budget. NSGA-II and the ZDT test problems is a combination that is commonly used to assess the performance of a new metaheuristic. Finding the optimal parameters and the corresponding hypervolume for this combination will also allow a new metaheuristic to be compared against the NSGA-II best-case performance on the ZDT problems. Other test problems are not included in this study because that would reduce the number of experiments performed on each problem. This trade-off will allow for a more in-depth analysis of the ZDT problems.

It is worthwhile to mention here that the goal of this paper is not to find parameter settings that work across a range of problems, but to study how the optimal parameters vary for a given problem with the number of variables and budget size. In order to achieve this, several experiments will be performed on each problem to get multiple sets of optimized parameters. A secondary aim of this paper is to see how NSGA-III compares to NSGA-II in terms of performance and whether they use similar optimal parameter settings. Though NSGA-III was originally designed to handle many (> 3) objective problems, this paper will

address how it performs against NSGA-II on the ZDT problems. Testing against problems with three or more objectives would be interesting but is out of scope of this paper and left for future work.

The rest of the paper is organized as follows. Section 2 introduces the parameter setting problem and related work. In Section 3 a description of the experimental design is provided. The experimental results appear in Section 4. The conclusions are summarized in Section 5.

2 Background

The problem of finding the optimal set of parameters for a particular problem can itself be formulated as an optimization problem and solved by an EA. This bilevel optimization approach is called a meta-EA [5]. Though computationally intensive, the approach is highly parallelizeable since replications of the optimizations, at both the upper and lower-level, are independent. A software framework was developed as part of this work that could efficiently distribute and run optimizations in parallel. This software together with a cluster of homogeneous commodity computers enabled the scope of the experiments to be extended well beyond what would have been feasible on a single computer.

One issue that has to be considered when testing different parameters is that they are usually not independent. This means that changing the parameters one by one may lead to to sub-optimal settings. Changing them simultaneously on the other hand will require a large number of experiments to be performed. It is therefore impractical to perform parameter tuning manually, even though there exist different techniques to overcome this problem to some extent. A detailed description and taxonomy of the available techniques can be found in [5].

2.1 Classification and Terminology

It is possible to distinguish three layers in parameter tuning: The application layer, the algorithm (lower) layer and the design (upper) layer [5]. The problem to be solved is located on the application layer and the metaheuristic to solve that problem is on the algorithm layer. On the design layer is the parameter tuner that tests different parameters for the metaheuristic on the algorithm layer. To avoid confusion, the quality of solutions for the problem on the application layer is called fitness while the quality of the parameters is called utility [5]. The classification that was proposed in [6] distinguishes between parameter tuning where the parameters are static and parameter control where they can change during the optimization.

Tuners can be divided into two main categories: iterative and non-iterative [5]. Non-iterative tuners generate all parameters at the start, usually in a systematic fashion. This allows the utility landscape to be modeled from the utility of the evaluated parameters. Iterative tuners, on the other hand, generate the parameters iteratively as the tuner progresses. This makes them more suitable for finding the (near-)optimal parameter vectors, because they can perform a search of the utility landscape.

2.2 Related Work

Using a bilevel optimization approach to do parameter tuning has been done before in the literature. In [1] a Genetic Algorithm (GA) was tuned on single objective sphere problem. The authors found that the GA using the optimized parameters to be significantly better than a GA with "standard" parameters.

Another example is [7] which used a GA to tune the parameters for a GA on a set of numerical test functions. The result were then validated on a image registration task, showing a small but statistical significant advantage to the tuned GA against a "standard" GA.

A more recent example is [9] in which the authors used NSGA-II to tune the parameters of Partical Swarm Optimization (PSO) and Differential Evolution (DE). The algorithms were tuned against both the precision and speed of convergence. It was found that in addition to finding good parameters, the approach could also extract relationships between parameters and the impact of a parameter on the quality criteria.

3 Experimental Design

The meta-EA approach only provides a single optimal parameter set $\mathbf{p}^*$ for each experiment, meaning that it does not provide much insight into the utility landscape. This paper will address this issue by running several different experiments on the same test problem. Two things will be varied for all test problems: the function evaluation budget and the number of decision variables (N).

3.1 Experimental Setup

The experiments involve four aspects that this paper studies, these are listed in Table 1. Each experimental setting is combination of these different aspects. Thus, in total there are 350 (2 MOEAs $\times$ 5 test problems $\times$ 7 budget sizes $\times$ 5 problem sizes) different experimental settings each of which is independently replicated 20 times. The outcome of each replication is the set of parameters with the best hypervolume. Therefore, each experimental setting produced 20 different sets of parameters.

Each experimental setting was a bilevel optimization with a function evaluation budget of 1000, using the parameters shown in the third column of Table 2. The budget was based on manually analyzing a small number of bilevel optimizations and identifying the fact that most of them stopped improving after about 500 evaluations. The MOEA being optimized, at the algorithm layer, was also independently replicated 20 times for each set of parameters being evaluated. The average hypervolume from these optimizations was then used as the utility of that set of parameters.

3.2 Experimental Settings

The following paragraphs explain each row of Table 1.

Table 1. Experimental settings and corresponding choices

Experimental setting	Experimental choices
MOEA	NSGA-II, NSGA-III
Test problem	ZDT1, ZDT2, ZDT3, ZDT4, ZDT6
Function evaluations	100, 500, 2000, 3500, 5000, 6500, 8000
Number of decision variables	2, 10, 20, 30, 40

MOEAs and Test Problems The two tuned MOEAs, NSGA-II and NSGA-III, on the algorithm layer are both real-value coded. That is why the binary coded test problem ZDT5 was excluded from the experiments. All other ZDT test problems are used in this study. The reference point for the hypervolume calculations for ZDT{1, 2, 3, 6} is (11, 11) and (11, 1000) for ZDT4. The reason for the higher reference point on ZDT4 is that some of the optimizations failed to reach any solution within the (11, 11) reference point.

