Elitism in Multiobjective Hierarchical Strategy [⋆]

Michał Idzik[0000−0002−8446−4966], Radosław Łazarz[0000−0002−3151−9759], and Aleksander Byrski[0000−0001−6317−7012]

AGH University of Science and Technology {miidzik,lazarz,olekb}@agh.edu.pl

Abstract. The paper focuses on complex metaheuristic algorithms, namely multi-objective hierarchical strategy, which consists of a dynamically evolving tree of interdependent demes of individuals. The main contribution presented in this paper is the introduction of elitism in a form of an archive, locally into the demes and globally into the whole tree and developing necessary updates between them. The newly proposed algorithms (utilizing elitism) are compared with their previous versions as well as with the best state of the art multi-objective metaheuristics.

Keywords: multi-objective optimization · hierarchical metaheuristics · elitist multi-objective evolutionary algorithms.

1 Introduction

Contradictory objectives are present in the optimization area and such problems are much harder to solve than global-optimization ones. Their difficulty is even increased, as we have to seek not only one optimum but a whole set of Pareto-optimal solutions. We have to try to distribute those solutions evenly, make sure that we do not lose good solutions between the subsequent approximations of the Pareto-front, especially when the number of those functions becomes higher than several (many-objective problems). Natural weapons of choice to deal with such problems are metaheuristic algorithms.

One of the very interesting and successful metaheuristics is Hierarchic Genetic Search, introduced by Schaefer and Kolodziej [22]. It assumes a very strict structuralization of the sub-populations in a form of a tree. Particular demes are created when interesting solutions are found and the parameters of the search are adapted (the search grid is modified). The demes can sprout other demes. Of course, the demes can also be removed so the whole structure is dynamic.

HGS algorithm was successfully applied to solving global optimization problems, however, later it was adapted by Łazarz et al. [17] to multi-objective problems and further developed in this area. In this paper we want to present a subsequent, important development of the HGS-related multi-objective metaheuristics, introducing the notion of elitism, by applying local and global archives into HO-mHGS, which step is necessary in to retain good solutions in the current frontier.

[⋆] The research presented in this paper was financed by Polish National Science Centre PRELUDIUM project no. 2017/25/N/ST6/02841.

The next section of this paper shows the most important aspects of elitism in the area of multi-objective optimization, then the multi-objective hierarchical strategy is discussed along with the presentation of the aspects of local and global archives, then the experimental results are presented, showing not only the relation between newly introduced algorithms and its older versions but also referring to the most known state of the art algorithms, like NSGAII.

2 Elitism in multi-objective evolutionary algorithms

Elitism, in a broad sense, is a mechanism enhancing evolutionary computation. It works by ensuring that the current best individuals have a higher chance of remaining in the population throughout the iterations (or being otherwise involved in the search process). As a result, it guarantees evolution towards the objective function improvement (or at least prevents deterioration to inferior solutions) [10]. On the other hand, it may decrease the population diverseness due to increased selection pressure.

There have been numerous attempts at implementing this paradigm in practice. One approach is to simply combine the population with its offspring and then apply the survival selection [20]. Its variation, known as multi-elitism [1], instead keeps in population the so-called residents — individuals occupying separate peaks of the fitness function. Another option, popular in the case of multi-objective problems, is to employ a relaxed version of the idea (termed controlled elitism [6]), where solutions are first divided into an ordered sequence of subsets (called fronts), and then each of them contributes to the next generation several individuals based on geometric distribution. Finally, δ-similar elimination [21] procedure can be used to filter the joint parent-offspring population and ensure that no two individuals are within a given distance from one another, thus giving an advantage to more exploratory solutions. All those variants have one thing in common: they strive to preserve the elitism benefits while simultaneously trying to counteract its deficiencies.

An alternative to those techniques is to store the chosen individuals, not in the working population, but in a separate set, usually referred to as archive [8]. Specific optimisation models tend to differ by both their organisation and how archived organisms are utilised during the run. PAES [14] keeps an archive of non-dominated solutions with additional crowding-based thinning. Its contents are used in two situations: to estimate the true dominance ranking in a pair of otherwise incomparable solutions, and to be returned as the final outcome. MOIPSO [26] modifies this scheme by making use of the distribution entropy as a crowding measure, introducing Gaussian mutation throw points as a way of improving the external archive uniformity, and applying it to a particle swarm optimisation technique. A similar archive system is also successfully adopted in the case of chaos optimisation algorithms, such as MPCOA [18]. ME-MOEA/D [9] (a decomposition-based method) identifies elite solutions based on a dominance index, fine-tunes them using a local search mechanism, and transfers them to the isolated external population afterwards. This supplementary population is then used as a potential substitute source of individuals during the crossover operation. Other notable examples of effective archive integration are the classical algorithms PESA-II [4] and SPEA2 [31].

Naturally, certain solutions elude being classified by such a simple dichotomy. For example, BBO [25] is an algorithm utilising a multi-population island-based computation scheme, with organisms representing solutions emigrating from and immigrating to the said islands. Its elitism is enforced by prohibiting migration for the top portion of individuals in each subset. Other researchers experiment with unbounded archives and propose novel frameworks for assembling the final result from the subset of all examined solutions [13]. Lastly, some studies explore the potential of methods that completely abstain from employing any forms of elitism whatsoever, demonstrating that they are still able to perform significantly well on particular types of problems [27].

3 Optimizing MO-mHGS with archives

3.1 Multi-Objective Hierarchical Strategy

Multi-Objective Hierarchical Strategy (MO-HGS) [2] is a framework designed to adapt and run multi-objective evolutionary algorithms (MOEAs) in an environment split into several populations (called *demes*). MO-HGS dynamically creates population nodes and lays them out in a tree-like hierarchy. Each node runs an independent instance of MOEA (denoted as node's *driver*) and its search accuracy depends on its depth in the hierarchy. Nodes closer to the root perform a more chaotic search to find promising areas.

The process of MO-HGS consists of several steps, called *metaepochs*. In each metaepoch driver runs several epochs, progressing in an evolutionary process. Additionally, HGS-specific mechanics are applied. The most promising individuals (*delegates*) of each node have a chance to become seeds of next-level *child nodes* (*sprouting* procedure). A child node runs with reduced variance settings so that its population will mostly explore that region. To eliminate the risk of redundant exploration of independently evolving demes, two trimming procedures are performed. If the area is already explored by one of the other children, then sprouting is cancelled and the next candidate for sprouting is considered (*branch comparison*). Moreover, after each metaepoch MO-HGS checks if populations at the same level of the tree perform a search in the common landscape region or already explored regions and removes such nodes (*branch reduction*).

