



Fitness-based Linkage Learning and Maximum-Clique Conditional Linkage Modelling for Gray-box Optimization with RV-GOMEA

Georgios Andreadis
Leiden University Medical Center
Leiden, The Netherlands
G.Andreadis@lumc.nl

Tanja Alderliesten
Leiden University Medical Center
Leiden, The Netherlands
T.Alderliesten@lumc.nl

Peter A.N. Bosman
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
Peter.Bosman@cwi.nl

ABSTRACT

For many real-world optimization problems it is possible to perform partial evaluations, meaning that the impact of changing a few variables on a solution's fitness can be computed very efficiently. It has been shown that such partial evaluations can be excellently leveraged by the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA) that uses a linkage model to capture dependencies between problem variables. Recently, conditional linkage models were introduced for RV-GOMEA, expanding its state-of-the-art performance even to problems with overlapping dependencies. However, that work assumed that the dependency structure is known a priori. Fitness-based linkage learning techniques have previously been used to detect dependencies during optimization, but only for non-conditional linkage models. In this work, we combine fitness-based linkage learning and conditional linkage modelling in RV-GOMEA. In addition, we propose a new way to model overlapping dependencies in conditional linkage models to maximize the joint sampling of fully interdependent groups of variables. We compare the resulting novel variant of RV-GOMEA to other variants of RV-GOMEA and V_kD-CMA on 12 problems with varying degree of overlapping dependencies. We find that the new RV-GOMEA not only performs best on most problems, also the overhead of learning the conditional linkage models during optimization is often negligible.

CCS CONCEPTS

• **Mathematics of computing** → **Evolutionary algorithms.**

KEYWORDS

Gray-box optimization, GOMEA, linkage modeling

ACM Reference Format:

Georgios Andreadis, Tanja Alderliesten, and Peter A.N. Bosman. 2024. Fitness-based Linkage Learning and Maximum-Clique Conditional Linkage Modelling for Gray-box Optimization with RV-GOMEA. In *Genetic and Evolutionary Computation Conference (GECCO '24)*, July 14–18, 2024, Melbourne, VIC, Australia. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3638529.3654103>



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

GECCO '24, July 14–18, 2024, Melbourne, VIC, Australia
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0494-9/24/07.
<https://doi.org/10.1145/3638529.3654103>

1 INTRODUCTION

Optimizing real-world problems using a Black-Box Optimization (BBO) perspective is generally applicable, but this perspective ignores problem-specific knowledge, despite it being available in many real-world optimization settings [5, 13, 14]. In a Gray-Box Optimization (GBO) setting, any such knowledge is provided to the optimization method, which can take many forms. It has been demonstrated that Evolutionary Algorithms (EAs) can be adjusted to exploit such knowledge to improve performance, using techniques such as partition crossover operators [24], problem-specific variation operators [13], and partial evaluations [12]. The latter assumes that it is possible to efficiently recalculate the objective value of a solution when a subset of its variables is modified.

For various real-world optimization problems, such as medical deformable image registration [5] and internal radiation treatment planning for cancer [14], the evaluation of solutions can be decomposed into such partial evaluations. Despite their decomposability, however, these problems remain difficult to solve, often having characteristics such as non-smoothness, non-separability, and multimodality. The Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA) has shown excellent performance on these real-world GBO problems [5, 14], as well as on benchmark GBO problems [10, 12]. On optimization problems with strong overlapping dependencies, the recent introduction of conditional linkage models in RV-GOMEA has proven to be crucial to maintaining this performance improvement over a black-box approach [11].

These conditional linkage models currently require the Variable Interaction Graph (VIG) to be provided, as dependencies between variables are core to conditional distribution sampling. However, the interactions between problem variables may not be fully known in advance. Even if a generic VIG formulation can be made, not all edges of this VIG may materialize as (strong) fitness dependencies during optimization, and fitness dependency structures may change over time. In these cases, it may prove beneficial to empirically learn the VIG of a problem during optimization, as has been proposed in discrete optimization [23]. On top of potentially speeding up optimization, this could unlock problem-specific, structural insights, which could be transferred to other problem instances. For these instances, linkage then no longer needs to be detected during optimization, resulting in further speed-ups.

In this work, we bring the performance benefits of conditional linkage-models in RV-GOMEA [11] to GBO problems where the VIG is not known a priori. To this end, we explore the use of fitness-based dependency tests to incrementally build a Dependency Strength Matrix (DSM), describing pairwise variable interactions. These tests have recently shown promise within RV-GOMEA on separable

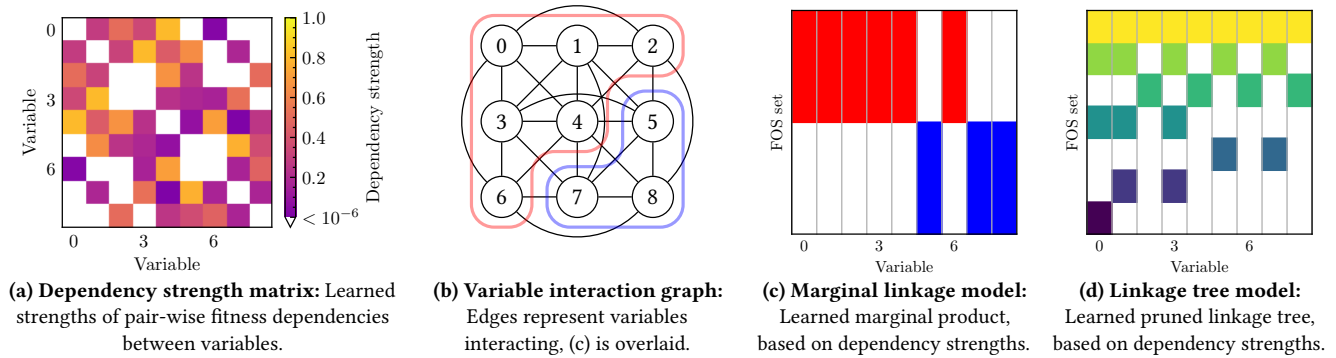


Figure 1: A case study of fitness-based linkage learning: the REBGrid problem described in Section 5.1, with $\ell = 9$.

problems [20], but have not yet been combined with conditional models. Furthermore, we introduce a novel linkage model that models fully interdependent groups of variables jointly, maximizing the exploitation of sub-functions in the variation operator of GOMEA. In experiments, we compare the performance of RV-GOMEA using these models to VxD-CMA [2, 3], which is a state-of-the-art optimization method for problems with overlapping dependencies. Although VxD-CMA is not able to leverage partial evaluations, it has a dimensionality-reduction mechanism capable of tackling high-dimensional problems with overlapping dependencies.