Both NSGA-II and NSGA-III use the SBX crossover operator and a polynomial mutation.

Function Evaluations It has been argued that keeping the parameters static during an optimization is not optimal [6]. This would also indicate that it is advantageous to use different parameters for different function evaluation budgets, even though the parameters are static during the run. In order to test this, each experiment will be performed with different budget sizes.

Number of Decision Variables A number of different rules of thumb have been proposed in the literature. For example, in a binary coded GA, the mutation rate should be proportional to the length of the chromosome [8], $p_m = 1/l$. For a real-value coded GA the length is substituted with the number of decision variables. Previous work has found this rule to be accurate on a single objective sphere problem [1]. This rule will be tested by varying the number of decision variables for each problem.

3.3 Meta-EA Parameters

At the design layer is a real-value coded meta-EA using the SBX crossover and a polynomial mutation. This introduces the problem of choosing a good set of parameters at the design layer as well. To avoid using yet another meta-EA to solve this problem, a full factorial experimental design was performed instead. The values for each parameter is shown in Table 2. To limit the runtime of these experiments only one test problem, ZDT1, was selected as the test problem on the application layer. To further limit the scope only the NSGA-II algorithm

was used at the algorithm layer. The function evaluation budget for NSGA-II on the algorithm layer was 1000 and the number of replications were 10. On the design layer the function evaluation budget for the meta-EA was 250 with 10 replications. The parameters with the highest average hypervolume was then chosen as the set of parameters to use at the design layer for the rest of the experiments. The chosen parameters are shown in the third column in Table 2.

Table 2. Full factorial experimental design for meta-EA parameter settings

Meta-EA Parameter	Possible Values	Selected
Population Size	4, 8, 16	8
Mutation Probability	0.2, 0.4, 0.6, 0.8	0.4
Mutation Distribution Index	1, 2, 5, 10, 20, 40	1
Crossover Probability	0.2, 0.4, 0.6, 0.8	0.6
Crossover Distribution Index	1, 2, 5, 10, 20, 40	40

3.4 Parameters

NSGA-II and NSGA-III have very similar parameters, the only difference is that NSGA-III does not directly specify the population size. It is instead based on the number of reference points. The reference points are systematically created by placing them on a normalized hyperplane as described in [2]. To obtain the number of reference points created by this method the following equation is used: $H = \binom{M-1+divisions}{divisions}$

1. Population size for NSGA-II (*pop*): An integer in the range [2, 300]. Upper bound determined by small scale experiments that showed all optimizations used a population size less than 300.
2. Divisions for NSGA-III (*divisions*): The number of divisions along each objective. The population size is set to exactly the number of reference points created by the divisions. An integer in the range [1, 299]. Upper bound set to 299 to get the same population size limits as for NSGA-II.
3. Mutation probability (p_m): The probability of random changes to the decision variables. A real-value in the range [0, 1].
4. Mutation Distribution Index (η_m): Index governing the proximity of the mutated child to its parent. A real-value in the range [0, 300]. Upper bound determined by small scale experiments that showed all optimizations used a η_m less than 300.
5. Crossover probability (p_c): The probability of creating offspring from parents. A real-value in the range [0, 1].

6. Crossover Distribution Index (η_c): Index governing the proximity of the mutated children to the parents. A real-value in the range [0, 300]. Upper bound determined by small scale experiments that showed all optimizations used a η_c less than 300.

The parameters of the optimization on the algorithm layer in Equation (1) become variables for the optimization on the design layer. Thus the variable vector $\mathbf{p}$ in Equation (1) is $\mathbf{p} = \{pop, p_m, \eta_m, p_c, \eta_c\}$ for NSGA-II and $\mathbf{p} = \{divisions, p_m, \eta_m, p_c, \eta_c\}$ for NSGA-III.

3.5 Performance Measure

The hypervolume measure is used to assess the performance of the EA at the algorithm layer. The hypervolume is the volume in objective space formed by a reference point and the Pareto front. The hypervolume is calculated using the technique described in [11], which also discusses the hypervolume measure in more detail. The advantage of the hypervolume measure is that provides single measure for both the convergence and spread of the solutions. The drawback is that it can be computationally expensive and that it can be sensitive to inclusion, or exclusion, of extremal points. Each EA keeps a Pareto archive of unlimited size that is used to calculated the hypervolume at the end of the optimization. So even though no limit was set for the archive size it is of course limited in practice by the available memory and running complexity of the hypervolume calculation. Neither proved to be a problem for the experiments in this study.

4 Experimental Results

This section will present the results from the experiments. Due to the large number of experiments conducted, totally 350, only a subset of all results can fit in this paper. The results for the most common problem size, 30, are shown in Table 3 and Table 4 for NSGA-II and NSGA-III respectively. The values are the median together with the standard deviation.

The experiments were run on a heterogeneous cluster of commodity hardware. In total there were 91 computers and the experiments took approximately 170 hours to complete.

4.1 Population Size

Most of the experiments found that a small population size was most optimal. Many found the smallest possible size, two, to be the best. Having a small population size increases the selection pressure since only a small amount of solutions survive each generation. Thus, allowing the optimization to advance more quickly. However, this comes at the cost of diversity among the solutions, but based on the results, the ZDT problems do not seem to require much diversity among the solutions. One reason the population size can be kept small is the fact