MO-HGS was inspired by a more general hierarchical scheme, Hierarchical Genetic Strategy [23], which was adapted into a multi-objective environment. Further improvements were made to the model, including the hypervolume-based sprouting procedure introduced in [17]. The result meta-model was denoted as Multiobjective Optimization Hierarchic Genetic Strategy with maturing (MO-mHGS).

With MO-mHGS we focus mainly on reducing the cost of fitness evaluation. We proved that this meta-model can be a perfect tool to solve problems with a high cost of the evaluation. It can be achieved by providing fitness operators with different accuracies bound to specific HGS tree levels. However, in the recent research [12] we have shown that even without this fitness adjusting mechanism, MOEA enforced by MO-mHGS can achieve better performance comparing to the situation when it's run alone. It is possible thanks to the natural parallel execution capabilities of the hierarchical model.

In this paper, we continue the exploration of MO-mHGS features as a generic tool improving the performance of its driver (preferably: any MOEA). This time we will consider two approaches to elitism in the MO-mHGS meta model: global fitness archive and local node-level archives. We expect it should lead to better convergence and more stable results, regardless of the incorporated driver.

3.2 Global Fitness Archive in MO-mHGS

In each metaepoch MO-mHGS delegates the work to its internal driver of the given tree node. As a result, it obtains a new state of all demes that can be used in further procedures (sprouting, reduction). These populations are completely independent which by design (combined with redundancy reduction procedures) leads to better coverage of multiple different areas. But, at the same time, we lose information about solutions found during the process, that might be valuable but were abandoned with simulation progress. Furthermore, we strongly rely on a driver ability to provide any elitism mechanics and compose the final MO-mHGS result as a union of final populations of all nodes. Such result population may be large in the case of broader HGS trees. It is unnecessary – there's no guarantee that a larger set of solutions is better in terms of individuals distribution over a Pareto front. To address these issues we modified our MO-mHGS model by adding a global fitness archive located in the nodes supervisor.

This MO-mHGS variant (denoted as *Multi-Objective Elitist Hierarchical Genetic Strategy*, MO-EHGS or EHGS) offers a new metaepoch procedure presented in Listing 1. Each alive (still progressing and not redundant) node invokes several steps of its driver and produces finalized population. How the population is finalized is driver-dependent – it may be similar to the delegates set described in the following section. Solutions from the population are inserted into the archive. Eventually, the archive is truncated according to the comparison operator. In our experiments, we use the crowding distance attribute from fast non-dominated sorting to compare results. Note that solutions from dead nodes can be still present in the final result if they are strong enough to prevail.

Additionally, in simulations driven by a budget of allowed fitness evaluations number, it is important to preserve fair assumptions – solutions are added to the archive only if node execution does not exceed the budget. Finally, fitness-based archive size can be easily constrained so we can now control the maximum size of MO-EHGS result front approximation. In our experiments, we set maximum archive size $|A|$ depending on a number of a problem's objectives k. According to the rules set in the CEC09 competition [29]: $|A| = 100$ for $k = 2$, $|A| = 150$ for $k = 3$ and $|A| = 800$ for $k = 5$.

Algorithm 1 MO-EHGS metaepoch procedure with elitism support

```
Require:
    $hgsNodes$, list of all HGS nodes
    $archive$, fitness based archive
    $budget$, maximum allowed evaluation cost
    $totalCost < budget$, current cost of HGS simulation

1: for each $node$ in $hgsNodes$ do:
2:     if $\textsc{isAlive}(node)$ then
3:         $totalCost \leftarrow totalCost + \textsc{runMetaepoch}(node)$
4:         if $\textsc{isBudgetMet}(budget, totalCost)$ then
5:             break
6:         end if
7:         $nodePopulation \leftarrow \textsc{finalPopulation}(node)$
8:         $archive \leftarrow archive \cup nodePopulation$
9:     end if
10: end for
11: $\textsc{truncate}(archive, \textsc{CrowdingDistanceOperator}())$
```

3.3 Local archives

During our previous research, we observed OMOPSO [24] algorithm was the driver most significantly improved by MO-mHGS. OMOPSO is one of the particle swarm optimizers adjusted to solve multi-objective problems. This specific observation has lead us to the conclusion that characteristics of OMOPSO could be applied in the general scheme of the MO-HGS model to recreate these phenomena in other applications. The crucial part of OMOPSO algorithm is the way it handles elitism. There are two layers of best individual storages: swarm leaders archive and global ϵ-archive. The latter is based on the concept of ϵ-dominance [16]. A search space vector X_1 ϵ-dominates solution X_2 when: $f_i(X_1)/(1+\epsilon) \leq f_i(X_2), \forall_{i=1...k}$ and $f_i(X_1)/(1+\epsilon) < f_i(X_2), \exists i \in 1..k$, where $f_1, ..f_k$ are objective functions of k-objective problem. The value of ϵ should be greater than 0 and is set arbitrary, depending on a solved problem.

An archive built on ϵ-dominance trimming operator can be used as an additional filter of best solutions, reducing outcomes size. This procedure may also supplement the MO-mHGS nodes sprouting process. After a metaepoch, each HGS node selects *delegates* – individuals that will seed new populations whenever sprouting is performed. Choosing the right delegates is specific to a driver, e.g. in NSGAII we take the first layer of non-dominance sorting results, while in SPEA2 we consider the whole result archive. More generically we could assume that the delegates set consists of the whole current node's population. It may lead to worse MO-mHGS performance, but sometimes it is hard to find a better adaptation that could take advantage of driver characteristics. However, by adding an additional layer in form of ϵ-archive we could improve this process and institute more robust independence from a used driver. Figure 1 shows how the final delegates nomination procedure works in another MO-mHGS modification (de-

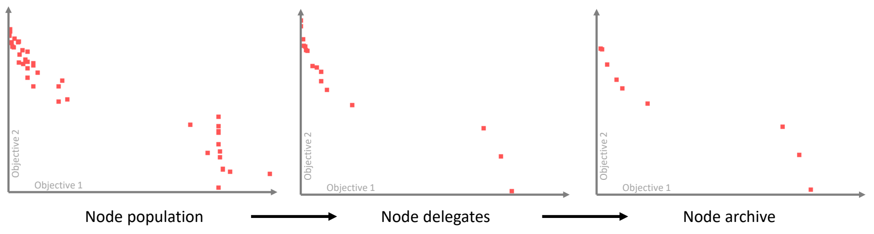


Fig. 1. Presentation of MO-ϵ-EHGS node ϵ-archive update process for 2-objective problem example. Results of each metaepoch are filtered in two stages: as population delegates (specific to a driver) and as ϵ-archive truncation outcomes.

noted as *Multi-Objective ϵ-Elitist Hierarchical Genetic Strategy*, MO-ϵ-EHGS or ϵ-EHGS). At the end of a metaepoch node produces a new population. Then we choose the best solutions using driver-specific mechanics. Finally, the delegates are inserted into ϵ-archive. At later stages, the sprouting procedure randomly selects new seeds from the archive.