This paper is structured as follows: First, we introduce the problem setting in Section 2 and RV-GOMEA in Section 3. In Section 4, we describe how linkage models can be learned using fitness-based dependency tests. We evaluate and compare the proposed linkage models in RV-GOMEA to existing linkage models and VxD-CMA, in Sections 5 and 6. Finally, we discuss our results in Section 7 and conclude in Section 8.

2 GRAY-BOX OPTIMIZATION

We consider an objective function $f(x) : \mathbb{R}^\ell \rightarrow \mathbb{R}$ which is to be minimized. We refer to the variables of the problem as $X = \{X_0, \dots, X_{\ell-1}\}$, indexed by the set of indices $\mathcal{I} = [0, \dots, \ell - 1]$. A realization of X , i.e., a solution, is denoted by $x = \{x_0, \dots, x_{\ell-1}\}$.

The GBO problems considered in this work allow for partial evaluations, which is the efficient computation of the objective value of a solution after a subset of its variables has been modified [8]. Let $Y \subseteq \mathcal{I}$ denote the indices of this subset of variables, and x_Y the corresponding variables. The objective function $f(x)$ is then composed of q sub-functions $F = \{f_0, \dots, f_{q-1}\}$. Each sub-function $f_j(x_{\mathbb{I}_j}) \in F$ takes variables x_i with $i \in \mathbb{I}_j \subseteq \mathcal{I}$, where sub-function index sets $\mathbb{I} = \{\mathbb{I}_0, \dots, \mathbb{I}_{q-1}\}$ are given by the problem definition and can overlap. These sub-functions are considered not to be separable themselves, and are therefore treated as a black box.

As such, a GBO objective function can be expressed as follows:

$$f(x) = f_0(x_{\mathbb{I}_0}) \oplus f_1(x_{\mathbb{I}_1}) \oplus \dots \oplus f_{q-1}(x_{\mathbb{I}_{q-1}}), \quad (1)$$

with \oplus denoting a binary operator with a known inverse operator \ominus (such as addition or multiplication).

When one variable x_u is changed, all sub-functions $f_j(x_{\mathbb{I}_j})$ for which $u \in \mathbb{I}_j$ holds, need to be evaluated. We consider the computational complexity of a sub-function f_j to be approximately proportional to the number of involved variables $|\mathbb{I}_j|$. Therefore,

the cost of a partial evaluation for f_j is calculated as $|\mathbb{I}_j|/|\mathcal{I}|$ in this setting. This deviates from a previous definition [11] which does not take sub-function size into account. The definition used in this work portrays problems with heterogeneous sub-function index set sizes and strongly overlapping dependencies more accurately.

In this gray-box structure, we consider two different variables x_u and x_v with indices $u, v \in \mathcal{I}$ to be directly dependent when there exists a sub-function f_j with $\{u, v\} \subseteq \mathbb{I}_j$. This dependency is denoted by $x_u \leftrightarrow x_v$. Variables x_u and x_v are considered indirectly dependent when there exists a set $\{u, \dots, v\} \subseteq \mathcal{I}$ for which $x_u \leftrightarrow \dots \leftrightarrow x_v$ holds, but not $x_u \leftrightarrow x_v$. Given the dependencies in a problem, a so-called Variable Interaction Graph (VIG) [29] can be constructed. The VIG of a problem is an undirected graph $G = (V, E)$, with each vertex $v \in V$ corresponding to the variable x_v and an edge $(u, v) \in E$ exists for every pair of directly dependent variables x_u and x_v . Indirect dependencies can be translated to this graph representation as follows: if x_u and x_v are indirectly independent, there exists a path between u and v , but not an edge (u, v) . As illustration, the VIG of the REBGrid problem (defined in Section 5.1) is depicted in Figure 1b.

3 REAL-VALUED GENE-POOL OPTIMAL MIXING EVOLUTIONARY ALGORITHM

In this section, we present an overview of RV-GOMEA. The algorithm is described in more detail in [12].

3.1 Family of Subsets

The dependencies between variables X_i are modelled explicitly by a linkage model in RV-GOMEA. This linkage model is described by a Family of Subsets (FOS) $\mathcal{F} = \{\mathcal{F}_0, \mathcal{F}_1, \dots, \mathcal{F}_{m-1}\}$, with $\mathcal{F}_i \subseteq \mathcal{I}$. The elements \mathcal{F}_i of this FOS each represent a group of variables which are deemed to be jointly dependent.

Two generic FOS linkage models can be defined regardless of the problem structure: a univariate FOS $\mathcal{F} = \{\{0\}, \{1\}, \dots, \{\ell - 1\}\}$, which models all variables to be independent, and a full FOS $\mathcal{F} = \{\{0, \dots, \ell - 1\}\}$, which models all variables as jointly dependent. More generally, we can define a marginal product FOS model, as a set of disjoint sets of dependent variables, i.e., $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for all $\mathcal{F}_i, \mathcal{F}_j \in \mathcal{F}$ where $i \neq j$. An example is visualized in Figure 1c.

As a generic solution that can capture different orders of dependency at once, the linkage tree FOS model has been introduced [12, 28]. To construct such an FOS, all singleton elements are

first included. Then, elements are recursively merged into larger elements \mathcal{F}_k , for which the following holds: $\mathcal{F}_i, \mathcal{F}_j \in \mathcal{F}$ exist ($i \neq j \neq k$) such that $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ and $\mathcal{F}_i \cup \mathcal{F}_j = \mathcal{F}_k$. To decide which $\mathcal{F}_i, \mathcal{F}_j$ to merge, the Unweighted Pair Grouping Method with Arithmetic-mean (UPGMA) [16] is commonly used. This method is supplied with dependency strength information, which can be provided *a priori* or can be discovered during optimization, for instance by measuring mutual information in the population, although this technique has proven to be ineffective in continuous optimization [20].

A linkage tree FOS can be restricted in size from two directions. From the root of the tree, its elements can be bounded to a maximum element size, which is then referred to as a bounded linkage tree [12]. From the leaves of the tree upward, the model can exclude smaller elements which are better solely modelled jointly, referred to as a pruned linkage tree [20]. This pruning is iteratively performed until there exist no $\mathcal{F}_i, \mathcal{F}_j, \mathcal{F}_k$ ($i \neq j \neq k$) for which the following two conditions hold: $\mathcal{F}_i \cup \mathcal{F}_j = \mathcal{F}_k$, and all variables in \mathcal{F}_k are jointly dependent. If a group of elements exists for which this holds, only the largest element \mathcal{F}_k is preserved, since this element is considered to model the joint dependencies sufficiently. An example of a pruned linkage tree FOS is visualized in Figure 1d.