Table 3. Optimal parameter values for NSGA-II with $N = 30$

	Budget	HV	pop	p_m	η_m	p_c	η_c
ZDT1	100	105.46 ± 2.62	2.0 ± 33.56	0.17 ± 0.03	0.04 ± 33.24	0.64 ± 0.24	178.27 ± 101.50
	500	118.42 ± 0.06	2.0 ± 0.0	0.07 ± 0.00	0.07 ± 0.21	0.43 ± 0.11	138.81 ± 88.95
	2000	120.62 ± 0.00	2.0 ± 0.0	0.04 ± 0.00	0.15 ± 0.41	0.49 ± 0.06	1.57 ± 25.64
	3500	120.66 ± 0.00	2.0 ± 0.0	0.03 ± 0.00	0.07 ± 0.55	0.74 ± 0.08	0.33 ± 0.19
	5000	120.66 ± 0.0	2.0 ± 0.0	0.02 ± 0.00	0.12 ± 0.45	0.94 ± 0.04	0.09 ± 0.08
	6500	120.66 ± 2.84	2.0 ± 0.0	0.02 ± 0.00	0.01 ± 0.46	0.99 ± 0.01	0.05 ± 0.07
	8000	120.66 ± 2.84	2.0 ± 0.0	0.01 ± 0.00	0.06 ± 0.35	0.99 ± 0.00	0.03 ± 0.05
ZDT2	100	92.19 ± 2.78	2.0 ± 2.83	0.16 ± 0.03	0.05 ± 65.34	0.83 ± 0.25	131.68 ± 91.13
	500	114.19 ± 4.27	2.0 ± 4.79	0.07 ± 0.01	0.08 ± 64.09	0.54 ± 0.18	149.06 ± 90.18
	2000	120.25 ± 0.00	2.0 ± 0.0	0.05 ± 0.00	0.30 ± 0.42	0.44 ± 0.07	254.41 ± 78.27
	3500	120.32 ± 0.00	2.0 ± 0.0	0.03 ± 0.00	0.07 ± 0.45	0.66 ± 0.07	0.31 ± 0.18
	5000	120.33 ± 0.00	2.0 ± 0.0	0.02 ± 0.00	0.05 ± 0.71	0.93 ± 0.06	0.05 ± 0.07
	6500	120.33 ± 2.84	2.0 ± 0.0	0.02 ± 0.00	0.20 ± 0.84	0.99 ± 0.02	0.01 ± 0.03
	8000	120.33 ± 1.42	2.0 ± 0.0	0.02 ± 0.00	0.07 ± 0.67	0.99 ± 0.00	0.01 ± 0.02
ZDT3	100	110.31 ± 1.58	2.0 ± 1.74	0.17 ± 0.13	0.06 ± 20.89	0.77 ± 0.22	107.82 ± 85.04
	500	125.72 ± 0.20	2.0 ± 0.0	0.07 ± 0.01	0.05 ± 0.34	0.59 ± 0.13	93.57 ± 63.37
	2000	128.69 ± 0.00	2.0 ± 0.0	0.05 ± 0.01	1.48 ± 0.86	0.62 ± 0.09	207.58 ± 127.42
	3500	128.76 ± 0.00	2.0 ± 0.0	0.03 ± 0.00	0.31 ± 0.88	0.85 ± 0.07	0.16 ± 0.11
	5000	128.77 ± 0.00	2.0 ± 0.6	0.02 ± 0.00	0.63 ± 2.90	0.98 ± 0.02	0.03 ± 0.62
	6500	128.77 ± 0.00	3.5 ± 1.92	0.02 ± 0.00	0.05 ± 1.17	0.99 ± 0.00	1.92 ± 2.40
	8000	128.77 ± 0.0	6.0 ± 2.70	0.02 ± 0.01	0.25 ± 39.46	0.99 ± 0.02	3.27 ± 7.72
ZDT4	100	7874.47 ± 40.69	2.0 ± 0.86	0.10 ± 0.03	5.23 ± 97.64	0.98 ± 0.05	28.36 ± 100.76
	500	9534.32 ± 218.76	2.0 ± 4.15	0.05 ± 0.03	3.46 ± 130.49	0.94 ± 0.11	33.89 ± 81.29
	2000	10233.5 ± 230.31	33.5 ± 18.85	0.03 ± 0.01	67.14 ± 119.39	0.99 ± 0.07	41.67 ± 13.91
	3500	10644.55 ± 177.93	29.5 ± 33.41	0.02 ± 0.01	26.21 ± 132.47	0.99 ± 0.14	41.89 ± 33.14
	5000	10898.0 ± 147.19	4.0 ± 45.68	0.02 ± 0.01	7.73 ± 125.00	0.99 ± 0.16	39.21 ± 58.90
	6500	10945.1 ± 99.07	3.0 ± 44.47	0.02 ± 0.01	9.14 ± 95.20	0.94 ± 0.20	26.79 ± 16.83
	8000	10960.95 ± 99.50	4.5 ± 62.07	0.02 ± 0.01	10.35 ± 115.74	0.99 ± 0.14	33.41 ± 15.60
ZDT6	100	43.81 ± 0.34	2.0 ± 0.0	0.18 ± 0.03	0.03 ± 0.13	0.73 ± 0.16	142.08 ± 103.52
	500	70.17 ± 0.41	2.0 ± 0.0	0.07 ± 0.01	0.01 ± 0.22	0.53 ± 0.18	79.68 ± 91.39
	2000	106.75 ± 0.19	2.0 ± 0.0	0.05 ± 0.00	0.02 ± 0.15	0.40 ± 0.07	128.93 ± 77.69
	3500	114.81 ± 0.04	2.0 ± 0.0	0.05 ± 0.00	0.11 ± 0.27	0.32 ± 0.05	223.53 ± 86.08
	5000	116.17 ± 0.01	2.0 ± 0.0	0.05 ± 0.00	0.10 ± 0.41	0.28 ± 0.05	209.40 ± 89.19
	6500	116.37 ± 0.00	2.0 ± 0.0	0.05 ± 0.00	0.08 ± 0.43	0.30 ± 0.05	16.30 ± 42.55
	8000	116.40 ± 0.00	2.0 ± 0.0	0.04 ± 0.00	0.24 ± 0.87	0.38 ± 0.05	3.42 ± 35.33