It is also worth noting that each node possesses a separate (local) ϵ-archive. The state of each local archive is preserved between metaepochs thus it serves not only as a filtering operator but also as elitist storage. When MO-ϵ-EHGS gathers its final results from all nodes, it receives the content of these local archives (instead of finalized populations).

4 Experiments and results

4.1 Preparing the experiment's environment

The main aim at this stage of experiments was to compare MO-EHGS and MO-ϵ-EHGS with specific MOEA-based driver against basic (single-deme) version of this MOEA. The previous (non-elitist) version of MO-mHGS was also included in the survey. We selected two drivers basing on two different MOEA approaches: NSGAII [5] and SPEA2 [31]. Additionally, we prepared a separate set of simulations for NSGAIII [7] driver, intended to solve many-objective problems.

We conducted several simulations that were set up according to rules inspired by CEC09 competition [29]: each algorithm had a maximum budget of 300000 fitness function evaluations and was run independently 30 times for each test problem to improve statistical accuracy. Simulations were run on benchmark problems from two widely known families: ZDT [30] and CEC09. We divided problems into 3 categories: basic 2-objective (ZDT1-ZDT6), more demanding 2-objective (UF1-UF7) and complex with more objectives. The last category included a 3-objective (UF8-UF10) and one 5-objective (UF11). Problems from the first two groups were tested with NSGAII and SPEA2 drivers and problems from the third group with NSGAIII.

At the end of each run (after reaching the maximum budget), result populations were stored and evaluated with 3 different quality indicators. Inverted Generational Distance (IGD) [3] computes the average size of the gap between the members of optimal Pareto front and the one approximated by an assessed algorithm. Hypervolume (HV) [28] measures the volume of the result space dominated by the found solution. Spacing [3] describes how uniformly distributed the points in solution are.

Metric values from all 30 runs were then statistically analyzed. We calculated minimum, average and maximum value with standard deviation. To ensure statistical significance we also performed Kruskal-Wallis [15] and Mann-Whitney U [19] tests. Basing on the tests results, we were able to create set of indifferent algorithms I_{α_i} for each result of an algorithm α_i. We define a result as a **global winner** if it outperforms other algorithms and $I_{\alpha_i} = \emptyset$. Because we are interested in analyzing impact of MO-mHGS in improving driver's performance we also introduce concept of **local winners**. An algorithm α_i is marked as strong if it outperforms other algorithms and:

$$I_{\alpha_i} = \begin{cases} \emptyset \vee \{\alpha_j \mid \alpha_j \in A_{HGS} \wedge 0 \leq j \leq N \wedge j \neq i\} & \alpha_i \in A_{HGS} \\ \emptyset \vee \{\alpha_j \mid 0 \leq j \leq N \wedge j \neq i\} \wedge A_{HGS} \subsetneq I_{\alpha_i} & \text{otherwise} \end{cases}$$

where A_{HGS} is set of all MO-mHGS algorithm variants and N number of all algorithms included in the simulation. In other words, MO-mHGS results have to be statistically different from single-deme algorithms, but they can be indifferent about other MO-mHGS variants. On the other hand, a single-deme algorithm should win over at least one MO-HGS variant to be a local winner.

For completeness, we also put in our summary number of **weak winners**, that is algorithms that outperformed other solutions regardless of significance test result.

All described simulations were performed with the use of kEMO [1], our new platform written in Kotlin. It is also a wrapper for popular Java library, MOEA Framework [11] containing implementations of state-of-the-art MOEAs and multi-objective problems. Therefore, all algorithm parameters were set with default values provided by MOEA Framework. That includes i. a. ϵ values used in ϵ-archives specific to each considered benchmark.

4.2 Improving MOEA performance

Tables 1, 2 and 3 show comparison of MO-mHGS, MO-EHGS and MO-ϵ-EHGS with reference to NSGAII driver. In all problems but one MO-mHGS variants offer better performance than NSGAII in terms of Hypervolume and IGD. While final Hypervolume outcomes were close to each other in all cases (with notable

[1] https://github.com/Soamid/kEMO

Table 1. Summary of NSGAII with different variants of MO-mHGS after 300000 fitness evaluations (Hypervolume indicator, higher is better).

Hypervolume																
	NSGAII				MO-mHGS+NSGAII				MO-EHGS+NSGAII				MO-ϵ-EHGS+NSGAII			
Problem	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error
zdt1	0.65561	0.65678	0.65765	0.00041	0.66034	0.66086	0.66120	0.00019	**0.662015**	**0.662049**	**0.662073**	**0.000016**	0.66071	0.66112	0.66139	0.00020
zdt2	0.32265	0.32395	0.32486	0.00050	0.32706	0.32756	0.32785	0.00018	**0.328736**	**0.328764**	**0.328797**	**0.000016**	0.32765	0.32789	0.32814	0.00013
zdt3	0.51374	0.51412	0.51455	0.00020	0.515289	0.515455	0.515604	0.000070	**0.5159753**	**0.5160028**	**0.5160215**	**0.0000099**	0.51372	0.51468	0.51507	0.00033
zdt4	0.00	0.08	0.34	**0.10**	0.02	0.33	0.66	0.20	0.04	**0.34**	**0.66**	0.16	**0.04**	0.31	0.66	0.15
zdt6	0.39564	0.39632	0.39693	0.00035	0.39828	0.39964	0.40021	0.00038	**0.40050**	**0.40073**	**0.40105**	**0.00011**	0.39901	0.39957	0.39992	0.00022
UF1	0.470	0.558	0.609	0.033	0.515	0.565	0.612	0.028	0.545	**0.601**	**0.634**	**0.022**	**0.546**	0.599	0.631	0.022
UF2	0.6145	0.6222	0.6301	0.0039	0.6094	0.6251	0.6371	0.0063	0.6232	0.6320	**0.6389**	0.0047	**0.6252**	**0.6342**	0.6386	**0.0035**
UF3	0.353	0.452	0.525	0.051	0.369	0.458	0.524	**0.043**	**0.375**	**0.469**	**0.553**	0.052	0.351	0.454	0.544	0.050
UF4	**0.2518**	**0.2564**	**0.2591**	**0.0018**	0.2397	0.2488	0.2562	0.0042	0.2321	0.2426	0.2536	0.0053	0.2338	0.2445	0.2527	0.0049
UF5	0.055	0.169	0.297	0.069	0.152	0.235	0.303	0.031	**0.179**	**0.243**	0.304	**0.031**	0.133	0.232	**0.305**	0.036
UF6	0.000	0.193	0.324	0.084	**0.193**	**0.273**	**0.356**	**0.048**	0.060	0.231	0.349	0.072	0.088	0.247	0.352	0.074
UF7	0.10	0.34	0.46	0.11	0.192	0.364	0.461	0.099	0.193	0.383	**0.478**	0.095	**0.207**	**0.397**	0.477	**0.091**
Weak wins	1	1	1	2	1	1	1	2	**6**	**8**	**9**	**6**	4	2	1	2
Local wins	1	1	1	2	1	1	1	2	**5**	**7**	**8**	**6**	4	2	1	2
Global wins	1	1	1	2	0	0	0	2	**4**	**4**	**5**	**6**	1	1	0	2