Algorithm 1 RV-GOMEA

```

1: procedure RV-GOMEA( $f, n, \tau$ )
2:    $\mathcal{P} \leftarrow \text{InitializeAndEvaluatePopulation}(f, n)$ 
3:    $\mathcal{F} \leftarrow \text{InitializeLinkageModel}(f)$ 
4:   while  $\neg \text{TerminationCriterionSatisfied}()$  do
5:      $\mathcal{P}_0 \leftarrow \mathbf{x}^{\text{elitist}}$ 
6:     for  $\mathcal{F}_i \in \mathcal{F}$  do ▷ In random order
7:        $\mathcal{S} \leftarrow \lfloor \tau n \rfloor$  best in  $\mathcal{P}$ 
8:        $P(\{X_u : u \in \mathcal{F}_i\}) \leftarrow \text{MaxLikelihoodEstimate}(\mathcal{S})$ 
9:       for  $\mathbf{x} \in \mathcal{P}_{1\dots n-1}$  do
10:        GenepoolOptimalMixing( $f, \mathbf{x}, \mathcal{F}_i$ )
11:        AdaptiveVarianceScaling( $\mathcal{F}_i$ )
12:       for  $\mathbf{x} \in \mathcal{P}_{1\dots n}^{\text{AMS}}$  do
13:        AnticipatedMeanShift( $\mathbf{x}$ )
14:       for  $\mathbf{x} \in \mathcal{P}_{1\dots n-1}$  do
15:         if  $\text{NIS}(\mathbf{x}) > \text{NIS}^{\text{MAX}}$  then
16:           ForcedImprovement( $\mathbf{x}$ )

```

Algorithm 2 Gene-pool Optimal Mixing

```

1: procedure GENEPOOLOPTIMALMIXING( $f, \mathbf{x}, \mathcal{F}_i$ )
2:    $\mathbf{a} \leftarrow \mathbf{x}_{\mathcal{F}_i}$  ▷ Retain original
3:    $\mathbf{x}_{\mathcal{F}_i} \leftarrow P(\{X_u : u \in \mathcal{F}_i\})$ 
4:    $f'_x \leftarrow \text{Evaluate}(\mathbf{x}, f_x, \mathcal{F}_i)$  ▷ Partially evaluate, if possible
5:   if  $f'_x < f_x$  or  $\mathcal{U} < p^{\text{accept}}$  then  $f_x \leftarrow f'_x$ 
6:   else  $\mathbf{x}_{\mathcal{F}_i} \leftarrow \mathbf{a}$ 

```

3.2 Gene-pool Optimal Mixing

The variation operator of RV-GOMEA is a mechanism called Gene-pool Optimal Mixing (GOM). In GOM, variation is applied to solutions based on the dependencies encoded in the linkage model \mathcal{F} . GOM is applied at each generation, to each solution in the population, and for each FOS element \mathcal{F}_i . At each application of GOM, new values are sampled for the variables contained in \mathcal{F}_i , inserted into a parent solution, and evaluated (by partial evaluation). If this

modification improves the parent solution's objective value, it is accepted. Otherwise, it may still be accepted with probability p^{accept} , or else be rejected, in which case the parent's genotype is restored.

The GOM operator uses a sampling model to sample new variables. This work uses both the AMaLgAM sampling model [6] and the conditional RV-GOMEA sampling model [11] (described in Section 3.3). Previous work has shown that also models from other EAs, such as CMA-ES [17], can be integrated in GOM [9].

The AMaLgAM-based sampling model is based on a Gaussian distribution that is estimated using maximum likelihood based on selected solutions \mathcal{S} and is adapted to prevent premature convergence as well as better alignment with the direction of largest improvement in the landscape. These adaptation details are explained in further detail in [12]. Here, we include pseudocode of RV-GOMEA, in Algorithm 1, and the GOM operator in particular, in Algorithm 2. This definition of RV-GOMEA follows the definition in [11] by performing GOM at each FOS element iteration, as opposed to each generation [12].

3.3 Conditional Gene-pool Optimal Mixing

The above described marginal product and linkage tree FOS models have shown to lead to good performance of RV-GOMEA on problems without overlapping dependencies [12]. However, many real-world GBO problems have overlapping dependency structures. Previous work has indicated that problems with increasing dependency strength are increasingly difficult to optimize with such linkage models [11], as local GOM steps on certain structures can break linkage with other, overlapping structures. Directly modeling the network of dependencies between variables may be key to more efficiently optimizing such overlapping structures.

3.3.1 Bayesian and Markov network models. Dependency networks such as Bayesian networks have been included in Estimation of Distribution Algorithms (EDAs) before, for both discrete [15, 21, 22] and real-valued problems [1, 7]. Bayesian network-based EDAs such as the Bayesian Optimization Algorithm (BOA) [21, 22] learn a network from a selected set of solutions, using a quality metric such as Bayesian Information Criterion (BIC) score. Similarly, Markov networks have also been used in EDAs [27]. Examples for discrete optimization include the Distribution Estimation Using Markov networks (DEUM) algorithm [25], the Markovianity-based Optimization Algorithm (MOA) [26], and the Markovian Learning Estimation of Distribution Algorithm (MARLEDA) [4]. The Gaussian Markov Random Field EDA (GMRF-EDA) [18] is an example of a real-valued EDA using a factorized Gaussian Markov network. However, it is focused on modeling disjoint dependency structures.

In RV-GOMEA, a conditional linkage model has been recently introduced which derives a Gaussian Markov Field (GMF) from the VIG of a problem and estimates a conditional distribution based on a GMF traversal and factorization. This strategy has proven to be effective when optimizing problems with overlapping dependency structures [11]. Multiple variants of this model exist, featuring different factorizations of the GMF and being applied at different levels of GOM optimization. These variants are described below.

3.3.2 GMF factorization methods. Two factorization methods have been proposed within RV-GOMEA. Both conduct a breadth-first

traversal of the GMF to obtain a factorized conditional distribution. The first, *UCond*, effectively translates the GMF directly to a Bayesian network, modelling ℓ factors that represent a conditional density function of one variable, given a set of parent variables it depends on. Dependent variables encountered during traversal are recorded for later conditional sampling, in the direction that traversal occurred. The second, *MCond*, conducts a clique search during traversal, merging groups of unvisited variables which are all connected by edges in the VIG and all depend on the same set of dependent variables into clique elements [11]. This clique search effectively finds a marginal product partitioning of the VIG, with each found clique being modelled jointly afterward. These cliques are often not maximal, since their members need to be fully connected and conditioned on the same dependent variables. Of note: the conditional distributions defined by the MCond and UCond factorizations are probabilistically identical, i.e., they encode the same conditional distribution over all variables, but different FOS elements are identified for GOM.