Table 4. Optimal parameter values for NSGA-III with $N = 30$

	Budget	HV	$divisions$	p_m	η_m	p_c	η_c
ZDT1	100	105.02 ± 3.98	1.0 ± 88.43	0.20 ± 0.14	0.08 ± 48.35	0.83 ± 0.27	151.25 ± 100.56
	500	118.30 ± 0.06	1.0 ± 0.0	0.07 ± 0.01	0.06 ± 0.22	0.88 ± 0.11	95.01 ± 98.37
	2000	120.62 ± 0.00	1.0 ± 0.0	0.04 ± 0.00	0.15 ± 0.45	0.97 ± 0.05	1.60 ± 68.94
	3500	120.66 ± 0.00	1.0 ± 0.0	0.02 ± 0.00	0.20 ± 0.87	0.99 ± 0.00	0.23 ± 0.21
	5000	120.66 ± 2.84	1.0 ± 0.49	0.02 ± 0.00	0.18 ± 0.76	0.99 ± 0.02	0.32 ± 0.34
	6500	120.66 ± 0.0	2.0 ± 0.43	0.02 ± 0.00	0.08 ± 0.34	0.99 ± 0.01	0.27 ± 0.20
	8000	120.66 ± 2.84	2.0 ± 0.0	0.02 ± 0.00	0.46 ± 0.84	0.99 ± 0.00	0.22 ± 0.16
ZDT2	100	91.46 ± 3.43	1.0 ± 64.72	0.17 ± 0.03	0.01 ± 60.35	0.92 ± 0.24	148.35 ± 96.71
	500	114.01 ± 0.14	1.0 ± 0.0	0.06 ± 0.00	0.00 ± 0.09	0.92 ± 0.06	142.13 ± 78.08
	2000	120.24 ± 0.00	1.0 ± 0.0	0.05 ± 0.00	0.11 ± 0.50	0.94 ± 0.09	230.93 ± 78.70
	3500	120.32 ± 0.00	1.0 ± 0.0	0.03 ± 0.00	0.07 ± 0.54	0.99 ± 0.02	0.32 ± 0.17
	5000	120.33 ± 0.00	1.0 ± 0.21	0.02 ± 0.00	0.11 ± 1.15	0.99 ± 0.00	0.11 ± 0.19
	6500	120.33 ± 2.84	2.0 ± 0.45	0.02 ± 0.00	0.45 ± 2.65	0.99 ± 0.01	0.16 ± 0.23
	8000	120.33 ± 2.84	2.0 ± 0.35	0.02 ± 0.00	0.43 ± 1.02	0.99 ± 0.00	0.13 ± 0.14
ZDT3	100	109.38 ± 5.39	1.0 ± 106.74	0.19 ± 0.25	0.13 ± 109.57	0.83 ± 0.34	157.28 ± 86.58
	500	125.57 ± 0.11	1.0 ± 0.0	0.08 ± 0.01	0.24 ± 0.28	0.95 ± 0.15	82.53 ± 56.98
	2000	128.68 ± 0.00	1.0 ± 0.0	0.05 ± 0.01	0.60 ± 0.75	0.97 ± 0.06	11.93 ± 91.16
	3500	128.76 ± 0.00	1.0 ± 0.55	0.02 ± 0.01	0.51 ± 1.86	0.99 ± 0.05	0.23 ± 12.31
	5000	128.77 ± 0.00	2.0 ± 0.80	0.02 ± 0.01	1.22 ± 14.54	0.99 ± 0.03	0.85 ± 2.68
	6500	128.77 ± 0.00	2.0 ± 1.04	0.03 ± 0.01	17.44 ± 40.01	0.99 ± 0.00	0.01 ± 2.43
	8000	128.77 ± 0.00	4.0 ± 2.23	0.02 ± 0.01	0.39 ± 86.62	0.99 ± 0.01	5.81 ± 27.95
ZDT4	100	7808.50 ± 103.39	1.0 ± 0.92	0.10 ± 0.11	6.64 ± 111.21	0.98 ± 0.14	42.87 ± 108.69
	500	9117.17 ± 253.37	8.5 ± 4.40	0.10 ± 0.05	264.02 ± 129.77	0.99 ± 0.01	97.43 ± 107.31
	2000	10420.05 ± 217.01	13.5 ± 15.15	0.03 ± 0.01	32.66 ± 129.23	0.99 ± 0.07	42.41 ± 30.90
	3500	10662.05 ± 173.42	22.5 ± 24.28	0.02 ± 0.01	124.65 ± 140.48	0.99 ± 0.03	33.41 ± 35.91
	5000	10893.45 ± 144.74	2.0 ± 33.95	0.02 ± 0.01	7.56 ± 139.79	0.99 ± 0.06	43.74 ± 17.90
	6500	10942.75 ± 117.76	1.0 ± 42.92	0.02 ± 0.01	8.81 ± 130.25	0.99 ± 0.11	44.83 ± 16.98
	8000	10959.35 ± 98.00	1.0 ± 48.80	0.02 ± 0.01	9.50 ± 113.74	0.99 ± 0.09	39.72 ± 17.57
ZDT6	100	43.41 ± 3.90	1.0 ± 97.06	0.19 ± 0.09	0.15 ± 66.85	0.92 ± 0.31	110.77 ± 81.40
	500	69.93 ± 0.45	1.0 ± 0.0	0.08 ± 0.01	0.03 ± 0.18	0.89 ± 0.16	82.35 ± 79.93
	2000	106.27 ± 0.21	1.0 ± 0.0	0.05 ± 0.00	0.07 ± 0.13	0.69 ± 0.16	144.05 ± 73.31
	3500	114.63 ± 0.08	1.0 ± 0.0	0.05 ± 0.00	0.14 ± 0.33	0.58 ± 0.12	167.48 ± 88.15
	5000	116.13 ± 0.01	1.0 ± 0.0	0.05 ± 0.00	0.08 ± 0.28	0.58 ± 0.07	161.48 ± 77.15
	6500	116.36 ± 0.00	1.0 ± 0.0	0.05 ± 0.00	0.16 ± 0.92	0.61 ± 0.08	23.18 ± 94.26
	8000	116.40 ± 0.00	1.0 ± 0.0	0.05 ± 0.00	0.25 ± 0.68	0.78 ± 0.11	6.02 ± 2.21

that the hypervolume is calculated from the, unlimited, Pareto archive. Using the last generation to calculate the hypervolume would in most cases result in a smaller hypervolume, since fewer solutions would be used in the calculation. An exception to the small population size is experiments with $N = 2$. This is especially true when the budget size is 100. One reason for this might be that a random search, which both NSGA-II and NSGA-III degenerates to when the population size is greater or equal to the budget, has about the same performance when the problem is easy to solve and the function evaluation budget is very limited.