Table 2. Summary of NSGAII with different variants of MO-mHGS after 300000 fitness evaluations (IGD indicator, lower is better).

Inverted Generational Distance																
	NSGAII				MO-mHGS+NSGAII				MO-EHGS+NSGAII				MO-ϵ-EHGS+NSGAII			
Problem	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error
zdt1	0.00693	0.00736	0.00814	0.00028	0.003863	0.004020	0.004190	0.000083	**0.003639**	**0.003683**	**0.003717**	**0.000018**	0.003795	0.003906	0.004091	0.000068
zdt2	0.00676	0.00751	0.00815	0.00033	0.00406	0.00423	0.00458	0.00011	**0.003739**	**0.003788**	**0.003869**	**0.000027**	0.00399	0.00421	0.00451	0.00012
zdt3	0.00483	0.00534	0.00589	0.00025	0.002848	0.002982	0.003098	0.000052	**0.002668**	**0.002713**	**0.002791**	**0.000026**	0.00326	0.00361	0.00450	0.00027
zdt4	0.26	0.73	1.60	0.34	0.00	0.29	0.68	0.21	**0.00**	**0.28**	**0.67**	0.17	0.00	0.30	0.67	**0.16**
zdt6	0.00498	0.00578	0.00891	0.00087	**0.00308**	0.00338	0.00423	0.00028	0.003126	**0.003308**	**0.003519**	**0.000084**	0.00356	0.00389	0.00440	0.00019
UF1	0.045	0.080	0.148	0.025	0.040	0.076	0.121	0.024	**0.020**	**0.044**	**0.081**	0.017	0.022	0.047	0.082	**0.017**
UF2	0.0258	0.0350	0.0545	0.0055	0.0212	0.0342	0.0604	0.0089	**0.0179**	0.0263	0.0461	0.0064	0.0190	**0.0249**	**0.0370**	**0.0043**
UF3	0.101	0.183	0.317	0.058	0.089	**0.162**	**0.231**	**0.044**	**0.083**	0.166	0.302	0.054	0.086	0.200	0.341	0.060
UF4	**0.0484**	**0.0501**	**0.0542**	**0.0013**	0.0499	0.0556	0.0641	0.0034	0.0525	0.0602	0.0703	0.0045	0.0529	0.0584	0.0668	0.0039
UF5	0.169	0.280	0.583	0.096	0.157	0.192	0.250	0.020	**0.146**	**0.186**	**0.234**	**0.020**	0.148	0.200	0.349	0.041
UF6	0.09	0.24	0.53	0.14	**0.059**	**0.139**	**0.317**	**0.061**	0.09	0.19	0.45	0.11	0.07	0.20	0.51	0.13
UF7	0.02	0.19	0.57	0.17	0.02	0.16	0.43	0.15	**0.01**	0.14	0.42	0.14	0.01	**0.12**	**0.40**	**0.14**
Weak wins	1	1	1	1	2	2	2	2	**9**	**7**	**7**	**5**	0	2	2	4
Local wins	1	1	1	1	2	1	1	2	**8**	**7**	**7**	**5**	0	2	2	4
Global wins	1	1	1	1	1	1	1	2	**3**	**3**	**3**	**5**	0	0	0	4

Table 3. Summary of NSGAII with different variants of MO-mHGS after 300000 fitness evaluations (Spacing indicator, lower is better).

Spacing																
	NSGAII				MO-mHGS+NSGAII				MO-EHGS+NSGAII				MO-ϵ-EHGS+NSGAII			
Problem	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error
zdt1	0.0074	0.0109	0.0133	0.0014	0.00229	0.00292	0.00386	0.00031	**0.00067**	**0.00095**	**0.00124**	**0.00017**	0.003	0.059	0.156	0.045
zdt2	0.00825	0.01066	0.01240	0.00088	0.00281	0.00352	0.00478	0.00035	**0.00067**	**0.00096**	**0.00160**	**0.00021**	0.00250	0.00325	0.00410	0.00044
zdt3	0.0090	0.0131	0.0166	0.0015	0.00342	0.00408	0.00485	0.00047	**0.00192**	**0.00249**	**0.00304**	**0.00032**	0.006	0.051	0.219	0.048
zdt4	0.0116	0.0145	**0.0195**	**0.0022**	0.004	0.026	0.080	0.028	**0.001**	**0.010**	0.073	0.017	0.00	0.16	1.25	0.28
zdt6	0.0070	0.0088	0.0110	0.0010	0.0032	0.0042	0.0091	0.0011	**0.00055**	**0.00095**	**0.00145**	**0.00022**	0.002	0.010	0.196	0.035
UF1	0.0017	0.0040	0.0098	0.0022	**0.0006**	**0.0024**	**0.0060**	**0.0015**	0.0018	0.0040	0.0099	0.0021	0.0012	0.0034	0.0084	0.0017
UF2	0.0073	0.0093	0.0116	**0.0010**	0.0037	0.0090	0.0225	0.0053	0.0033	**0.0051**	**0.0079**	0.0013	**0.0030**	0.0075	0.0267	0.0051
UF3	0.000	0.022	0.061	0.018	0.000	0.016	0.084	0.019	0.0001	0.0094	0.0360	0.0092	**0.0001**	**0.0061**	**0.0290**	**0.0084**
UF4	0.0097	0.0115	0.0137	**0.0010**	0.0039	**0.0053**	**0.0093**	0.0012	**0.0037**	0.0053	0.0129	0.0016	0.0039	0.0065	0.0104	0.0014
UF5	**0.000**	**0.030**	**0.117**	**0.031**	0.006	0.048	0.133	0.032	0.001	0.034	0.186	0.041	0.000	0.088	0.364	0.095
UF6	**0.000**	**0.026**	0.170	0.035	0.000	0.033	**0.092**	**0.028**	0.000	0.029	0.147	0.031	0.00	0.10	0.77	0.19
UF7	**0.0002**	0.0048	**0.0116**	0.0037	0.0002	0.0042	0.0141	0.0036	0.0008	0.0042	0.0129	**0.0027**	0.0005	**0.0034**	0.0137	0.0027
Weak wins	3	2	3	4	1	2	3	2	**6**	**6**	**5**	**5**	2	2	1	1
Local wins	1	1	2	4	1	2	2	2	**6**	**6**	**5**	**5**	2	1	1	1
Global wins	0	0	0	4	1	1	1	2	**5**	**6**	**5**	**5**	1	1	1	1