An example MCond factorization is visualized in Figure 2a, next to the derived factorized conditional distribution, in Figure 2b. This factorization encodes the following probability distribution:

$$\begin{aligned} P(\mathbf{X}) &= P\left(\mathbf{X}_{\{0,1,2,4\}}\right) * P\left(\mathbf{X}_{\{3\}} \mid \mathbf{X}_{\{0,1,4\}}\right) \\ &\quad * P\left(\mathbf{X}_{\{5\}} \mid \mathbf{X}_{\{1,2,3,4\}}\right) * P\left(\mathbf{X}_{\{6\}} \mid \mathbf{X}_{\{0,3,4,7\}}\right) \\ &\quad * P\left(\mathbf{X}_{\{7\}} \mid \mathbf{X}_{\{1,3,4,5\}}\right) * P\left(\mathbf{X}_{\{8\}} \mid \mathbf{X}_{\{2,4,5,6,7\}}\right) \end{aligned}$$

3.3.3 GOM optimization levels. The derived factorized conditional distribution can be leveraged at different levels. At the Factorized GOM (FG) level, both conditional FOS elements and non-conditional FOS elements, pertaining to respectively conditional dependencies and joint dependencies are identified. In the factorized distribution above, each factor corresponds to an FOS element. This results in ℓ conditional FOS elements in the UCond factorization, and at most ℓ elements in the MCond factorization. GOM is applied to each FOS element separately. For a non-conditional FOS element, GOM is equivalent to standard GOM as outlined in Section 3.2. For a conditional element (i.e., conditional GOM), GOM is essentially the same, but samples from the conditional distribution where the values for the variables being conditioned on are taken from the solution that is undergoing GOM.

At the Generational GOM (GG) level, the entire factorized conditional distribution, consisting of all variables, is modelled as one conditional FOS element, and sampled in one forward sampling step. Although this requires a full solution evaluation, this step can repair linkages that GOM steps on smaller FOS elements may have broken, because each GOM step ends with a separate selection step. For this reason, a Hybrid GOM (HG) mode has been proposed, which combines both levels. This performed best on various optimization problems [11].

In previous work, the order in which FOS elements of the FG and GG levels are optimized within one generation of HG was random. In this work, we have adapted this order to always perform GOM at the GG level first, followed by GOM on all smaller FG elements, in random order. This is based on preliminary experiments, showing a slight improvement in performance across the problems considered in this paper if the full GG element is used first.

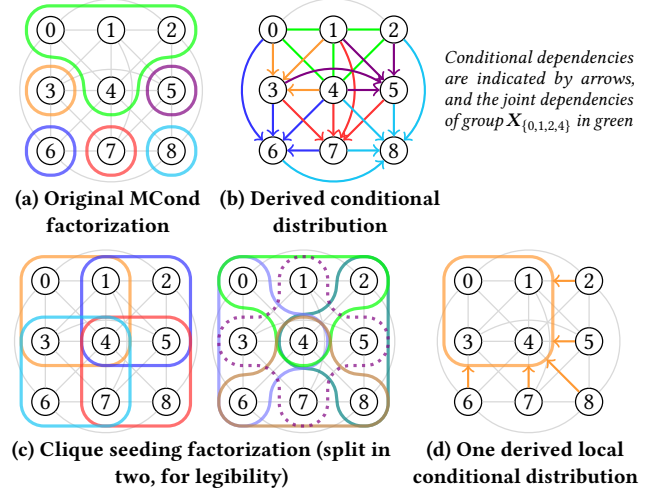


Figure 2: VIG factorizations and derived conditional distributions, on the *REBGrid* problem ($\ell = 9$). Shown are the original MCond partitioning technique (starting from randomly selected vertex 8), in (a) and (b), and the proposed clique seeding technique, in (c) and (d).

4 LEARNING THE PROBLEM STRUCTURE

In some real-world problems, the VIG of a problem may not be known. Even if it is, not all direct dependencies encoded by a generic sub-function definition \mathbb{I} may actually be of sufficient importance during optimization, making it desirable to learn the VIG during optimization. In this section, we describe how variable dependencies can be learned within RV-GOMEA, and can then be used to learn a VIG and an FOS linkage model. Finally, a new linkage model is proposed to better reflect overlapping dependency cliques.

4.1 Learning Variable Dependencies

To construct a VIG during optimization, information about direct dependencies between variables is needed. Previous work has shown that this information can be collected in a Dependency Strength Matrix (DSM) of size $\ell \times \ell$, using fitness dependency strength tests between variables [20]. A DSM is constructed by taking each possible pair of variables $i, j \in \mathcal{I}$ and testing the strength of their dependency. A dependency is present when $|\Delta_i - \Delta_{i,j}| \geq \epsilon$ for some small ϵ , with Δ_i and $\Delta_{i,j}$ being defined as:

$$\Delta_i = (f(\mathbf{x}) | x_i = a_i, x_j = a_j) \quad (2)$$

$$- (f(\mathbf{x}) | x_i = a_i + b_i, x_j = a_j),$$

$$\Delta_{i,j} = (f(\mathbf{x}) | x_i = a_i, x_j = a_j + b_j) \quad (3)$$

$$- (f(\mathbf{x}) | x_i = a_i + b_i, x_j = a_j + b_j),$$

where a_i and b_i are chosen randomly such that x_i and x_j stay within function bounds. The dependency strength is then defined as:

$$d_{i,j} = \begin{cases} 1 - \frac{\Delta_{i,j}}{\Delta_i}, & \text{if } \Delta_i \geq \Delta_{i,j} \\ 1 - \frac{\Delta_i}{\Delta_{i,j}}, & \text{otherwise.} \end{cases} \quad (4)$$

Although the strengths in a DSM are not subject to an absolute order, relative differences between $d_{i,j}$ and $d_{i,k}$ can be compared. Dependencies with a strength below a cut-off value of $\eta = 10^{-6}$ are considered negligible and assigned a strength of 0.

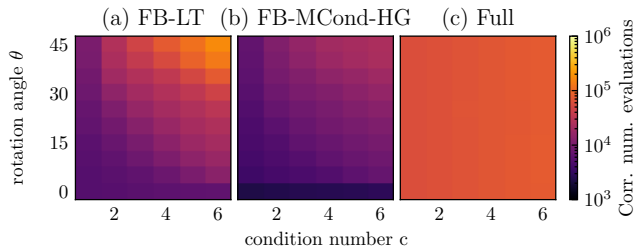


Figure 3: Required corrected number of evaluations for different linkage models on $f^{REB}(x, c, \theta, k = 2, s = 1)$ with $\ell = 20$, as determined by bisection. The median of 5 repeats is shown.