The median values for the population size parameter using NSGA-II are shown in Table 5. The same observations can be made for the NSGA-III and are not shown for that reason.

4.2 Mutation Probability: $1/N$ Rule of Thumb

Each experiment was run with five different values for the number of decision variables. This was done to test the accuracy of the rule of thumb that the mutation rate should be set to one divided by the number decision variables. Since each experiment was also run with different function evaluation budgets, it is also possible to see if that had any affect on the mutation probability. The usefulness of this evaluation is limited by the small number of problems used in this paper and no generalization can be made how this rule works on other problems. The mutation probabilities are also only from the best set of parameters found. Therefore, this evaluation does not test the accuracy of this rule of thumb for sub-optimal sets of parameters.

The experiment results can be divided into two groups based on the relationship between the mutation probability and the number of variables. ZDT{1, 2, 3} is in one group and ZDT{4, 6} is in the other. The first group start with a relatively high mutation probability for two variables, which then decreases and is kept almost constant for 10, 20, 30 and 40 variables. The second group has a more gradual decrease in the mutation probability. The median values from two problems are shown here, ZDT1 in Figure 1 and ZDT4 in Figure 2.

On ZDT1 the rule slightly overestimates the mutation probability for budget sizes greater than 500 when N is 10 and 20 because the optimized mutation probability does not change much between N 10 and 40. It also underestimates for all N when the budget size is less than 2000. On ZDT4 the rule overestimates when N is 2 and the budget size is greater than 500. It also underestimates for all N when the budget size is 100. For all other cases the rule matches well with the data.

Based on these results the rule of thumb is able to estimate good values for the mutation probability, especially for larger budgets, on the ZDT test problems.

4.3 Mutation Probability vs Budget Sizes

A trend that can be observed throughout all experiments is that mutation probability decreases as the function evaluation budget increases. The trend is most

Table 5. Experimental results for population sizes in NSGA-II

|	N	100	500	2000	3500	5000	6500	8000
ZDT1	2	79.5 ± 121.99	11.0 ± 3.02	12.0 ± 5.83	15.0 ± 6.70	15.5 ± 6.27	20.0 ± 4.48	22.0 ± 6.66
	10	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.8
	20	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
	30	2.0 ± 33.56	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
	40	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
ZDT2	2	2.0 ± 100.89	6.0 ± 1.65	17.5 ± 2.24	18.5 ± 6.49	22.0 ± 4.43	26.5 ± 4.02	28.0 ± 4.57
	10	2.0 ± 3.26	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.43
	20	2.0 ± 59.58	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
	30	2.0 ± 2.83	2.0 ± 4.79	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
	40	2.0 ± 1.74	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
ZDT3	2	4.0 ± 119.75	4.5 ± 2.53	14.0 ± 5.20	32.0 ± 8.46	37.5 ± 7.40	48.5 ± 24.04	56.0 ± 34.45
	10	2.0 ± 3.31	2.0 ± 0.0	4.0 ± 1.90	9.0 ± 2.71	11.0 ± 1.10	11.5 ± 0.78	11.0 ± 3.28
	20	2.0 ± 2.71	2.0 ± 0.0	2.0 ± 0.73	2.0 ± 1.69	4.5 ± 2.53	9.0 ± 3.28	10.0 ± 2.25
	30	2.0 ± 1.74	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.6	3.5 ± 1.92	6.0 ± 2.70
	40	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.43	2.0 ± 0.0	2.0 ± 0.87
ZDT4	2	3.0 ± 110.51	2.0 ± 15.87	8.0 ± 4.01	8.0 ± 2.47	12.0 ± 3.46	12.0 ± 2.78	17.0 ± 5.01
	10	2.5 ± 1.81	2.0 ± 8.92	3.0 ± 30.32	3.0 ± 28.57	5.0 ± 56.69	6.0 ± 35.50	11.0 ± 85.31
	20	2.5 ± 1.57	12.0 ± 6.07	4.0 ± 23.04	4.0 ± 34.12	4.0 ± 50.59	6.5 ± 60.39	6.0 ± 50.64
	30	2.0 ± 0.86	2.0 ± 4.15	33.5 ± 18.85	29.5 ± 33.41	4.0 ± 45.68	3.0 ± 44.47	4.5 ± 62.07
	40	2.5 ± 0.97	2.0 ± 2.62	15.0 ± 16.91	2.5 ± 28.88	77.0 ± 40.10	4.0 ± 38.47	6.0 ± 59.35
ZDT6	2	2.0 ± 79.04	2.0 ± 0.53	11.0 ± 2.94	20.0 ± 5.35	23.5 ± 4.63	26.0 ± 3.62	29.0 ± 5.31
	10	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.6	2.0 ± 1.39
	20	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
	30	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
	40	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
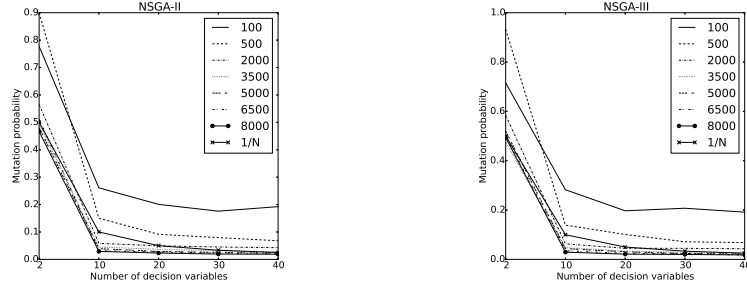
Fig. 1. Applicability of 1/N rule of thumb for p_m on ZDT1
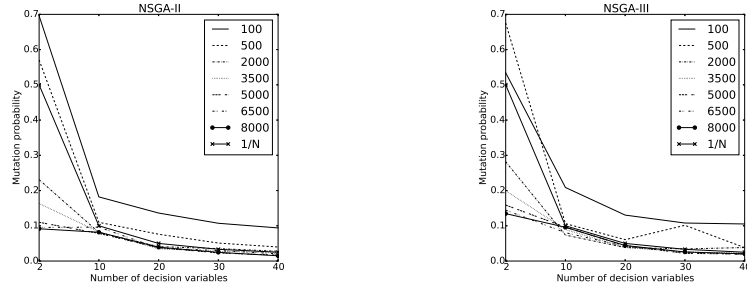