MO-mHGS winners), in IGD indicator we can observe improvement up to 50-60% for simpler ZDT problems and 20-30% for UF familly. Spacing of the outcomes also is the best in MO-mHGS variants, but it's less consistent – in simpler problems spacing gains even 90% improvement, but the last 3 UF problems reach slightly (but insignificantly) better results without MO-HGS.

Concerning elitist mechanics impact on MO-mHGS performance, it is also clear that MO-EHGS figures as the best variant in any considered metric. It wins

Table 4. Comparison of NSGAII and SPEA2 results with different variants of elitist MO-mHGS after 300000 fitness evaluations (Hypervolume indicator, higher is better).

Hypervolume												
	NSGAII		EHGS+NSGAII		ϵ-EHGS+NSGAII		SPEA2		EHGS+SPEA2		ϵ-EHGS+SPEA2	
Problem	Average	Error	Average	Error	Average	Error	Average	Error	Average	Error	Average	Error
zdt1	0.65678	0.00041	**0.662049**	**0.000016**	0.66112	0.00020	0.65849	0.00021	0.661858	0.000058	0.66078	0.00016
zdt2	0.32395	0.00050	**0.328764**	**0.000016**	0.32789	0.00013	0.32548	0.00020	0.328655	0.000052	0.32782	0.00013
zdt3	0.51412	0.00020	**0.5160028**	**0.0000099**	0.51468	0.00033	0.51448	0.00013	0.515841	0.000042	0.51446	0.00012
zdt4	0.08	**0.10**	**0.34**	0.16	0.31	0.15	0.09	0.12	0.21	0.13	0.18	0.14
zdt6	0.39632	0.00035	**0.40073**	0.00011	0.39957	0.00022	0.39771	**0.00011**	0.40051	0.00022	0.39932	0.00024
UF1	0.558	0.033	0.601	0.022	0.599	0.022	0.553	0.021	0.618	0.021	**0.622**	**0.014**
UF2	0.6222	0.0039	0.6320	0.0047	0.6342	0.0035	0.6250	0.0085	**0.6351**	**0.0031**	0.6348	0.0033
UF3	0.452	0.051	0.469	0.052	0.454	0.050	0.433	**0.041**	0.488	0.048	**0.499**	0.043
UF4	0.2564	0.0018	0.2426	0.0053	0.2445	0.0049	**0.2626**	**0.0016**	0.2548	0.0036	0.2546	0.0025
UF5	0.169	0.069	0.243	**0.031**	0.232	0.036	0.155	0.088	**0.264**	0.037	0.261	0.048
UF6	0.193	0.084	0.231	0.072	0.247	0.074	0.232	0.083	0.305	0.054	**0.311**	**0.045**
UF7	0.34	0.11	0.383	0.095	0.397	0.091	0.32	0.11	0.404	0.089	**0.431**	**0.080**
Weak wins	0	1	**5**	**4**	0	0	1	3	2	1	4	3
Local wins	0	1	**5**	**4**	0	0	1	3	2	1	4	3
Global wins	0	1	4	4	0	0	1	3	0	1	0	3

in 6-8 out of 12 test cases with all competitors (**bold values** in the tables), almost always beating NSGAII significantly (in terms of local wins), but in several situations is also statistically different than other considered MO-mHGS variants (denoted with **red bolding** in the tables). MO-ϵ-EHGS is generally worse in this case, generating similar outcomes to the basic MO-mHGS algorithm.

Table 5. Comparison of NSGAII and SPEA2 results with different variants of elitist MO-mHGS after 300000 fitness evaluations (IGD indicator, lower is better).

