Transfer Strategies from Single- to Multi-objective Grouping Mechanisms

Frederick Sander
Institute for Intelligent Cooperative Systems, Otto-von-Guericke University
Magdeburg, Germany
frederick.sander@posteo.de

Heiner Zille
Institute for Intelligent Cooperative Systems, Otto-von-Guericke University
Magdeburg, Germany
heiner.zille@ovgu.de

Sanaz Mostaghim
Institute for Intelligent Cooperative Systems, Otto-von-Guericke University
Magdeburg, Germany
sanaz.mostaghim@ovgu.de

ABSTRACT

In large-scale optimisation, most algorithms require a separation of the variables into multiple smaller groups and aim to optimise these variable groups independently. In single-objective optimisation, a variety of methods aim to identify best variable groups, most recently the Differential Grouping 2. However, it is not trivial to apply these methods to multiple objectives, as the variable interactions might differ between objective functions. In this work, we introduce four different transfer strategies that allow to use any single-objective grouping mechanisms directly on multi-objective problems. We apply these strategies to a popular single-objective grouping method (Differential Grouping 2) and compare the performance of the obtained groups inside three recent large-scale multi-objective algorithms (MOEA/DVA, LMEA, WOF). The results show that the performance of the original MOEA/DVA and LMEA can in some cases be improved by our proposed grouping variants or even random groups. At the same time the computational budget is dramatically reduced. In the WOF algorithm, a significant improvement in performance compared to random groups or the standard version of the algorithm can on average not be observed.

CCS CONCEPTS

• **Mathematics of computing → Evolutionary algorithms**; **Bio-inspired optimization**; *Optimization with randomized search heuristics*;

KEYWORDS

Multi-objective optimization, Large-scale Optimization, Metaheuristics, Many-variable optimization, Evolutionary Algorithms, Grouping Mechanisms, Variable Interaction, Random Grouping

ACM Reference Format:
Frederick Sander, Heiner Zille, and Sanaz Mostaghim. 2018. Transfer Strategies from Single- to Multi-objective Grouping Mechanisms . In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205491

1 INTRODUCTION

In the area of multi-objective optimisation, a growing interest is drawn to the field of large-scale problems. Large-scale Optimisation (LSO) refers to the optimisation of problems with a large number of decision variables, usually between 200 and 5000. Popular concepts from single-objective optimisation like Cooperative Coevolution [12] require a separation of the decision parameters into subcomponents, called *variable groups*. Most algorithms developed for single- and multi-objective LSO utilise such a form of grouping mechanism.

In single-objective optimisation, a variety of methods exist to determine optimal groups of variables, some of them based on the interaction between the decision parameters. This often comes with a growing computational budget. When dealing with multiple objective functions, it is desirable to extend the developed methods to tackle multi-objective large-scale problems. A small number of papers has paid attention to this area [1, 7, 17, 18, 20], but no study exists so far that deals with different strategies to apply existing single-objective grouping mechanisms to multi-objective problems. Since variable interactions and their influence on the fitness value depends on each single objective function, the optimal groups for a multi-objective problem might be conflicting. In addition, when multiple-objectives are present, groups cannot only be based on variable interactions, but also on the variables' contributions to convergence or diversity of the non-dominated set.

In this article we examine the use of variable grouping mechanisms for multi-objective optimisation. By proposing a set of different transfer strategies, we enable existing grouping mechanisms from the single-objective area to be applied for more than one objective function simultaneously. The contributions of this paper are as follows:

- We introduce four transfer strategies that allow for arbitrary single-objective grouping mechanism to be applied to multiple objectives simultaneously.
- We perform a comparison study by applying these strategies to a recent single-objective grouping mechanism called Differential Grouping 2 [11] in three popular multi-objective large-scale algorithms (MOEA-DVA [7], LMEA [17] and WOF [20]).

The remainder of this article is structured as follows. In Section 2 we briefly introduce the concept of multi-objective and large-scale optimisation and the three algorithms used in this work. The next section gives a short introduction to existing single- and multi-objective grouping methods 3. The different transfer strategies are proposed in Section 4. In Section 5 the experimental evaluation is reported before concluding this article in Section 6.

2 MULTI-OBJECTIVE AND LARGE-SCALE OPTIMISATION

Problems in nature and science often contain multiple conflicting objectives. These problems are called multi-objective problems (MOPs) and can mathematically be formulated as:

$$\begin{aligned} Z: \quad & min \quad \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), ..., f_m(\vec{x}))^T \\ & s.t. \quad \vec{x} \in \Omega \subseteq \mathbb{R}^n \end{aligned} \tag{1}$$

where $m \geq 2$. This kind of MOP maps the decision Space $\Omega = \{\vec{x} \in \mathbb{R}^n \,|\, \vec{g}(\vec{x}) \leq 0\}$ of dimension n to the objective space of dimension m. For such problems, a single optimal solution can often not be determined. Modern problem solving methods often concentrate on finding an approximation of a Pareto-optimal solution set.

Large-scale optimisation (LSO) usually deals with optimising MOPs that contain a large number (usually $\geq$ 200) of variables. LSO has been studied extensively for the single-objective case. One of the most popular concepts is called *Cooperative Coevolution* (CC) and was first introduced by Potter and De Jong in 1994 [12]. CC aims to optimise several independent populations, each of which only contains a subset of the n decision variables. Such a subset is called a *variable group* in the remainder of this work. New solution candidates have to be formed by combining the variable values from different populations. However, genetic operators are only applied within each groups' population. The concept of CC has since been used in a variety of large-scale single-objective algorithms [2, 5, 6, 15, 16]. An overview of existing large-scale global optimisers for single-objective problems can be found in [8]. In contrast, the area of multi-objective problems with large numbers of variables has only grown popularity within the last few years [1, 7, 17, 18, 20]. Most recently, three algorithms (MOEA/DVA, LMEA and WOF) to solve multi-objective large-scale problems have been proposed.