Fig. 2. Applicability of 1/N rule of thumb for p_m on ZDT4
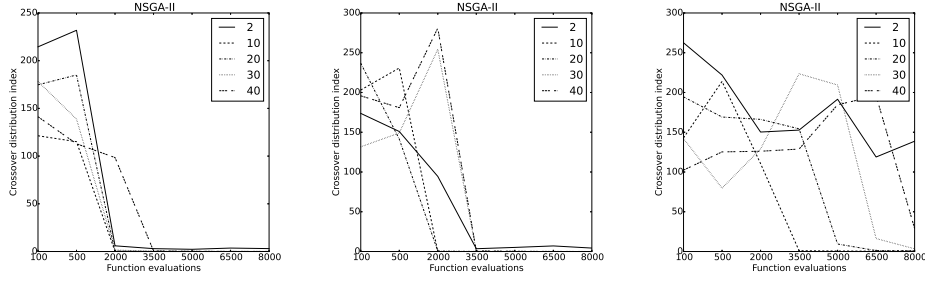
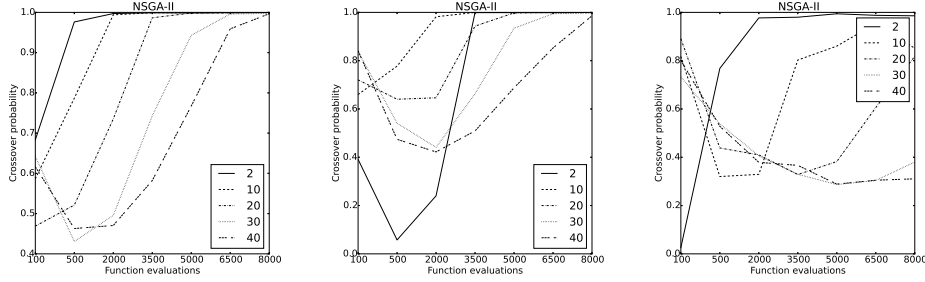**Fig. 3.** Trends with budget size and η_c on ZDT1, ZDT2 and ZDT6

Fig. 4. Trends with budget size and p_c on ZDT1, ZDT2 and ZDT6

prominent on ZDT$\{1, 2, 3\}$ and less so on ZDT$\{4, 6\}$. Figure 5 and 6 shows the median values for NSGA-II on ZDT1 and ZDT6. The results for NSGA-III are similar, but they are not shown here because of space limitations.

4.4 Budget Size, p_c and η_c

A trend how p_c changes with respect to the budget size can be observed on all problems except ZDT4, although it is most clear on ZDT$\{1, 2, 3\}$. For small budgets p_c is relatively high. As the budget size increases p_c first falls and then rises, approaching a value of one. The point at which it starts to rise is related to the number of decision variables. Another trend is that there is a point at which an increase of the budget size causes a sharp fall of η_c. One explanation for why p_c is high and η_c is low for large budget sizes is based on the fact that most experiments use a population size of two. The two individuals are pushed apart by the crowding distance and with a large enough budget they will end up at each of the two extreme values. The rest of the non-dominated front is then filled by crossing these two solutions, and since they are at opposite extremes a low η_c is preferred. These trends are shown for NSGA-II on ZDT1, ZDT2 and ZDT6 in Figure 3 and Figure 4, the values are the median.