Constructing a complete DSM requires $(\ell * (\ell - 1))/2$ fitness dependency tests. In the absence of partial evaluations, this requires a quadratic overhead of function evaluations to perform, as each test consists of four evaluations. Partial evaluations can make it possible to decrease this overhead to a linear order [20], depending on the size of sub-functions. These fitness dependency test evaluations could all be performed at the start, but as many real-world GBO problems can be assumed to have sparse DSMs, it is desirable to spread this computational load across time, thereby biasing this method slightly towards problems which are not fully dependent. Moreover, dependency strengths may change during optimization, further motivating an incremental dependency learning method. Such a method has been proven effective at discovering DSMs within RV-GOMEA during optimization [20]. Within this method, a scheduling mechanism is used to limit the number of simultaneous checks and to pause temporarily if no dependencies are found. An example of a DSM constructed from fitness dependency tests, using this method in RV-GOMEA, can be seen in Figure 1a.

In this work, we remove one of the pause conditions of the scheduling mechanism. In the previously proposed method, if no dependencies are found in one test iteration (consisting of ℓ dependency tests), an exponentially sized dependency searching pause is added to the schedule. Preliminary experiments have indicated that this can lead to premature pauses on problems with very sparse DSMs and weak dependencies, as an unfortunate random draw of tested variable pairs can result in zero dependencies being found and the discovery being paused. Removing this condition prevents these premature pauses while not significantly increasing the learning overhead for problems without dependencies. The other pause condition, which tests overall dependency discovery performance across a longer interval, is kept in place.

4.2 Learning a VIG from a DSM

We assume that each non-zero dependency strength in the learned DSM represents a direct dependency. The direct dependencies derived in this way encode a VIG, which can then be used to build conditional models such as the MCond-HG model described in Section 3.3. In this work, we rebuild the VIG in every generation in which the DSM has changed, meaning that the VIG improves in quality along with the DSM, as the optimization progresses.

A preliminary experiment is performed to explore the effectiveness of this method on problems with variable dependencies of increasing strength. We consider the Rotated Ellipsoid Block (REB) problem function, which is defined in Equation 8. This problem is

defined using blocks of the well-known rotated ellipsoid problem, in which every variable is a dependent on every other variable. These sub-functions then may overlap, depending on the parameters of the problem. As the rotation angles and condition numbers of the ellipsoidal sub-functions increase, the strength of dependencies between variables, and thereby the problem difficulty, increases. In Figure 3, we show the required number of evaluations to solve this problem to a value of 10^{-10} with $\ell = 20$, using a bisection process described in Section 5.2. The figure compares a fitness-based linkage tree FOS model [20], a fitness-based conditional MCond-HG model (this work), and a full FOS model as baseline. Both the linkage tree and conditional model can successfully be constructed from an incrementally constructed DSM, but it is clear that the conditional model scales far better than the linkage tree model. The full model requires a stable number of required evaluations, regardless of the dependency strengths. However, this model is known not to scale well to higher dimensions, as well as not to exploit the GBO nature of the problem at all.

4.3 Detecting Overlapping Structures

The MCond-HG linkage model is constructed using a breadth-first traversal partitioning strategy. If conditional dependencies are disregarded, this process produces a (non-overlapping) marginal product FOS model, even if there are overlapping cliques of jointly dependent variables. The disjoint nature of this model is needed for a valid forward sampling process of the factorized conditional distribution, but leads to a number of overlapping cliques not being directly modelled. In GBO problems with sparse DSMs, it may be beneficial to model these overlapping cliques, independently, to treat all clique sub-structures, equally. Of note: this is the case because the GOM variation operator is used that tests for an improvement after each sampling step, which can therefore break clique sub-structures not modelled as such. At the GG optimization level, the MCond and UCond factorizations are probabilistically equivalent.

We propose an amended version of the MCond-HG model by replacing the FG step with a potentially overlapping clique FOS model. This model is built using a technique we call *clique seeding*. This technique starts with a set of breadth-first clique searches, with independent traversals started from each variable. For each of these searches, the maximal candidate clique C'_i surrounding the start-variable is kept. From all candidate cliques, any cliques C'_i are removed for which $C'_i = C'_j$ holds for another candidate clique C'_j . The final set of cliques is used as the conditional FOS model for the FG step, with the parent variables of the clique being all variables outside the clique that any variable inside the clique has a dependency with. Each maximal clique with its dependencies effectively forms a local conditional probability distribution. Although this can result in more dependencies being modelled than strictly necessary (there may be variables on which only some, but not all clique members depend on), this technique ensures that all maximal cliques are modelled. This effectively increases the probability that all variables pertaining to a single sub-function are jointly sampled, which may be very important to achieve improved efficiency in case of sub-functions with strongly dependent variables.

In Figure 2c, we show an example of the maximal cliques found by the new clique seeding technique, and compare this to the disjoint cliques found by the existing partitioning technique (see Figure 2a).

Table 1: Overview of compared linkage models. Models marked with an asterisk (*) are proposed in this work.

Linkage model	Description
Univariate	Each variable modeled independently
Static-UCond-HG	Pre-defined conditional univariate FOS with a full forward sampling step [11]
Static-MCond-HG	Pre-defined conditional marginal product FOS with a full forward sampling step [11]
Static-MCond-HG-CS (*)	Static-MCond-HG with a clique-seeded FOS
FB-LT	Fitness-based bounded, pruned linkage tree [20]
FB-UCond-HG (*)	Fitness-based conditional univariate FOS with a full forward sampling step
FB-MCond-HG (*)	Fitness-based conditional marginal product FOS with a full forward sampling step
FB-MCond-HG-CS (*)	FB-MCond-HG with a clique-seeded FOS

The found maximal cliques are:

$$\mathbf{X}_{\{0,1,3,4\}}, \mathbf{X}_{\{1,2,4,5\}}, \mathbf{X}_{\{3,4,6,7\}}, \mathbf{X}_{\{4,5,7,8\}}, \\ \mathbf{X}_{\{0,1,2,4\}}, \mathbf{X}_{\{2,4,5,8\}}, \mathbf{X}_{\{4,6,7,8\}}, \mathbf{X}_{\{0,3,4,6\}}, \mathbf{X}_{\{1,3,4,5,7\}}.$$

The derived local conditional distribution of one maximal clique is shown in Figure 2d. In RV-GOMEA, we use the new clique seeding method alongside the existing partitioning technique. The GG step, which samples from the factorized conditional distribution, still uses the original MCond factorization, as this is guaranteed to be consistent across all variables. This global sampling step can serve to restore lost linkage where overlaps exist.