MOEA/DVA (Multi-objective Evolutionary Algorithm based on Decision Variable Analysis) was proposed in 2016 [7]. As previous methods based on CC, it divides the variables into multiple groups and then optimises these groups independently. In comparison to previous single-objective optimisation methods and grouping mechanisms, the decision variable analysis in MOEA/DVA was designed to identify not only interaction between variables, but also the contribution of a variable to convergence towards optimal solutions, diversity of the solution set or both (for details see [7]). Some details on the used grouping mechanism to detect variable interactions are explained below in Section 3. The actual optimisation phase of MOEA/DVA starts after these grouping steps are completed. Each group of convergence-related variables is optimised independently. The diversity-related variables are fixed in this step. A utility value is computed at the end of each second generation to determine whether or not to continue optimising the independent groups. If the increase in solution quality falls below a certain threshold, a so-called *uniformity optimisation* is carried out, which optimises the original problem, including the diversity-related variables, as a whole without using groups. A disadvantage of the MOEA/DVA is that it requires a large computational budget for the initial grouping phase. The complexity varies depending on the problem, but rises roughly quadratically with increasing numbers of function evaluations. In [20], MOEA/DVA needed more than 8,000,000 function evaluations for dividing a 1000-variable problem.

Similar to MOEA/DVA, LMEA (Large-scale Many-objective Evolutionary Algorithm) was designed specifically to deal with problems with many-objectives and large numbers of variables [17]. The total number of decision variables is first divided into convergence- and diversity-related variables using a clustering approach. After that, an interaction analysis is carried out on the convergence-related variables to further divide them into smaller groups, before the actual optimisation takes place. A drawback of LMEA is, similar to MOEA/DVA, that it requires a very large amount of function evaluations to obtain the variable groups. The optimization approach used in MOEA/DVA and LMEA differs from CC. In CC a solution is evaluated by combining the values of the currently optimised group with the best values for the variables in all other groups. MOEA/DVA and LMEA do not use completely independent populations. They utilise one population for the whole algorithm and keep the values of the variables in the other groups fixed when changing a solution by genetic operators.

The Weighted Optimization Framework (WOF) was developed in [19] and [20]. It reduces the dimensionality of the problem by altering a large part of the decision variables at the same time and by the same amount. This is done through so-called transformation functions, which assign one *weight*-value to each group of decision variables. These weight-values (the same amount as the number of variable groups) are then subject to change through a separate optimisation step. The original solutions are altered indirectly through applying the weights in the transformation function to the original variables. For the optimisation inside the WOF algorithm, SMPSO (Speed-constrained Multi-objective Particle Swarm Optimisation) [9] has been shown efficient [20]. As MOEA/DVA and LMEA, WOF needs to divide the variables into groups. In contrast to these two however, WOF does not propose a unique grouping mechanism, but instead uses random or linear groups or just assigns groups based on absolute variable values [19–21]. This process, as opposed to MOEA/DVA and LMEA, does not require any function evaluations.

MOEA/DVA, LMEA and WOF have been tested on a variety of benchmark functions [7, 17, 19–21]. In 2017 a comparison study in [21] evaluated all three of them on the LSMOP (Large-scale Many-objective Problems) benchmarks [3] and paid special attention to the convergence speed, since the former two need large computational budgets for obtaining groups. While MOEA/DVA and LMEA had advantages in smaller problem instances with 200 variables, WOF performed best in terms of convergence speed and final solution quality in the 1000-dimensional experiments.

Another approach to large-scale multi-objective optimisation was shown in 2016 in [18]. Variable groups were directly used within a polynomial mutation operator, without any further change to the surrounding algorithm. The experiments with 1000 variables applied different grouping methods, most of which did not need any additional function evaluations. Even with random groups, the performance of the optimisation was improved significantly by applying the changed mutation operator.

3 EXISTING GROUPING MECHANISMS

In this section we describe briefly some basics of variable interaction and give a short overview of existing grouping mechanisms for single- and multi-objective optimisation.

3.1 Variable Interaction

For obtaining suitable groups, many methods rely on the interaction between decision variables. The interaction between variables according to [7] and [17] is described as follows. For a given objective function $f(\vec{x})$, two decision variables x_i and x_j are assumed to interact with each other if there exist values a_1, a_2, b_1, b_2 so that both of the following conditions are fulfilled:

$$f(\vec{x})|_{x_i=a_1,x_j=b_1} < f(\vec{x})|_{x_i=a_2,x_j=b_1} \tag{2a}$$

$$f(\vec{x})|_{x_i=a_1,x_j=b_2} > f(\vec{x})|_{x_i=a_2,x_j=b_2} \tag{2b}$$

where

$$f(\vec{x})|_{x_i=a,x_j=b} = f(x_1, ..., x_{i-1}, a, x_{i+1}, ..., x_{j-1}, b, x_{j+1}, ..., x_n)$$

This formalizes the following concept: For non-interacting variables, the order between two values $f(\vec{x})|_{x_i=a_1}$ and $f(\vec{x})|_{x_i=a_2}$ does not depend on the value of the variable x_j. If $f(\vec{x})|_{x_i=a_1}$ is smaller than $f(\vec{x})|_{x_i=a_2}$ for a certain value of $x_j = b_1$ (Equation 2a), but larger for another value of $x_j = b_2$ (Equation 2b), the two variables are considered interacting.

3.2 Single-Objective Grouping Methods

In single-objective optimisation, a variety of grouping strategies has been applied so far. Some of them, and especially the most recent approaches will be introduced in this section. An overview over existing grouping methods is given in [8].

3.2.1 Linear and Random Grouping. Random grouping is one of the simplest methods and has often been used in the literature [15], [8]. Linear grouping was used in [19] and has a similar approach. In both methods the number of groups must be specified beforehand. Linear grouping just divides the total number of n variables into γ equal sized groups using the index of the variable as ordering. Random grouping also creates γ equal sized groups but distributes the variables into the groups randomly. Since groups can be obtained without any computational costs (in terms of function evaluations), these are fast and inexpensive methods. This also has the advantage that the grouping can be repeated multiple times during the runtime of an algorithm to increase the possibility that interacting variables end up together in one group. On the downside however, the effect can be quite limited when the problem at hand contains a lot of interacting variables.

3.2.2 Differential Grouping. The idea of Differential Grouping (DG) [10] is to use Equation 2 for the detection of interacting variables. For each combination of two variables x_i and x_j, DG aims to answer the following question: "When changing the value of variable x_i, does the amount of change in the fitness f_k remains the same, no matter which values the other variable x_j takes?". This is checked by comparing absolute fitness differences in response to changes in x_i and x_j. DG needs a predefined threshold value ϵ to determine how much variation in fitness is necessary to regard two variables as interacting. Although it has been shown to work well in the single-objective experiments in [10], DG needs a large computational budget for finding the interactions, which rises quadratically with the number of decision variables.