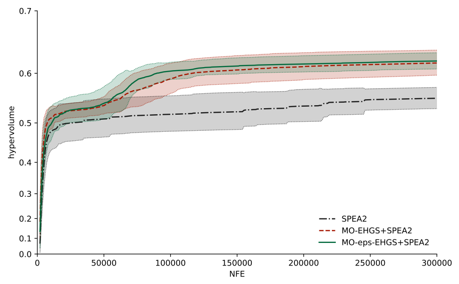
Inverted Generational Distance												
	NSGAII		EHGS+NSGAII		ϵ-EHGS+NSGAII		SPEA2		EHGS+SPEA2		ϵ-EHGS+SPEA2	
Problem	Average	Error	Average	Error	Average	Error	Average	Error	Average	Error	Average	Error
zdt1	0.00736	0.00028	**0.003683**	**0.000018**	0.003906	0.000068	0.00626	0.00013	0.003743	0.000044	0.004033	0.000072
zdt2	0.00751	0.00033	**0.003788**	**0.000027**	0.00421	0.00012	0.006224	0.000094	0.004006	0.000090	0.00430	0.00012
zdt3	0.00534	0.00025	**0.002713**	**0.000026**	0.00361	0.00027	0.00492	0.00013	0.002794	0.000040	0.00363	0.00014
zdt4	0.73	0.34	**0.28**	0.17	0.30	**0.16**	0.68	0.29	0.42	0.17	0.46	0.19
zdt6	0.00578	0.00087	**0.003308**	**0.000084**	0.00389	0.00019	0.00495	0.00026	0.00378	0.00020	0.00439	0.00031
UF1	0.080	0.025	0.044	0.017	0.047	0.017	0.087	0.017	0.033	0.015	**0.031**	**0.010**
UF2	0.0350	0.0055	0.0263	0.0064	0.0249	0.0043	0.036	0.016	**0.0226**	0.0042	0.0236	**0.0041**
UF3	0.183	0.058	0.166	0.054	0.200	0.060	0.202	0.047	**0.143**	0.046	0.144	**0.043**
UF4	0.0501	0.0013	0.0602	0.0045	0.0584	0.0039	**0.0469**	**0.0012**	0.0527	0.0027	0.0536	0.0019
UF5	0.280	0.096	0.186	**0.020**	0.200	0.041	0.290	0.094	**0.168**	0.029	0.172	0.031
UF6	0.24	0.14	0.19	0.11	0.20	0.13	0.22	0.13	**0.101**	**0.039**	0.101	0.066
UF7	0.19	0.17	0.14	0.14	0.12	0.14	0.22	0.16	0.12	0.13	**0.08**	**0.12**
Weak wins	0	0	**5**	**5**	0	1	1	1	4	1	2	4
Local wins	0	0	**5**	**5**	0	1	1	1	4	1	2	4
Global wins	0	0	**5**	**5**	0	1	1	1	0	1	0	4

In order to ensure if driver reinforced by elitist MO-mHGS can be still competitive when compared to another MOEA, we also show a comparison of NSGAII and SPEA2 (with their MO-mHGS versions) in Tables 4, 5 and 6. Again, basic single-deme versions of the algorithms are rarely seen as winners, with MO-EHGS+NSGAII remains the most profitable in ZDT familly and MO-EHGS+SPEA2 in UF problems.

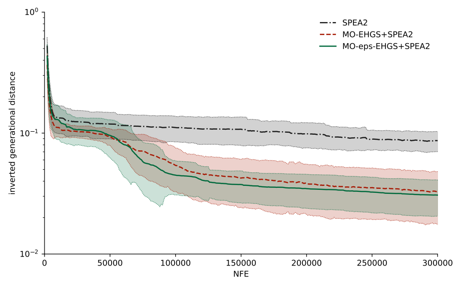
Figure 2 shows detailed simulation run of SPEA2 with elitist MO-mHGS in UF1 problem. While final results of MO-EHGS+SPEA2 and MO-ϵ-EHGS+SPEA2

Table 6. Comparison of NSGAII and SPEA2 results with different variants of elitist MO-mHGS after 300000 fitness evaluations (Spacing indicator, lower is better).

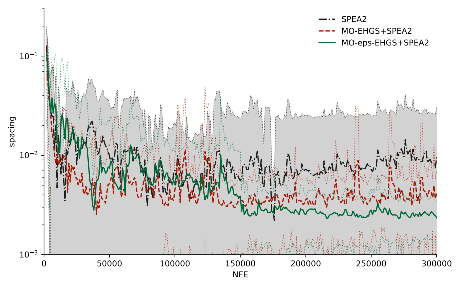
	Spacing											
	NSGAII		EHGS+NSGAII		ϵ-EHGS+NSGAII		SPEA2		EHGS+SPEA2		ϵ-EHGS+SPEA2	
Problem	Average	Error	Average	Error	Average	Error	Average	Error	Average	Error	Average	Error
zdt1	0.0109	0.0014	**0.00095**	**0.00017**	0.059	0.045	0.00519	0.00067	0.00198	0.00069	0.036	0.033
zdt2	0.01066	0.00088	**0.00096**	**0.00021**	0.00325	0.00044	0.00536	0.00062	0.00287	0.00072	0.00365	0.00058
zdt3	0.0131	0.0015	**0.00249**	**0.00032**	0.051	0.048	0.00662	0.00087	0.00355	0.00052	0.0142	0.0074
zdt4	0.0145	0.0022	0.010	0.017	0.16	0.28	**0.0068**	**0.0010**	0.025	0.032	0.101	0.089
zdt6	0.0088	0.0010	**0.00095**	**0.00022**	0.010	0.035	0.00377	0.00044	0.00170	0.00053	0.0055	0.0092
UF1	0.0040	0.0022	0.0040	0.0021	0.0034	0.0017	0.010	0.021	0.0040	0.0026	**0.0025**	**0.0012**
UF2	0.0093	**0.0010**	**0.0051**	0.0013	0.0075	0.0051	0.0062	0.0025	0.0057	0.0022	0.0069	0.0020
UF3	0.022	0.018	0.0094	0.0092	**0.0061**	**0.0084**	0.019	0.018	0.021	0.015	0.023	0.014
UF4	0.0115	0.0010	0.0053	0.0016	0.0065	0.0014	0.0067	0.0020	**0.00458**	**0.00040**	0.0053	0.0012
UF5	**0.030**	**0.031**	0.034	0.041	0.088	0.095	0.055	0.057	0.082	0.077	0.12	0.11
UF6	**0.026**	0.035	0.029	**0.031**	0.10	0.19	0.10	0.21	0.047	0.050	0.06	0.11
UF7	0.0048	0.0037	0.0042	0.0027	0.0034	0.0027	0.009	0.022	**0.0028**	**0.0014**	0.0050	0.0032
Weak wins	2	2	**5**	**5**	1	1	1	1	2	2	1	1
Local wins	1	2	**4**	**5**	1	1	1	1	1	2	0	1
Global wins	0	2	4	5	1	1	0	1	1	2	0	1



(a) Hypervolume



(b) IGD



(c) Spacing

Fig. 2. Quality indicators values over number of fitness evaluations in UF1 (2-objective) problem (SPEA2 algorithm).

does not differ significantly, they both quickly reach Hypervolume and IGD values far from single-deme algorithm. Better convergence of MO-ϵ-EHGS+SPEA2 at earlier stages of simulation is also worth mentioning. Spacing metrics tend to have much higher variance and thus their outcomes are not always interpretable,

Table 7. Summary of NSGAIII results with different variants of MO-mHGS after 300000 fitness evaluations (Hypervolume indicator, higher is better).