5 EXPERIMENTS

In this section, we evaluate and compare the performance of RV-GOMEA with the proposed learned conditional linkage models to RV-GOMEA with previously proposed models. Table 1 lists all linkage models, which include generic models, static (pre-defined) models, and fitness-based (learned) models. We use the UCond-HG and MCond-HG variants as representatives of static conditional linkage models [11] and the pruned linkage tree as a representative of fitness-based non-conditional models [20].

We also compare the performance of RV-GOMEA to Vkd-CMA [2, 3], which is a state-of-the-art optimization method for continuous optimization problems. Vkd-CMA is also suited for problems with a sparse DSM, but cannot leverage partial problem evaluations due to its design.

5.1 Benchmark Problems

We conduct this comparison on a set of benchmark problems that allow for partial evaluations. Both separable and non-separable problems are included, to verify if the fitness dependency detection mechanism included in several linkage models is robust to both the absence and presence of separable sub-structures.

As a decomposable baseline problem, we consider the well-known *Sphere* function. This problem is defined as:

$$f^{Sphere}(\mathbf{x}) = \sum_{i=0}^{\ell-1} [x_i^2]. \quad (5)$$

The second problem we consider is the *Rosenbrock* function:

$$f^{Rosenbrock}(\mathbf{x}) = \sum_{i=0}^{\ell-2} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]. \quad (6)$$

The remaining problems are all derived from rotated ellipsoid block (REB) functions, defined as:

$$f^{Ellipsoid}(\mathbf{x}, c) = \sum_{i=0}^{|\mathbf{x}|-1} \left[10^{c*i/(|\mathbf{x}|-1)} x_i^2 \right], \quad (7)$$

$$f^{REB}(\mathbf{x}, c, \theta, k, s) = \sum_{i=0}^{\lceil \frac{|\mathbf{x}|-k}{s} \rceil - 1} f^{Ellipsoid}(R^\theta(\mathbf{x}_{[is:is+k-1]}), c), \quad (8)$$

where c defines the condition number of the REBs, θ the rotation angle with which they are rotated, k their size, and s the stride $1 \leq s$ with which they are spaced. The rotation function $R^\theta(\mathbf{x})$ rotates \mathbf{x} by θ degrees around the origin, counter-clockwise.

REB functions provide us with fine-grained control over the degree of overlap and difficulty of optimization problems. Large condition numbers c and rotation angles θ induce strong dependencies between variables grouped together by blocks.

We derive the following problem functions from f^{REB} :

$$f^{REB2Weak}(\mathbf{x}) = f^{REB}(\mathbf{x}, c = 1, \theta = 5, k = 2, s = 1), \quad (9)$$

$$f^{REB2Strong}(\mathbf{x}) = f^{REB}(\mathbf{x}, c = 6, \theta = 45, k = 2, s = 1), \quad (10)$$

$$f^{REB5NoOverlap}(\mathbf{x}) = f^{REB}(\mathbf{x}, c = 6, \theta = 45, k = 5, s = 5), \quad (11)$$

$$f^{REB5SmallOverlap}(\mathbf{x}) = f^{REB}(\mathbf{x}, c = 6, \theta = 45, k = 5, s = 4), \quad (12)$$

$$f^{REB5LargeOverlap}(\mathbf{x}) = f^{REB}(\mathbf{x}, c = 6, \theta = 45, k = 5, s = 1). \quad (13)$$

The *OSoREB* problem, which previous work has optimized using the FB-LT linkage model [20], is also included:

$$f^{OSoREB}(\mathbf{x}) = f^{REB}(\mathbf{x}, c = 6, \theta = 45, k = 5, s = 4) \\ + f^{REB}(\mathbf{x}_{[4:|\mathbf{x}|-1]}, c = 6, \theta = 45, k = 2, s = 5). \quad (14)$$

To test heterogeneous block strengths, we also define:

$$f^{REB2Alternating}(\mathbf{x}) = f^{REB}(\mathbf{x}, c_i, \theta_i, k = 2, s = 1), \quad (15)$$

$$f^{REB5Alternating}(\mathbf{x}) = f^{REB}(\mathbf{x}, c_i, \theta_i, k = 5, s = 4), \quad (16)$$

with $c_i = 1, \theta_i = 5$ if i is even, otherwise $c_i = 6, \theta_i = 45$,

$$f^{REB5DisjointPairs}(\mathbf{x}) = f^{REB}(\mathbf{x}, c = 6, \theta = 45, k = 5, s_i), \quad (17)$$

with $s_i = 4$ if i is even, otherwise $s_i = 5$.

Finally, we consider a more tightly connected REB function, where variables are arranged as vertices $v \in V$ in a square grid of size $\sqrt{\ell} \times \sqrt{\ell}$. Horizontally and vertically neighboring vertices ($a \in N(b)$) are connected by an edge. This problem is defined as:

$$f^{REBGrid}(\mathbf{x}) = \sum_{v \in V} f^{Ellipsoid}(R^{45}(\mathbf{x}_{\{v\} \cup N(v)}), 6). \quad (18)$$

5.2 Experimental Setup

On all problems, the optimal solution is the origin. An initialization range of $[-115, -110]$ is used, as this range does not bracket the optimum. We use an evaluation budget of 10^7 full evaluations and a computation time budget of 3 hours. Whenever either of these budgets is exhausted without finding an individual with an objective