3.2.3 Differential Grouping 2. More recently in 2017 an extended version of DG was introduced [11]. The Differential Grouping 2 (DG2) improves the quality of found groups, i.e. it is able to find more interactions, and at the same time reduces the required computational budged. The required amount of function evaluations is reduced to $n(n + 1)/2$. For a 1000-variable problem it only requires around 500,000 evaluations. Although this is a smaller amount than DG, it might still be a very large or intractable amount for real applications. The reduced number of function evaluations was achieved through storing and reusing the results of function evaluations within the grouping process.

3.3 Multi-Objective Grouping Methods

In the area of multi-objective large-scale optimisation, the beforementioned algorithms MOEA/DVA and LMEA contain grouping methods which are designed to take the interactions of all objective functions into account simultaneously. Each of them utilize two different grouping types. As mentioned, they first divide the variables based on their contribution to convergence towards optimal solutions and influence on diversity of the solution set. After that groups based on the interaction between variables are determined. As the focus of this work is to make single-objective methods applicable to multi-objective problems, in the following we focus on the latter of these two grouping steps.

3.3.1 MOEA/DVA Interaction Grouping. The grouping mechanism of MOEA/DVA is based on the definition mentioned in Equation 2. After creating an initial population, for each pair of variables x_i and x_j $(i, j \in 1, .., n)$, an individual is randomly selected from the population. Three other solutions are now created by random changes of the values of these variables. With these four solutions the interaction is checked according to Equation 2 separately for each of the m objective functions. To increase the chances of finding interactions, this is repeated multiple times (according to a predefined parameter) for each pair. Two variables are regarded as interacting if they are both convergence-related and an interaction exists in *any* of the m objectives. As an additional measure, the created solutions which contain altered convergence-related variables are merged with the current population, to obtain a better initial population once the optimisation phase begins.

3.3.2 LMEA Interaction Grouping. The interaction analysis in LMEA works in a similar way than the one of MOEA/DVA in the way of how an interaction between two variables is detected. As above, an interaction is assumed if it exists in any of the objective functions. The difference lies in the assignment of variables into groups. For each variable x_i, the algorithm iterates over all existing groups and checks the interactions with all variables x_j in this group. x_i is added to the group, if it interacts with at least one variable x_j of this group. In addition, the analysis of LMEA performs the interaction analysis only on the convergence-related variables, which might lead to lower computational costs.

4 TRANSFER STRATEGIES

In this section we introduce the transfer strategies that enable single-objective methods to be applied to multi-objective problems. In all of the above methods to detect variable interaction, this is done based on the fitness values of the objective functions. As a result, these methods can only make assumptions about the interaction on one specific function. If multiple objective functions exist, these results can lead to conflicting results, as two variables can be interacting in one objective, but not in the other.

One way to deal with this problem might be to only take into account the groups obtained from one single objective. This was done for example in [20]. This strategy however is not very promising, since it neglects much valuable information about the interaction in the other functions. In contrast, the methods used by MOEA/DVA and LMEA perform interaction checks on each function, and group variables together if an interaction exists in *any* of the objectives.

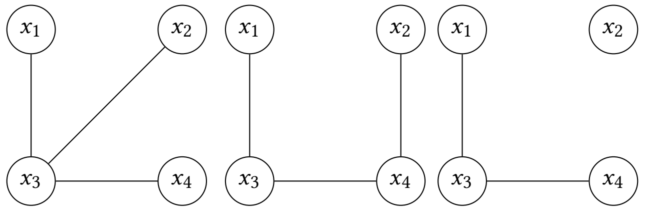
Except these, other ways of combining the groups of multiple objective functions are possible. MOEA/DVA and LMEA each proposed a method of combining the interaction information of different objectives. However, these have usually high computational costs, and for finding interactions, efficient methods like DG2 have been developed in the single-objective area. The goal of this work is therefore to examine different *Transfer Strategies* (TS) to make use of multiple objectives and at the same time apply arbitrary single-objective grouping methods.

These TS can be divided into two consecutive steps. The first one is the strategy for combining the interaction information of different objective functions, called the *Transfer Strategy for Objectives* (TSO). The second part is the *Transfer Strategy for Variables* (TSV), which determines the composition of groups from the combined information of the TSO. In the following we describe possible variants for each of these steps and then define the overall TS as combinations of these variants.

4.1 Transfer Strategies for Objectives

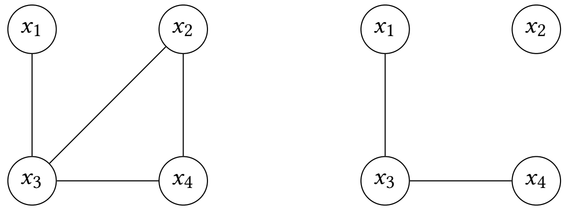
As described above, single-objective grouping mechanisms can only make assumptions about the interaction in one objective function. If applied to multiple objectives, the results can differ. Variables which interact in one function might not do so in another one. As an example we show the interaction graphs of a hypothetical 3-objective problem in Figure 1. The four decison variables x_1 to x_4 are shown as vertices in a graph, where edges indicate that an interaction between the variables exists in the respective function. In this example it can be seen that an interaction between x_2 and x_4 exists in objective function f_2 (Figure 1b) but not in f_3 (Figure 1c). For combining this conflicting information, we propose two different strategies to determine when two variables are regarded as interacting for the overall multi-objective problem. The first one, called TSO_{single}, regards two variables x_i and x_j as interacting, if an interaction exists in *any* of the objective functions. That means that a *single* interaction in one function is sufficient for this strategy to assume interaction for the whole MOP. In contrast, the second approach, called TSO_{all}, only assumes interaction if this link is present in *all* objective functions. With these strategies, the interaction graphs of the problem can be combined into one common graph that contains the interaction information according to the respective strategy. For the example seen in Figure 1, the resulting combined graphs are shown in Figure 2.

Using TSO_{single} leads to the graph depicted in Figure 2a. An interaction in a single objective function is sufficient for the edge to



(a) $f1$ (b) $f2$ (c) $f3$

Figure 1: Interaction graphs of three objective functions. Edges show interaction between two decision variables



(a) TSO_{single} (b) TSO_{all}

Figure 2: Combined interaction graphs of the three functions from Figure 1 for two variants of TSO

be added to the combined graph. The interactions $x_1 - x_3, x_2 - x_3$ and $x_3 - x_4$ are present in the first objective function f_1. $x_2 - x_4$ is present in f_2. The two edges from f_3 are already in the combined graph. TSO_{all} leads to the graph depicted in Figure 2b. An edge only exists in the combined graph if it is present in all three objective functions. In the example only the connections $x_1 - x_3$ and $x_3 - x_4$ are present in all three functions.