Hypervolume																
	NSGAIII				MO-mHGS+NSGAIII				MO-EHGS+NSGAIII				MO-ϵ-EHGS+NSGAIII			
Problem	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error
UF8	0.000	0.193	0.257	0.041	0.124	0.172	0.221	**0.023**	0.000	0.218	0.266	0.048	**0.207**	**0.243**	**0.307**	0.034
UF9	0.408	0.521	**0.618**	0.092	0.389	0.450	0.516	**0.035**	**0.461**	0.537	0.590	0.044	0.440	**0.562**	0.616	0.051
UF10	0.000	0.056	0.162	0.044	**0.013**	0.058	0.130	**0.029**	0.001	0.061	0.152	0.034	0.002	**0.072**	**0.185**	0.048
UF11	0.016	0.051	0.103	0.023	0.0000	0.0014	0.0089	**0.0020**	0.049	0.084	0.124	0.017	**0.167**	**0.195**	**0.229**	0.014
Weak wins	0	0	1	0	1	0	0	**4**	1	0	0	0	**2**	**4**	**3**	0
Local wins	0	0	1	0	0	0	0	**4**	0	0	0	0	**2**	**2**	**2**	0
Global wins	0	0	0	0	0	0	0	4	0	0	0	0	2	2	2	0

Table 8. Summary of NSGAIII results with different variants of MO-mHGS after 300000 fitness evaluations (IGD indicator, lower is better).

Inverted Generational Distance																
	NSGAIII				MO-mHGS+NSGAIII				MO-EHGS+NSGAIII				MO-ϵ-EHGS+NSGAIII			
Problem	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error
UF8	0.13	0.19	0.75	0.11	0.142	0.170	0.209	**0.018**	0.10	0.17	0.75	0.11	**0.094**	**0.150**	**0.207**	0.039
UF9	0.079	0.169	0.310	0.095	0.126	0.171	0.217	**0.024**	0.089	0.130	0.207	0.038	**0.070**	**0.109**	**0.186**	0.038
UF10	0.23	0.38	0.66	0.11	0.206	**0.261**	**0.325**	**0.032**	0.211	0.298	0.441	0.061	**0.19**	0.38	0.70	0.13
UF11	0.411	0.536	0.678	0.059	0.629	0.747	0.846	0.056	0.407	0.455	0.513	0.024	**0.313**	**0.342**	**0.385**	**0.016**
Weak wins	0	0	0	0	0	1	1	**3**	0	0	0	0	**4**	**3**	**3**	1
Local wins	0	0	0	0	0	1	1	**3**	0	0	0	0	**3**	**3**	**3**	1
Global wins	0	0	0	0	0	1	1	3	0	0	0	0	2	2	2	1

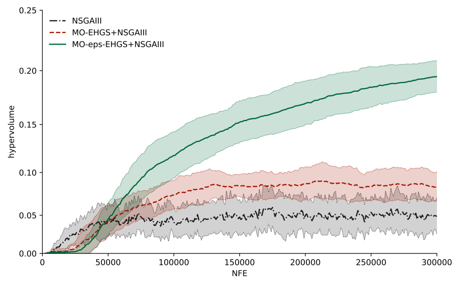
Table 9. Summary of NSGAIII results with different variants of MO-mHGS after 300000 fitness evaluations (Spacing indicator, lower is better).

Spacing																
	NSGAIII				MO-mHGS+NSGAIII				MO-EHGS+NSGAIII				MO-ϵ-EHGS+NSGAIII			
Problem	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error	Min	Average	Max	Error
UF8	**0.000**	0.098	0.181	0.037	0.038	0.077	0.304	0.047	0.010	**0.056**	**0.072**	**0.012**	0.040	0.057	0.102	0.013
UF9	**0.024**	0.057	0.110	0.021	0.042	0.070	0.152	0.027	0.0391	0.0546	0.0785	**0.0094**	0.0334	**0.0517**	**0.0732**	0.0099
UF10	0.033	0.125	0.410	0.075	0.091	0.145	0.268	0.048	0.038	0.112	**0.205**	**0.039**	**0.000**	**0.093**	0.216	0.057
UF11	0.157	0.228	0.313	0.038	0.216	0.255	0.305	0.023	0.1455	0.1601	**0.1815**	**0.0078**	**0.126**	**0.155**	0.184	0.013
Weak wins	**2**	0	0	0	0	0	0	0	0	1	**3**	**4**	**2**	**3**	1	0
Local wins	**1**	0	0	0	0	0	0	0	0	**1**	**2**	**4**	**1**	**1**	0	0
Global wins	1	0	0	0	0	0	0	0	0	0	1	4	1	1	0	0

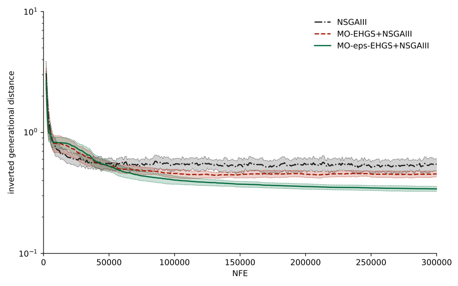
but there's a visible trend of MO-ϵ-EHGS+SPEA2 minimizing both indicator value and error rate at later stages.

4.3 Evaluation of many-objective problem

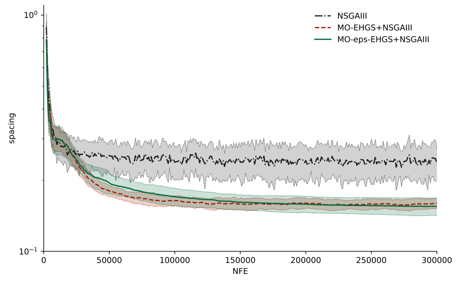
The last group of benchmarks were problems with higher objective dimensionality. In Tables 7, 8 and 9 we present outcomes of NSGAIII driver, whose basic form was designed to deal with many-objective problems. This time basic, non-elitist version of MO-mHGS was not able to improve MOEA – its performance was meaningfully worse (about 10-30% or even more in the hardest, 5-objective problem). At the same time, elitist variants provided the best possible solutions regardless of metrics. Furthermore, even though MO-EHGS outperformed bare NSGAIII, it is MO-ϵ-EHGS that should be considered as a winner here. Local archives used in sprouting procedure improved delegates selection and in the end resulted in up to 2-3 times better improvement than MO-EHGS with only one,



(a) Hypervolume



(b) IGD



(c) Spacing

Fig. 3. Quality indicators values over number of fitness evaluations in UF11 (5-objective) problem (NSGAIII algorithm).

global archive. This behaviour is also visible in Figure 3 of NSGAIII run solving the hardest, 5-objective problem. MO-ϵ-EHGS+NSGAIII reaches new regions in terms of Hypervolume and IGD while remaining stable and well distributed.