4.5 Hypervolume Comparisons Between NSGA-II and NSGA-III

The mean hypervolume values for both NSGA-II and NSGA-III are shown in Table 6. It is not possible, due to the number of experiments performed, to
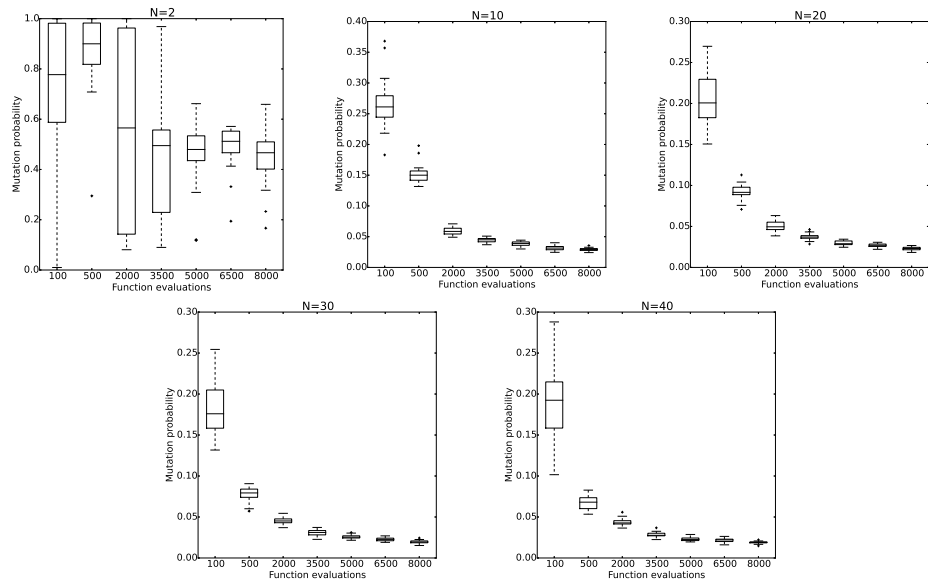
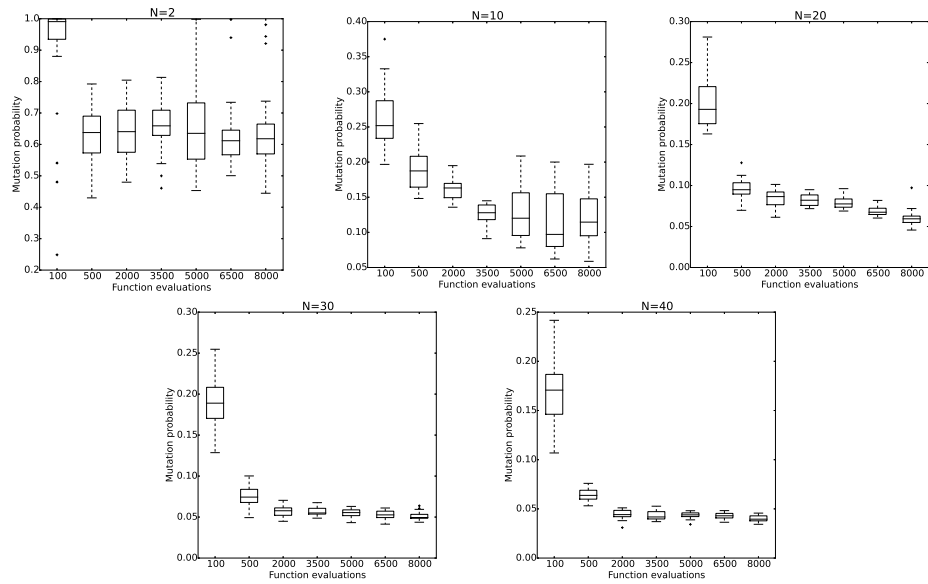**Fig. 5.** Mutation probabilities with varying budget sizes for NSGA-II on ZDT1

Fig. 6. Mutation probabilities with varying budget sizes for NSGA-II on ZDT6

Table 6. Hypervolume results for NSGA-II and NSGA-III

	N	ZDT1 NSGA-II	ZDT1 NSGA-III	ZDT2 NSGA-II	ZDT2 NSGA-III	ZDT3 NSGA-II	ZDT3 NSGA-III
100	2	119.68	119.43	**118.58**	116.73	126.41	125.58
	10	**115.35**	111.60	107.64	105.04	119.34	117.46
	20	**109.27**	106.38	95.78	96.77	**112.78**	108.35
	30	104.92	103.35	91.39	90.65	**110.06**	104.97
	40	**103.06**	101.24	87.85	86.87	**108.01**	102.54
500	2	120.65	**120.66**	120.32	**120.32**	128.76	**128.76**
	10	**120.53**	120.51	**120.11**	120.08	**128.50**	128.46
	20	**119.77**	119.69	**117.87**	117.77	**127.41**	127.27
	30	**118.43**	118.29	113.22	114.00	**125.71**	125.55
	40	**116.88**	116.64	110.72	109.39	**123.96**	122.75
2000	2	120.66	**120.66**	120.33	120.33	128.77	128.77
	10	**120.66**	120.66	**120.33**	120.32	**128.77**	128.77
	20	**120.65**	120.65	**120.31**	120.31	**128.74**	128.74
	30	120.62	120.62	**120.25**	120.24	**128.69**	128.68
	40	**120.55**	120.55	**120.14**	120.12	**128.61**	128.60
3500	2	120.66	120.66	120.33	120.33	128.77	128.77
	10	**120.66**	120.66	120.33	120.33	128.77	128.77
	20	**120.66**	120.66	**120.33**	120.33	**128.77**	128.77
	30	**120.66**	120.66	**120.32**	120.32	**128.76**	128.76
	40	**120.65**	120.65	120.31	120.31	128.75	128.75
5000	2	120.66	120.66	120.33	120.33	128.77	128.77
	10	120.66	120.66	120.33	120.33	128.77	128.77
	20	**120.66**	120.66	120.33	120.33	128.77	**128.77**
	30	**120.66**	120.66	**120.33**	120.33	**128.77**	128.77
	40	**120.66**	120.66	**120.32**	120.32	**128.77**	128.77
6500	2	120.66	120.66	120.33	120.33	128.77	128.77
	10	120.66	120.66	120.33	120.33	**128.77**	128.77
	20	120.66	120.66	**120.33**	120.33	128.77	128.77
	30	**120.66**	120.66	120.33	120.33	128.77	128.77
	40	120.66	120.66	**120.33**	120.33	**128.77**	128.77
8000	2	120.66	120.66	120.33	120.33	128.77	128.77
	10	120.66	120.66	120.33	120.33	128.77	**128.77**
	20	120.66	120.66	120.33	120.33	128.77	**128.77**
	30	120.66	120.66	**120.33**	120.33	128.77	128.77
	40	**120.66**	120.66	120.33	120.33	**128.77**	128.77