4.2 Transfer Strategies for Variables

The second step of a complete TS is the Transfer Strategy for Variables. A TSV defines which variables form a common group to be used in the optimisation. The information obtained from the TSO is used as input, i.e. the TSV works on the combined interaction graph. Similar to the previous step, we call the different approaches TSV_{single} and TSV_{all}. The strategy TSV_{single} is the recently used standard method in most of the single and multi-objective methods [2, 7, 8, 17]. A variable x_i is added to a group if the combined interaction graph contains an edge between x_i and any variable x_j in the group. This means for x_i to be included in a group, a *single* connection to one of the groups' members is sufficient. In other words each group of variables consists of a maximally connected subgraph of the combined interaction graph. If we look back to Figure 2a, all four variables would end up in one group, since they are all connected by at least one edge.

In contrast, TSV_{all} only considers maximal complete subgraphs (MCS). MCS are fully connected subgraphs to which no other vertex can be added so that the new subgraph is still complete. This results in groups with pairwise interactions between *all* of its members. This approach has one drawback, as considering only MCS leads to a problem with shared variables, as was also described in [11].

Shared variables are variables that are members of multiple different MCS and the decision which of these MCS is regarded as a group is not well-defined. In Figure 2a, the variable x_3 is a shared variable, since it appears in the MCS containing x_2, x_3 and x_4, as well as a second MCS with x_1 and x_3. Thus, the resulting variable groups can be $\{x_1, x_3\}\{x_2, x_4\}$ or $\{x_1\}\{x_2, x_3, x_4\}$. In the implementation of the TSV in this work, the variables are distributed to groups in natural order, and therefore each variable is assigned to the MCS which was first found by the algorithm. This solution is deterministic, but it only finds one set of groups when there are maybe more. As a result, the situation in Figure 2a results in the first of the two alternatives: $\{x_1, x_3\}\{x_2, x_4\}$. Other implementations are also possible but are not studied in this work.

4.3 Combinations of TSO and TSV

As mentioned above, a TS requires two components, a TSO and a TSV. The combination of the shown alternatives for each step leads to four different Transfer Strategies as follows:

- OS+VS = $\{TSO_{single}, TSV_{single}\}$
- OS+VA = $\{TSO_{single}, TSV_{all}\}$
- OA+VS = $\{TSO_{all}, TSV_{single}\}$
- OA+VA = $\{TSO_{all}, TSV_{all}\}$

As shown above, the three different interaction graphs for the three objective functions (Figure 1) result in two combined graphs by using the respective TSO (Figures 2a and 2b). In the next step, the TSV obtain variable groups out of these. As a result, the following groups will be created by the four strategies:

- OS+VS: $\{x_1, x_2, x_3, x_4\}$
- OS+VA: $\{x_1, x_3\}\{x_2, x_4\}$
- OA+VS: $\{x_1, x_3, x_4\}\{x_2\}$
- OA+VA: $\{x_1, x_3\}\{x_2\}\{x_4\}$

While the strategies containing TSO_{single} have been explained above, the strategies OA+VS and OA+VA make use of the interaction graph shown in Figure 2b. In the first case, using TSV_{single}, a single connection to a component is sufficient and the resulting groups are $\{x_1, x_3, x_4\}$ and $\{x_2\}$. If TSV_{all} is used, the subgraph containing x_1, x_3 and x_4 is not complete, and therefore only x_1 and x_3 are grouped together.

4.4 Relationship to Existing Methods

We now briefly compare the proposed TS to the existing approaches used in MOEA/DVA and LMEA. Both of these methods do multiple repeated checks of Equation 2 with different values for the variables to determine interaction. When combining the information of the objective functions, variables were regarded as interacting if a single interaction in any of the functions was found. This corresponds to TSO_{single} in our approach. In MOEA/DVA, after building this combined interaction graph, maximally connected subgraphs are identified as the variable groups, which is equivalent to TSV_{single}. The same is true for LMEA. As described above, each variable is checked against already formed groups and added if only one interaction within the group is found. As a result, both MOEA/DVA and LMEA use equivalent approaches to the transfer strategy OS+VS.

5 EVALUATION AND ANALYSIS

In this section we evaluate the proposed TS experimentally. For this purpose we use three most recent large-scale algorithms for multi-objectve optimisation: MOEA/DVA, LMEA and WOF. A direct comparion of the performance of these three and their advantages and disadvantages has been studied in the literature (e.g. [21]). The aim of our experiments is therefore not mainly to compare the algorithms with each other, but to examine the influence of different variable groups on their respective performance. A central aspect of this is not only the final solution quality, but also to set this performance into perspective with the necessary computational budget. For the experiments, we replace the original grouping mechanism inside the three algorithms with DG2, using the four different TS to apply it to the multi-objective case. In total, each of the three algorithms is used in six different configurations. First the original algorithms are used. Secondly, we apply the proposed four TS to DG2 and replace the original grouping mechanisms in the algorithms with it, resulting in four more versions. Lastly, for comparison we apply a random grouping to all algorithms, which splits the variables into evenly-sized groups randomly.

As described above, MOEA/DVA and LMEA first perform a grouping mechanism to decide whether a decision variable contributes to diversity or convergence. Secondly, the variables related to convergence towards the optimal front are then grouped by their interactions. Since we try to make single-objective methods like DG2 applicable, which focus on the interaction of variables, in our modified versions of the three algorithms this second step is replaced by either DG2 with one of four TS, or by a random grouping. WOF in its original form does not perform a separation in diversity- and convergence-related variables. Therefore its usual grouping phase is replaced entirely. As a result, DG2 is always used on the entirety of variables, while LMEA and MOEA/DVA only use DG2 for a specific set of variables, i.e. the ones they regard as convergence-related. Note that in the case of MOEA/DVA, the original version updates the population during the interaction grouping. To ensure fairness we make sure every algorithm uses the same amount of evaluations for the optimisation process. Therefore, the initial population of MOEA/DVA (that was created before the interaction grouping) is used to start the optimisation process with the found groups. In the following, this modified version of MOEA/DVA is used. A comparison with the original version (using multiple millions of evaluations to update the population) is given as reference in Table 12 of the supplement material.