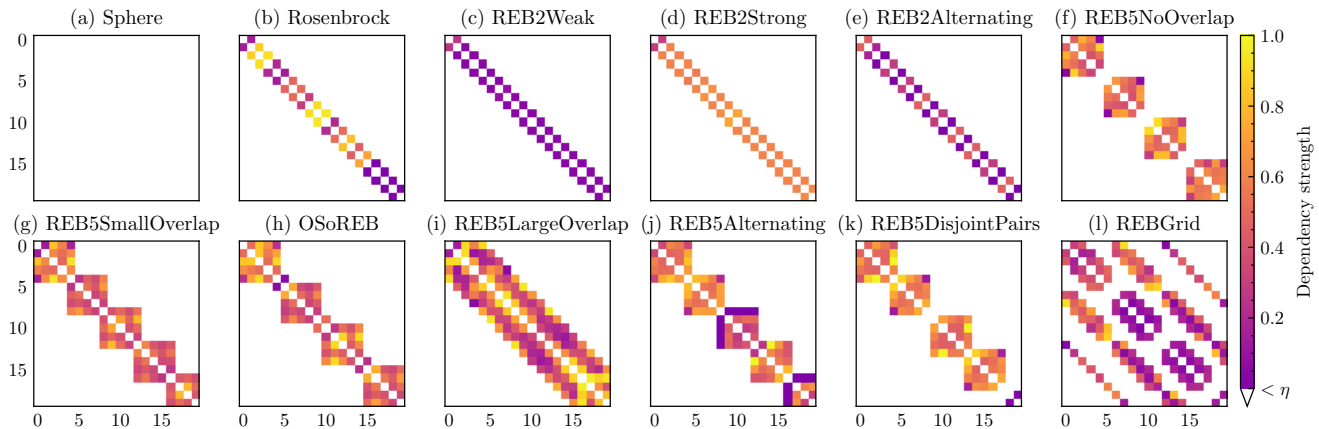


Figure 4: Heatmaps depicting a part ($\bar{I}_{[0:19]} \times \bar{I}_{[0:19]}$) of the computed DSM for each tested problem at a dimensionality $\ell > 30$. Variable pairs which are deemed independent are depicted in white.

value smaller than or equal to the value-to-reach (10^{-10}), the run is considered failed. If premature convergence occurs within either of these budgets, the population is restarted with the same size. In the interest of reproducibility, all non-deterministic factors are seeded. Source code of the modified RV-GOMEA version, implemented in C++, is provided on GitHub¹.

For each experimental configuration (consisting of a problem, a linkage model, and a dimensionality) we conduct a bisection search for the population size that yields the lowest expected number of evaluations. During this search, each tested population size is repeated 30 times. As some of these repeats may fail, we use as a metric the mean required number of evaluations of all successful repeats, divided by the fraction of repeats that are successful. We call this metric the *corrected number of evaluations*. The bisection search is limited to population sizes between 8 and 2048, and starts at a population size of $17 + 3 * \ell^{1.5}$, following a previously established guideline for the full FOS linkage model [6]. The size is then decreased exponentially while the required number of evaluations also monotonically decreases, to form a bracket for bisection. We repeat this bisection 5 times and report the median outcome.

We perform scalability experiments by testing four dimensionalities: 10, 20, 40, and 80. On problems with additional dimensionality constraints (e.g., *REBGrid*), we choose a close compatible dimensionality. As bisection on large-dimensional problems can require large amounts of computation time, the population sizes for two larger dimensionalities (160 and 320) are estimated after all bisections have completed. We use linear extrapolation on a log-log scale to estimate the (non-decreasing) slope of required population sizes. Using this slope, we estimate population sizes for the two larger dimensionalities and execute 5 sets of 30 repeats for each.

6 RESULTS

We visualize the DSMs of all problems in Figure 4, as computed during optimization. As fitness dependency accuracy can deteriorate near convergence, the shown DSMs are automatically selected from a generation before convergence. All shown dependencies are consistently detected across 30 runs, albeit with different relative strengths. While we only show DSMs of one run here, the DSMs

¹<https://github.com/gandreadis/conditional-rv-gomea>

of another run and the average over 30 runs are included in the supplementary material. We conclude from visual inspection that the problem structure is successfully learned during optimization.

In Figure 5, we show the scalability of the corrected mean number of evaluations needed by Vkd-CMA and the compared linkage models in RV-GOMEA. The corresponding population sizes, and the outcomes of statistical significance tests, are included in the supplementary material. We find that the fitness-based conditional linkage models scale similarly to their static counterparts, with only little overhead and with resilience against the absence of dependencies. As expected, on separable problems such as *REB5NoOverlap*, the MCond-HG models and their clique-seeding variants perform equally well, as their models are equivalent. On multiple problems with overlapping structures and heterogeneous dependency strengths, such as *REB5Alternating* and *REB2Alternating*, the proposed static clique seeding model scales better than existing static models ($p < 0.01$). On problems with less distinct structures, such as *REBGrid* and *REB5LargeOverlap*, the different conditional models scale similarly. Across the large majority of problems, conditional linkage models outperform Vkd-CMA at scale.

7 DISCUSSION

The results of this scalability analysis clearly show it is possible to learn the structure of a GBO problem while optimizing it, with little overhead. Moreover, RV-GOMEA with the newly introduced clique seeding technique can scale better on several problems than RV-GOMEA with the previously introduced conditional linkage models. However, there does not appear to be one clearly superior linkage model for all different dependency structures. This raises the question of how to choose the appropriate linkage model for a particular problem, especially if its structure is not known a priori.

One characteristic which may inform the choice of linkage model could be the relative strengths of connected dependency structures. Clear differences can be observed on the three *REB2* problems, which differ only in their dependency strengths: On *REB2Weak* and *REB2Strong*, both featuring homogeneous dependency strengths, scalability is similar for most linkage models, while on *REB2Alternating*, where strong and weak blocks are interleaved, clique seeding models outperform the other