	N	ZDT4 NSGA-II	ZDT4 NSGA-III	ZDT6 NSGA-II	ZDT6 NSGA-III
100	2	**10993.44**	10990.83	109.83	106.26
	10	**10461.61**	10411.53	63.63	61.04
	20	**9251.83**	9192.64	**49.29**	47.53
	30	**7889.13**	7781.64	**43.82**	41.48
	40	**6391.02**	6289.29	40.93	40.29
500	2	10999.46	10999.46	**116.41**	116.41
	10	10849.93	10828.18	**103.26**	102.69
	20	10221.37	10247.87	**82.07**	81.78
	30	**9447.10**	9273.20	70.23	70.03
	40	8524.69	8408.60	**63.34**	63.06
2000	2	10999.7	10999.69	116.42	**116.42**
	10	10970.92	10973.84	**116.35**	116.34
	20	10778.65	10793.25	**113.59**	113.37
	30	10416.11	10425.08	**106.76**	106.31
	40	9968.78	9899.27	**99.29**	98.87
3500	2	10999.7	10999.7	116.43	116.43
	10	10991.61	10984.64	**116.41**	116.41
	20	10896.30	10911.16	**116.24**	116.21
	30	10656.69	10666.82	**114.80**	114.65
	40	10402.90	10438.98	**111.48**	111.19
5000	2	10999.7	10999.7	116.43	116.43
	10	10993.38	10993.26	**116.42**	116.41
	20	10931.79	10946.89	**116.40**	116.39
	30	10779.01	10780.29	**116.17**	116.13
	40	10498.14	10557.02	**115.19**	115.03
6500	2	10999.7	10999.7	116.43	116.43
	10	**10997.87**	10994.22	116.42	116.42
	20	10945.27	10956.11	**116.41**	116.41
	30	10896.20	10849.03	**116.37**	116.36
	40	10787.64	10712.04	**116.11**	116.06
8000	2	10999.7	10999.7	116.43	116.43
	10	10995.99	10998.15	116.42	116.42
	20	10977.54	10966.47	**116.41**	116.41
	30	10887.89	10889.35	**116.40**	116.40
	40	10752.03	10840.82	**116.34**	116.32

include all parameter settings used to obtain the hypervolume results. A subset of all the parameter settings and their corresponding hypervolumes are presented in Table 3 and Table 4. A Welch-t test with a significance of 5% is performed to determine if the two samples, NSGA-II and NSGA-III, are statistical different. If the null hypothesis can be rejected, the greater hypervolume is shown in bold.

The difference in hypervolume between NSGA-II and NSGA-III is for the most part small. However, for some of the experiments, NSGA-II is slightly better. NSGA-III is statistically better on some experiments but the difference is too small to be concluded as significant.

To summarize, NSGA-II is found to be marginally better than NSGA-III on the ZDT problems. Both NSGA-II and NSGA-III can find solutions very close to the Pareto front for ZDT{1, 2, 3, 6}. The most difficult problem is ZDT4, for which with $N > 10$ none of algorithms could reach the maximum theoretical hypervolume within 8000 evaluations.

5 Conclusions and Further Work

This paper utilized a bilevel optimization framework to find optimal parameter values for two different MOEAs, namely NSGA-II and NSGA-III, for maximal performance on the ZDT test suite. Both the number of decision variables and the function evaluation budgets were simultaneously varied to determine how they affect the optimal parameter settings for the respective algorithm. This made it possible to test the rule of thumb that the mutation probability should be set to $1/N$. The results show that, on the ZDT test problems, this rule is a good heuristic.

The experiments also made it possible to see what affect the different function evaluation budgets has on the optimized parameters. An important observation was that the optimal mutation probability is not only dependent on the number of decision variables but also on the available budget size. Specifically, it was observed that the optimal mutation probability decreases with increasing budget.

It was also clear from the results that the ZDT test problems do not require much diversity in the population because most experiments found the optimal population size to be less than 10, often close to the minimum of just two individuals. This also indicates that a parameter controlling the elitism should have been included in the experiments.

Another aim of this paper was to compare the performance between NSGA-II and NSGA-III on the ZDT test problems. From the results, it is possible to discern a slight advantage with NSGA-II over NSGA-III on the ZDT problems. As far as the optimal parameter values are concerned, it was observed that the differences are small.

Extending these experiments to scale the number of objectives instead of the number of decision variables would be interesting, and is intended as future work. Since these results, as well as other earlier work, indicate that it is sub-optimal to keep parameter settings static during the run, it would be be worthwhile to modify an EA, on the algorithm layer, to be able to use multiple sets of

parameters during an optimization. This would allow a meta-EA to tune multiple sets of parameters at different intervals of the optimization, instead of being limited to a single set throughout the optimization run.

References

1. Bäck, T.: Parallel optimization of evolutionary algorithms. In: Parallel Problem Solving From Nature. pp. 418–427. Spring-Verlag (1994)
2. Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM Journal on Optimization 8(3), 631–657 (Aug 1998)
3. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation 18(4), 577–601 (Aug 2014)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
5. Eiben, A.E., Smit, S.K.: Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm and Evolutionary Computation 1(1), 19–31 (Mar 2011)
6. Eiben, A., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3(2), 124–141 (Jul 1999)
7. Grefenstette, J.: Optimization of control parameters for genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics 16(1), 122–128 (Jan 1986)
8. Mühlenbein, H.: How genetic algorithms really work: Mutation and hillclimbing. In: PPSN. pp. 15–26 (1992)
9. Ugolotti, R., Cagnoni, S.: Analysis of evolutionary algorithms using multi-objective parameter tuning. In: Proceedings of the 2014 Conference on Genetic and Evolutionary Computation. pp. 1343–1350. GECCO '14, ACM, New York, NY, USA (2014)
10. Wessing, S., Beume, N., Rudolph, G., Naujoks, B.: Parameter tuning boosts performance of variation operators in multiobjective optimization. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) Parallel Problem Solving from Nature, PPSN XI, pp. 728–737. No. 6238 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2010)
11. While, L., Bradstreet, L., Barone, L.: A fast way of calculating exact hypervolumes. IEEE Transactions on Evolutionary Computation 16(1), 86–95 (Feb 2012)
12. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Shaker Verlag (1999)
13. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput. 8(2), 173–195 (Jun 2000)