To test the performance we use 12 common benchmarks: six problems from the popular WFG (Walking Fish Group) family [4] (WFG2-5,7,8) and six problems from the LSMOP family [3] (LSMOP1-6). The WFG problems were chosen by an analysis done in [7], were WFG2 and 3 represent problems where the number of interacting variables is sparse, WFG4 and 5 have no interactions and WFG7 and 8 have a high number of interacting variables. For implementation, the PlatEMO framework (Evolutionary Multi-objective Optimization Platform) [13] version 1.5 was used.

For each experiment we perform 31 independent runs and report the median and interquartile range (IQR) values of the relative hypervolume (HV) indicator [14]. The relative HV is the hypervolume obtained by a solution set in relation to the hypervolume obtained by a sample of the Pareto-front of the problem. This sample consists of 10,000 solutions sampled on the true Pareto-front using the PlatEMO framework. The used reference point for the indicator is obtained by using the nadir point of our Pareto-front sample and

Table 1: Relative HV of the original LMEA and its variants using DG2 and random groups

	LMEA	DG2 (OA+VA)	DG2 (OA+VS)	DG2 (OS+VA)	DG2 (OS+VS)	Random
LSMOP1	0.49267 (1.64E-2)	0.48892 (1.44E-2)	0.48786 (1.43E-2)	**0.49299** (1.20E-2)	0.49038 (1.67E-2)	— (3.91E-2) *
LSMOP2	0.97873 (4.74E-3)	*0.97729* (3.56E-3) *	0.97760 (4.28E-3) *	**0.98088** (4.98E-3)	0.98005 (4.85E-3)	0.97801 (6.60E-3) *
LSMOP3	— (—)	— (—)	— (—)	— (—)	— (—)	— (—)
LSMOP4	0.97458 (4.52E-3)	**0.97670** (5.78E-3)	0.97612 (4.04E-3)	0.96840 (3.65E-3) *	0.96862 (3.76E-3) *	*0.96769* (5.83E-3) *
LSMOP5	— (—)	— (—)	— (—)	— (—)	— (—)	— (—)
LSMOP6	— (—)	— (—)	— (—)	— (—)	— (—)	— (—)
WFG2	0.59776 (5.72E-2) *	0.62466 (3.24E-2)	**0.62928** (5.91E-2)	0.60756 (5.57E-2)	0.61129 (4.07E-2)	*0.58214* (6.71E-2) *
WFG3	0.60883 (5.58E-3)	0.61090 (6.46E-3)	**0.61211** (5.01E-3)	0.61101 (4.12E-3)	0.61087 (3.50E-3)	*0.58549* (5.66E-3) *
WFG4	0.56200 (6.14E-3) *	0.56262 (7.35E-3)	0.56155 (5.44E-3) *	*0.56134* (4.54E-3) *	0.56253 (4.65E-3)	**0.56532** (6.55E-3)
WFG5	0.55891 (7.57E-3)	*0.55664* (5.03E-3) *	0.55787 (6.95E-3) *	0.55962 (8.75E-3)	0.55955 (6.10E-3)	**0.56156** (4.47E-3)
WFG7	*0.52548* (3.47E-2) *	0.61516 (3.29E-2)	**0.61679** (2.29E-2)	0.61631 (2.48E-2)	0.61282 (1.74E-2)	0.53816 (2.13E-2) *
WFG8	*0.53369* (6.77E-3) *	0.54550 (7.85E-3)	**0.54777** (7.07E-3)	0.54532 (4.76E-3)	0.54435 (5.20E-3)	0.54054 (4.43E-3) *

Table 2: Relative HV of the original MOEA/DVA and its variants using DG2 and random groups

	MOEA/DVA	DG2 (OA+VA)	DG2 (OA+VS)	DG2 (OS+VA)	DG2 (OS+VS)	Random
LSMOP1	— (—) *	— (—) *	— (—) *	— (—) *	— (—) *	**0.39075** (2.52E-1)
LSMOP2	0.98886 (2.92E-4)	0.98887 (2.93E-4)	**0.98887** (4.82E-4)	0.98879 (3.57E-4)	0.98876 (4.41E-4)	*0.98875* (2.59E-4)
LSMOP3	— (—) *	— (—) *	— (—) *	— (—) *	— (—) *	**0.56864** (3.51E-3)
LSMOP4	**0.97313** (1.02E-3)	0.80549 (1.08E-3) *	*0.80478* (1.65E-3) *	0.97246 (1.28E-3)	0.97277 (1.04E-3)	0.97298 (1.35E-3)
LSMOP5	— (—) *	— (—) *	— (—) *	— (—) *	— (—) *	**0.08203** (7.51E-2)
LSMOP6	— (—) *	**0.26054** (3.88E-1)	0.08484 (3.59E-1)	— (1.41E-1) *	— (—) *	— (5.78E-1)
WFG2	0.74352 (6.18E-3) *	0.73636 (1.57E-3) *	0.73624 (1.32E-3) *	*0.73599* (1.76E-3) *	0.73744 (5.26E-3) *	**0.75616** (1.00E-2)
WFG3	*0.57625* (3.33E-3) *	0.57676 (3.11E-3) *	0.57881 (3.67E-3) *	0.57691 (3.39E-3) *	0.57657 (6.91E-3) *	**0.66822** (8.09E-3)
WFG4	*0.54052* (3.10E-3) *	0.54202 (4.31E-3) *	0.54087 (6.18E-3) *	0.54097 (4.83E-3) *	0.54168 (6.32E-3) *	**0.76804** (3.10E-2)
WFG5	*0.54942* (6.15E-3) *	0.55239 (1.01E-2) *	0.54950 (5.70E-3) *	0.55056 (7.41E-3) *	0.55009 (7.91E-3) *	**0.84546** (8.37E-3)
WFG7	0.73057 (1.49E-2)	0.72937 (9.37E-3)	0.73318 (8.84E-3)	0.73049 (5.42E-3)	*0.72834* (8.41E-3)	**0.73387** (1.23E-2)
WFG8	*0.49228* (4.85E-3) *	0.50822 (3.48E-3) *	0.50771 (6.19E-3) *	0.50813 (6.89E-3) *	0.50790 (4.81E-3) *	**0.58759** (3.73E-2)

Table 3: Relative HV of the original WOF-SMPSO and its variants using DG2 and random groups