The reason why MO-ϵ-EHGS was considerably less impressive in previous experiments may be connected with our previous hypothesis about delegates nomination procedure impact. NSGAII and SPEA2 both incorporate simple elitism mechanics internally thus reducing their delegates set with $\epsilon - archive$ might not have been necessary. NSGAIII does not use an archive by itself and delegates nomination is based mainly on results of non-dominated sorting.

5 Conclusions

In this paper, we have shown the evolution of multi-objective hierarchical strategy, by introducing a notion of elitism by adding local and global archives into the search tree. We did our best to compare the proposed algorithms using a carefully selected set of benchmarks as well as the competitors.

Both proposed elitism upgrades of the MO-mHGS model were able to improve the performance of the hierarchical meta-model. While MO-EHGS stands out as a generic multi-purpose tool and may be paired with any MOEA in most of the considered situations, applications of MO-ϵ-EHGS are more specific to a driver. The good performance of MO-ϵ-EHGS in problems with more objectives is also notable and should be further investigated in future research. Correlation between MO-mHGS delegates nomination procedure and sprouts quality and other methods of controlling MO-mHGS node progress are also very promising topics that will be considered as next steps in our research project.

In our further research, we will also focus on the matters of concurrent implementation of the proposed algorithms, especially considering the scalability. Such hierarchical systems, constantly dynamically changing, should be implemented utilizing matters of concurrent processes, going towards parallelization, especially using HPC environments, which can help in speeding up the running of actually very complex metaheuristic algorithms.

References

1. Bellomo, D., Naso, D., Turchiano, B.: Improving genetic algorithms: an approach based on multi-elitism and lamarckian mutation. In: IEEE International Conference on Systems, Man and Cybernetics. vol. 4, pp. 6–pp. IEEE (2002)
2. Ciepiela, E., Kocot, J., Siwik, L., Dreżewski, R.: Hierarchical approach to evolutionary multi-objective optimization. In: Computational Science–ICCS 2008, pp. 740–749. Springer (2008)
3. Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A., et al.: Evolutionary algorithms for solving multi-objective problems, vol. 5. Springer (2007)
4. Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J.: Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. pp. 283–290 (2001)
5. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Lecture notes in computer science **1917**, 849–858 (2000)
6. Deb, K., Goel, T.: Controlled elitist non-dominated sorting genetic algorithms for better convergence. In: International conference on evolutionary multi-criterion optimization. pp. 67–81. Springer (2001)
7. 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 (2013)
8. Dulebenets, M.A.: Archived elitism in evolutionary computation: towards improving solution quality and population diversity. International Journal of Bio-Inspired Computation **15**(3), 135–146 (2020)
9. González-Almagro, G., Rosales-Pérez, A., Luengo, J., Cano, J.R., García, S.: Improving constrained clustering via decomposition-based multiobjective optimization with memetic elitism. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. pp. 333–341 (2020)
10. Guariso, G., Sangiorgio, M.: Improving the performance of multiobjective genetic algorithms: An elitism-based approach. Information **11**(12), 587 (2020)
11. Hadka, D.: Beginner's guide to the moea framework (2016)
12. Idzik, M., Byrski, A., Turek, W., Kisiel-Dorohinicki, M.: Asynchronous actor-based approach to multiobjective hierarchical strategy. In: International Conference on Computational Science. pp. 172–185. Springer (2020)

13. Ishibuchi, H., Pang, L.M., Shang, K.: A new framework of evolutionary multi-objective algorithms with an unbounded external archive (2020)
14. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary computation **8**(2), 149–172 (2000)
15. Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis. Journal of the American statistical Association **47**(260), 583–621 (1952)
16. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evolutionary computation **10**(3), 263–282 (2002)
17. Lazarz, R., Idzik, M., Gadek, K., Gajda-Zagorska, E.: Hierarchic genetic strategy with maturing as a generic tool for multiobjective optimization. Journal of Computational Science **17**, 249–260 (2016)
18. Li, Q., Liu, L., Yuan, X.: Multiobjective parallel chaos optimization algorithm with crossover and merging operation. Mathematical Problems in Engineering **2020** (2020)
19. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics pp. 50–60 (1947)
20. Sano, R., Aguirre, H., Tanaka, K.: A closer look to elitism in ε-dominance many-objective optimization. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 2722–2729. IEEE (2017)
21. Sato, M., Aguirre, H.E., Tanaka, K.: Effects of δ-similar elimination and controlled elitism in the nsga-ii multiobjective evolutionary algorithm. In: 2006 IEEE International Conference on Evolutionary Computation. pp. 1164–1171. IEEE (2006)
22. Schaefer, R., Kolodziej, J.: Genetic search reinforced by the population hierarchy. In: Jong, K.A.D., Poli, R., Rowe, J.E. (eds.) Proceedings of the Seventh Workshop on Foundations of Genetic Algorithms, Torremolinos, Spain, September 2-4, 2002. pp. 383–400. Morgan Kaufmann (2002)
23. Schaefer, R., Kolodziej, J.: Genetic search reinforced by the population hierarchy. In: Foundations of Genetic Algorithms. vol. 7, pp. 383–401 (2002)
24. Sierra, M., Coello Coello, C.: Improving pso-based multi-objective optimization using crowding, mutation and ϵ-dominance. In: Coello Coello, C., Hernández Aguirre, A., Zitzler, E. (eds.) Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, vol. 3410, pp. 505–519. Springer Berlin Heidelberg (2005)
25. Simon, D., Ergezer, M., Du, D.: Population distributions in biogeography-based optimization algorithms with elitism. In: 2009 IEEE International Conference on Systems, Man and Cybernetics. pp. 991–996. IEEE (2009)
26. Sun, Y., Gao, Y.: A multi-objective particle swarm optimization algorithm based on gaussian mutation and an improved learning strategy. Mathematics **7**(2), 148 (2019)
27. Tanabe, R., Ishibuchi, H.: Non-elitist evolutionary multi-objective optimizers revisited. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 612–619 (2019)
28. While, L., Bradstreet, L., Barone, L.: A fast way of calculating exact hypervolumes. IEEE Transactions on Evolutionary Computation **16**(1), 86–95 (2011)
29. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition (2008)
30. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation **8**(2), 173–195 (2000)
31. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. TIK-report **103** (2001)