	WOF-SMPSO	DG2 (OA+VA)	DG2 (OA+VS)	DG2 (OS+VA)	DG2 (OS+VS)	Random
LSMOP1	0.96588 (1.40E-3)	0.08886 (4.12E-2) *	0.08111 (3.61E-2) *	*0.07542* (3.69E-2) *	0.82963 (4.01E-3) *	**0.96613** (1.66E-3)
LSMOP2	**0.99620** (2.85E-4)	*0.98835* (2.88E-4) *	0.98839 (2.64E-4) *	0.99498 (4.17E-4) *	0.99535 (1.72E-4) *	0.99615 (2.89E-4)
LSMOP3	0.48418 (4.14E-3)	— (—) *	— (—) *	— (—) *	0.01911 (2.07E-3) *	**0.48429** (4.85E-3)
LSMOP4	0.99052 (2.98E-4) *	*0.97600* (7.95E-4) *	0.97608 (7.23E-4) *	**0.99279** (4.01E-4)	0.98910 (1.14E-4) *	0.99049 (2.75E-4) *
LSMOP5	0.92180 (4.48E-3)	— (—) *	0.29676 (3.09E-2) *	— (—) *	0.62224 (—) *	**0.92384** (3.63E-3)
LSMOP6	0.95063 (2.14E-3)	0.61632 (1.45E-1) *	0.91998 (2.70E-3) *	*0.59713* (4.99E-3) *	0.93066 (4.03E-3) *	**0.95117** (1.87E-3)
WFG2	**0.99231** (1.37E-3)	*0.68632* (7.73E-2) *	0.75684 (7.89E-2) *	0.93138 (4.79E-3) *	0.92991 (4.78E-3) *	0.99163 (2.14E-3)
WFG3	**0.85517** (6.43E-4)	0.69960 (6.62E-3) *	*0.69448* (9.93E-3) *	0.69465 (1.14E-2) *	0.69902 (1.29E-2) *	0.85512 (8.74E-4)
WFG4	0.98122 (7.39E-3)	0.85246 (3.61E-3) *	0.85394 (5.17E-3) *	*0.85215* (3.89E-3) *	0.85450 (5.20E-3) *	**0.98141** (7.54E-3)
WFG5	**0.98687** (1.91E-4)	0.82627 (4.86E-3) *	0.82676 (5.96E-3) *	*0.82439* (3.76E-3) *	0.82635 (4.32E-3) *	0.98685 (2.72E-4)
WFG7	**0.98846** (1.38E-3)	*0.71462* (1.01E-2) *	0.71758 (1.62E-2) *	0.71943 (1.01E-2) *	0.71730 (1.77E-2) *	0.98802 (1.15E-3)
WFG8	**0.94936** (1.28E-2)	0.50485 (1.50E-2) *	0.51209 (1.86E-2) *	*0.50466* (1.58E-2) *	0.50826 (1.51E-2) *	0.94916 (3.84E-2)

Table 4: Computational budget (function evaluations) to obtain variable groups by the different methods

	LMEA	LMEA + DG2	MOEA/DVA	MOEA/DVA + DG2	WOF-SMPSO	WOF-SMPSO + DG2
LSMOP1	9,089,228 (—)	513,564 (—)	9,119,490 (—)	525,736 (—)	0 (—)	506,522 (—)
LSMOP2	9,089,228 (—)	513,564 (—)	9,119,490 (—)	62,416 (—)	0 (—)	506,522 (—)
LSMOP3	9,086,855 (2.31E2)	513,564 (—)	9,119,490 (—)	525,736 (—)	0 (—)	506,522 (—)
LSMOP4	9,088,364 (3.00E2)	513,564 (—)	9,119,490 (—)	525,736 (—)	0 (—)	506,522 (—)
LSMOP5	4,602,638 (—)	264,019 (—)	9,119,490 (—)	525,736 (—)	0 (—)	506,522 (—)
LSMOP6	9,088,280 (1.17E2)	513,564 (—)	9,119,490 (—)	62,416 (—)	0 (—)	506,522 (—)
WFG2	5,059,705 (2.19E2)	289,634 (—)	9,029,120 (—)	397,267 (1.03E4)	0 (—)	501,502 (—)
WFG3	5,059,690 (8.40E1)	289,634 (—)	9,029,120 (—)	301,746 (—)	0 (—)	501,502 (—)
WFG4	5,063,750 (—)	289,626 (—)	9,011,100 (—)	301,726 (—)	0 (—)	500,501 (—)
WFG5	5,063,750 (—)	289,626 (—)	9,011,100 (—)	301,726 (—)	0 (—)	500,501 (—)
WFG7	272,039 (4.36E5)	102,831 (7.87E4)	9,011,100 (—)	20,101 (—)	0 (—)	500,501 (—)
WFG8	240,167 (3.28E4)	289,626 (—)	9,011,100 (—)	301,726 (—)	0 (—)	500,501 (—)

multiply it by 2.0 in each dimension. This is done to make sure most of the obtained solutions can contribute to the HV, even when the sets are not close to the optimal front. Statistical significance is tested using a Mann-Whitney-U Test and significance is assumed for a value of $p < 0.01$.

5.1 Parameter Settings

All experiments use two objective-instances of the mentioned problems with $n = 1000$ decision variables for WFG4,5,7 and 8, 1001 variables for WFG2 and 3, and 1006 variables for LSMOP1-6. In the WFG problems, the number of position-related variables have been set to $n/4$ for the WFG problems and the parameter n_k in the LSMOP benchmarks was set to 5.

All algorithms used a population size of 100. For all parameters, the standard values from the related works are used where possible. For MOEA/DVA, we set $NIA = 6$ and $NCA = 20$. In LMEA we set $nSel = 2$, $nPer = 4$, and $nCor = 6$. WOF uses a linear grouping mechanism with 4 equal-sized groups, and SMPSO is used as the optimiser inside the framework. Solutions for creating the q transformed problems are chosen by reference directions as reported in [21], q is set to $m + 1$, and the parameter-free transformation function and the restart mechanism from [21] are used. t_1, t_2 and δ

are set to 1000, 500 and 0.5 respectively. The population sizes for the transformed problems of WOF are set to 10. The distribution index used in all operators is set to 20.0 and the probabilities for Crossover and Mutation are set to 1.0 and $1/n$. For all other parameters, the standard values of the PlatEMO framework are used.

To test the influence of the groups, our experiments differ from other studies in the way that all methods are equipped with the same amount of function evaluations for the optimisation. That is, all algorithms use 500,000 function evaluations in their optimisation phase independently of their computational budget spent on obtaining the variable groups beforehand. As part of the analysis, the actual used evaluations for these steps are shown in Table 4.

5.2 Results

In this subsection, we analyse the outcome of the optimisation process and compare the original algorithms with the DG2 and random grouping versions. In addition, we pay attention to the necessary computational budgets spent for obtaining the variable groups, and set these into perspective with the corresponding performance.

In Table 1 we have listed the results of the LMEA algorithm and its variants. The first column gives the median and IQR values of the relative HV for the original LMEA algorithm. In the following columns, the grouping mechanism has been replaced by different DG2 strategies or a random grouping as described above. Tables 2 and 3 follow the same structure for the algorithms MOEA/DVA and WOF respectively. Table 4 lists the amount of evaluations each method spent to obtain the groups. Best results are shown in bold, while worst performance is given in italic font. An asterisk denotes statistical significance compared to the respective best performance in each row.

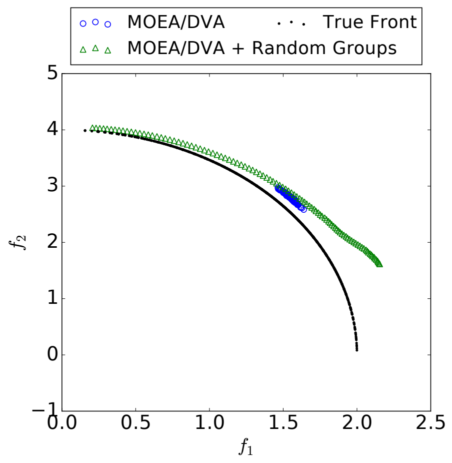


Figure 3: Final solutions sets for MOEA/DVA and MOEA/DVA with random grouping on the WFG5 problem. Shown runs obtained the median rel. HV values.

Regarding LMEA, it can be seen that the original version does not yield a best performance in any of the 12 problems. However, only in WFG2, 4 7 and 8 these differences are actually statistically significant. None of the LMEA variants was able to achieve a HV value for the LSMOP 3, 5 and 6 benchmarks, where the budget of 500,000 evaluations seemed not sufficient to achieve a solution reasonably close to the Pareto-front sample. When we take a closer look at LMSOP2, the TS including OS perform better than their OA counterparts. In this case, the TSV strategies do not influence the result much, the performance only depends on the selected TSO. For LSMOP4 the same behaviour can be noticed, only now the OA methods perform significantly better. However, the differences between the original LMEA and the respective best DG2 or random version are not significant in LSMOP1, 2, 4 and WFG3 and 5. Therefore, their final outcomes can be assumed as equal, although these numbers have to be seen in relation to the computational budget spent to achieve them. All variants used the same amount of evaluations in the optimisation step, but spent a different amount for obtaining their groups first (Table 4). The original LMEA spent more than 17 times more evaluations than the DG2 variants using the TS, and achieved the same or significantly worse performance in all 9 problems where HV values were obtained. In WFG4 and 5, the best performance was achieved using random groups, which needed no evaluations at all to be computed. In WFG2, 7 and 8 the best results were obtained by DG2 with the OA+VS strategy with statistical significance to the original LMEA and the version with random groups. On the other hand, random groups performed best on WFG4 and 5. This fits to the observation in [7] that there are no variable interactions in WFG4 and 5, therefore random groups can result in good performance, while WFG2, 7 and 8 show interactions and a more sophisticated grouping mechanism is necessary. We can conclude from these numbers that the original LMEA can be largely improved by the proposed transfer strategies and DG2, not necessarily in solution quality for all problems, but by a large factor in the computational budget needed to achieve these results.

Looking at MOEA/DVA, the most striking observation is the good performance of random groups compared to MOEA/DVA and its DG2 variants. In 8 out of 12 problems, MOEA/DVA with random groups performs significantly better than the original algorithm or any of the DG2 versions. Comparing the corresponding sizes and numbers of groups obtained (Tables 6 to 8 in the supplements), we see that the original MOEA/DVA creates multiple hundreds of groups for all these problems, mostly containing only between 1 and 3 variables each. The random grouping in contrast creates four groups of relatively large size and this could be one of the leading factors for its good performance. In most problems the large number of small groups in MOEA/DVA and its DG2 variants leads to a search behaviour of the algorithm where the above-mentioned uniformity optimisation is not carried out before the end of the optimisation process. This results in relatively low diversity of the final solution sets compared to the random grouping version. As an example, we show the final solution sets of the MOEA/DVA and the MOEA/DVA with random groups in Figure 3.

In the results of MOEA/DVA not much differences between the four TS can be noticed. In LSMOP4 the same scheme occurs that was observed in LMEA: the TSO is crucial for the result. However, in contrast to the results of LMEA, here the OS methods perform significantly better than their OA counterparts. This is also one of the rare cases, where the difference in HV between TS is rather high. A reason for this could be that the TS containing OA lead to potentially smaller groups. On the other hand, for LSMOP6, the OA methods are the only TS that create positive HV values, although these results seem not to be stable over multiple runs as indicated by the very large IQR values. When using MOEA/DVA on the WFG problems, no notable differences between the TS can be noticed.

In the results of WOF (Table 3), with the exception of LSMOP4, the best performance is always achieved by either the original WOF (using linear groups) or the version with random groups, both of which do not spend any budget on obtaining groups. Both of these versions use exactly four evenly sized groups, while the DG2 with the four TS mostly produces a much larger amount of groups (150 - 1000) or puts all variables in one large group (see also Table 10 in the supplement material). This good performance of random or linear groups can be observed in problems without interactions (e.g. LSMOP1 and 5, and WFG4 and 5 according to [3, 7]), but also in the problems that contain interacting variables, like for instance LSMOP6. When comparing the different TS with each other, interesting results can also be observed for LSMOP5 and 6. LSMOP5 does not contain any interacting variables according to [3], while LSMOP6 does. In both of these problems, the TSV seems to be the major influence. The strategies containing VS obtain by far better solutions compared to the VA strategies. Although in MOEA/DVA and LMEA, in some cases the choice of the TSO was more important than the TSV, this case shows that the choice of TSV can also affect the performance by a large factor. Overall, the results indicate that a smaller number of larger groups leads to a better performance of the WOF algorithm, regardless of the interaction of the variables. This also matches observations made by the original authors of the algorithm in [20].

In general, the differences in HV between the four proposed TS are rather small in most problems, although sometimes statistically significant. Overall, none of the four TS can be regarded as superior for all problems and the utility of the TS depends on each problems properties. It can be observed however, that exchanging the grouping mechanisms in MOEA/DVA and LMEA with any of the proposed TS with DG2 does in most cases not change the performance. However, the overall computational budget for obtaining the groups is reduced by a large factor.

6 CONCLUSION

In conclusion, in most problem instances the importance of correct variable groups is relatively low, especially when taking the necessary computational budget into account. Depending on the used algorithm, acceptable results can often be obtained using random groups. When equipped with the same computational budget, the proposed TS using DG2 were able to improve the performance of the MOEA/DVA and LMEA search mechanisms in some cases. In addition, the necessary budget for running DG2 is by far lower than the MOEA/DVA and LMEA grouping mechanisms. In general, the best performance among all methods is in most cases obtained by WOF using linear or random groups. Future research might include experiments with varying numbers of variables and objective functions as well as fine tuning of the algorithms' parameters for different grouping methods. More benchmark problems might be taken into account. In particular, the importance of correctly grouped variables needs further consideration, as our results indicate that a large number of small non-interacting groups does not necessarily lead to better performance of the current state-of-the-art algorithms, and that very good results can often be achieved using random groups.

REFERENCES

[1] Luis Miguel Antonio and Carlos A Coello Coello. 2013. Use of Cooperative Coevolution for Solving Large Scale Multiobjective Optimization Problems. In *IEEE Congress on Evolutionary Computation (CEC)*. 2758–2765.

[2] Wenxiang Chen, Thomas Weise, Zhenyu Yang, and Ke Tang. 2010. Large-Scale Global Optimization Using Cooperative Coevolution with Variable Interaction Learning. In *Parallel Problem Solving from Nature, PPSN XI*, Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph (Eds.). Lecture Notes in Computer Science, Vol. 6239. Springer Berlin Heidelberg, 300–309.

[3] Ran Cheng, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. 2017. Test Problems for Large-Scale Multiobjective and Many-Objective Optimization. *IEEE Transactions on Cybernetics* 47, 12 (Dec 2017), 4108–4121.

[4] Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. 2006. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation* 10, 5 (2006), 477–506.

[5] Xiaodong Li and Xin Yao. 2009. Tackling High Dimensional Nonseparable Optimization Problems By Cooperatively Coevolving Particle Swarms. In *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1546–1553.

[6] Xiaodong Li and Xin Yao. 2012. Cooperatively Coevolving Particle Swarms for Large Scale Optimization. *IEEE Transactions on Evolutionary Computation* 16, 2 (2012), 210–224.

[7] Xiaoliang Ma, Fang Liu, Yutao Qi, Xiaodong Wang, Lingling Li, Licheng Jiao, Minglei Yin, and Maoguo Gong. 2016. A Multiobjective Evolutionary Algorithm Based on Decision Variable Analyses for Multiobjective Optimization Problems With Large-Scale Variables. *IEEE Transactions on Evolutionary Computation* 20, 2 (2016), 275–298.

[8] Sedigheh Mahdavi, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. 2015. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences* 295 (2015), 407–428.

[9] Antonio J. Nebro, Juan J. Durillo, Paulino J. Garcia Nieto, Carlos A. Coello Coello, Francisco Luna, and Enrique Alba. 2009. SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. In *Symposium on Computational Intelligence in Multi-criteria Decision-making*. IEEE, 66–73.

[10] Mohammad Nabi Omidvar, Xiaodong Li, Yi Mei, and Xin Yao. 2014. Cooperative Co-Evolution With Differential Grouping for Large Scale Optimization. *IEEE Transactions on Evolutionary Computation* 18, 3 (2014), 378–393.

[11] Mohammad N. Omidvar, Ming Yang, Yi Mei, Xiaodong Li, and Xin Yao. 2017. DG2: A Faster and More Accurate Differential Grouping for Large-Scale Black-Box Optimization. *IEEE Transactions on Evolutionary Computation* PP, 99 (2017), 1–1.

[12] Mitchell A. Potter and Kenneth A. Jong. 1994. A Cooperative Coevolutionary Approach to Function Optimization. Lecture Notes in Computer Science, Vol. 866. Springer Berlin Heidelberg, 249–257.

[13] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. 2017. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *CoRR* abs/1701.00879 (2017). http://arxiv.org/abs/1701.00879

[14] Lyndon While, Philip Hingston, Luigi Barone, and Simon Huband. 2006. A faster algorithm for calculating hypervolume. *IEEE transactions on evolutionary computation* 10, 1 (2006), 29–38.

[15] Zhenyu Yang, Ke Tang, and Xin Yao. 2008. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences* 178, 15 (2008), 2985–2999.

[16] Zhenyu Yang, Jingqiao Zhang, Ke Tang, Xin Yao, and Arthur C. Sanderson. 2009. An adaptive coevolutionary Differential Evolution algorithm for large-scale optimization. In *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 102–109.

[17] Xingyi Zhang, Ye Tian, Yaochu Jin, and Ran Cheng. 2018. A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 22, 1 (Feb 2018), 97–112. https://doi.org/10.1109/TEVC.2016.2600642

[18] Heiner Zille, Hisao Ishibuchi, Sanaz Mostaghim, and Yusuke Nojima. 2016. Mutation operators based on variable grouping for multi-objective large-scale optimization. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*.

[19] Heiner Zille, Hisao Ishibuchi, Sanaz Mostaghim, and Yusuke Nojima. 2016. Weighted Optimization Framework for Large-scale Multi-objective Optimization. In *Companion of Genetic and Evolutionary Computation Conference - GECCO*. ACM.

[20] Heiner Zille, Hisao Ishibuchi, Sanaz Mostaghim, and Yusuke Nojima. 2018. A Framework for Large-scale Multi-objective Optimization based on Problem Transformation. *IEEE Transactions on Evolutionary Computation* 22, 2 (April 2018), 260–275. https://doi.org/10.1109/TEVC.2017.2704782

[21] Heiner Zille and Sanaz Mostaghim. 2017. Comparison Study of Large-scale Optimisation Techniques on the LSMOP Benchmark Functions. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*.