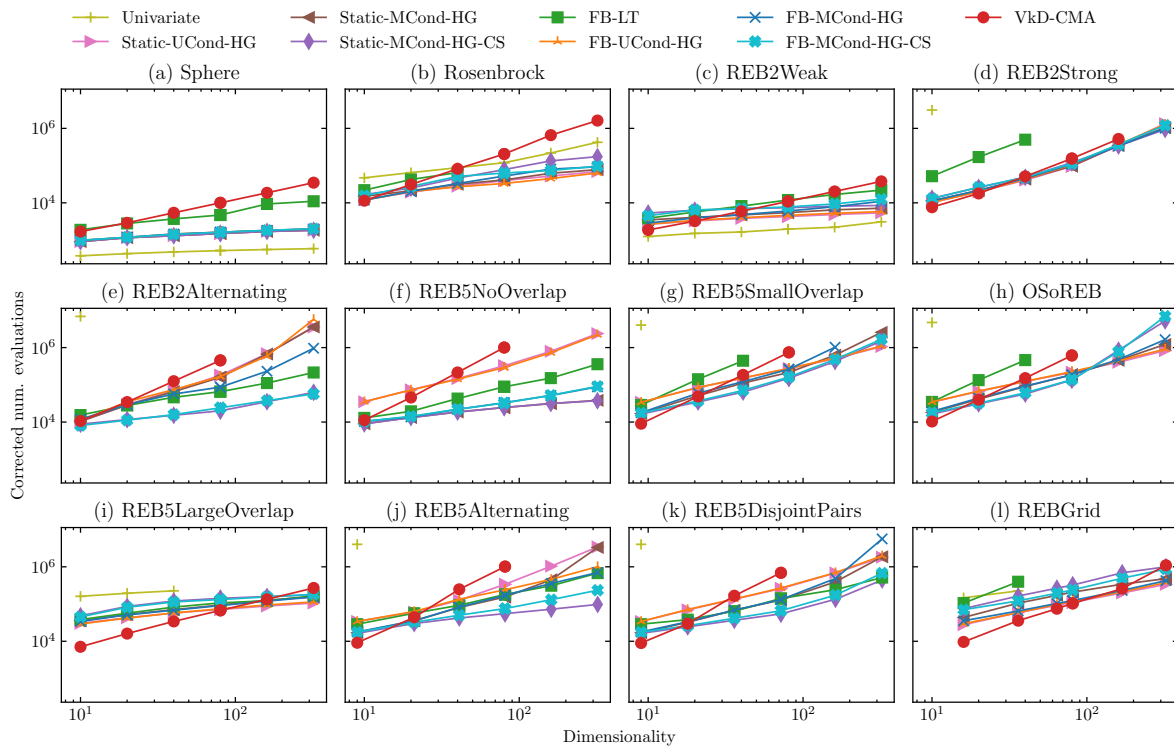


Figure 5: Results of scalability experiments for the tested benchmark problems and optimization approaches at different dimensionalities, as determined by bisection. The mean over 5 repeated bisections is shown.

models. Similar observations can be made on *REB5Alternating* and *REB5SmallOverlap*. These differences could be explained by the differing FOS models: whereas an MCond factorization may capture only some strong blocks directly as FOS elements due to unfortunate partitioning, the clique seeding technique models all blocks separately. In future work, the distinctness of dependency structures, i.e., the heterogeneity of their strengths, could therefore be an indicator for automatic linkage model selection. Furthermore, dependency strength could be modelled in a probabilistic fashion, e.g., with a dynamic cut-off dependency strength value.

The experiments in this work are limited to artificial benchmark problems. Although these allow for fine-grained control over dependency structures, many real-world problems feature less homogeneous structures. One example is the medical deformable image registration problem [5], which RV-GOMEA has shown to be capable of efficiently optimizing. Currently, the FOS model used in that work is a generic, non-conditional model based on locality as a heuristic for dependency strength. The fitness dependency learning and conditional modelling techniques presented here therefore would likely ensure a more tailored and efficient linkage model.

Finally, in this work we use an existing technique for fitness dependency learning, which is based on pairwise variable checks. A promising direction of future research could be the incorporation of more efficient, hierarchical fitness-dependency learning techniques [19], potentially further reducing the learning overhead.

8 CONCLUSIONS

In this paper, we set out to extend RV-GOMEA’s excellent performance on GBO problems with strong overlapping dependencies,

to the case where the VIG is not known beforehand. This was accomplished by applying fitness-based linkage learning techniques to construct the VIG during optimization, and deriving conditional linkage models from this VIG. Moreover, we proposed a new linkage modeling technique for overlapping structures. The efficiency of these learned conditional linkage models was compared to static linkage models and learned non-conditional linkage models in RV-GOMEA, and to Vkd-CMA, a state-of-the-art algorithm for continuous optimization. This comparison was conducted on benchmark GBO problems with varying dependency structures.

Results of this comparison showed that the learning of a conditional linkage model can be done accurately and with negligible overhead. On the majority of problems, learned conditional linkage models scaled equally well as, or better than Vkd-CMA. Furthermore, the newly introduced clique seeding technique demonstrated superior performance to existing partitioning techniques on several problems. The proposed linkage models can be applied to real-world optimization problems where the VIG is not known beforehand. This can also give valuable insights into problem structure, facilitating knowledge transfer between problem instances.

ACKNOWLEDGMENTS

This research is part of the research programme Open Technology Programme with project number 15586, which is financed by the Dutch Research Council (NWO), Elekta, and Xomnia. Further, the work is co-funded by the public-private partnership allowance for top consortia for knowledge and innovation (TKIs) from the Dutch Ministry of Economic Affairs.

REFERENCES

- [1] C. W. Ahn, R. S. Ramakrishna, and D. E. Goldberg. 2004. Real-coded Bayesian optimization algorithm: Bringing the strength of BOA into the continuous world. In *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, Vol. 3102, 840–851.
- [2] Y. Akimoto and N. Hansen. 2016. Online Model Selection for Restricted Covariance Matrix Adaptation. In *International Conference on Parallel Problem Solving from Nature*, 3–13.
- [3] Y. Akimoto and N. Hansen. 2016. Projection-based restricted covariance matrix adaptation for high dimension. In *Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, 197–204.
- [4] M. Alden and R. Miikkulainen. 2016. MARLEDA: Effective distribution estimation through Markov random fields. *Theoretical Computer Science* 633 (2016), 4–18.
- [5] G. Andreadis, P. A. N. Bosman, and T. Alderliesten. 2023. MOREA: a GPU-accelerated Evolutionary Algorithm for Multi-Objective Deformable Registration of 3D Medical Images. In *Proceedings of the 2023 Genetic and Evolutionary Computation Conference*, 1294–1302.
- [6] P. A. N. Bosman, J. Grahl, and D. Thierens. 2013. Benchmarking parameter-free AmALGaM on functions with and without noise. *Evolutionary Computation* 21, 3 (2013), 455–469.
- [7] P. A. N. Bosman. 2009. On Empirical Memory Design, Faster Selection of Bayesian Factorizations and Parameter-Free Gaussian EDAs. In *Proceedings of the 2009 Genetic and Evolutionary Computation Conference*, 389–396.
- [8] A. Bouter, T. Alderliesten, A. Bel, C. Witteveen, and Peter A. N. Bosman. 2018. Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the 2018 Genetic and Evolutionary Computation Conference*, 1199–1206.
- [9] A. Bouter, T. Alderliesten, and P. A. N. Bosman. 2020. Achieving highly scalable evolutionary real-valued optimization by exploiting partial evaluations. *Evolutionary Computation* 29, 1 (2020), 129–155.
- [10] A. Bouter, N. H. Luong, C. Witteveen, T. Alderliesten, and P. A. N. Bosman. 2017. The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, 537–544.
- [11] A. Bouter, S. C. Maree, T. Alderliesten, and P. A. N. Bosman. 2020. Leveraging conditional linkage models in gray-box optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 603–611.
- [12] A. Bouter, C. Witteveen, T. Alderliesten, and P. A. N. Bosman. 2017. Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, 705–712.
- [13] K. Deb and C. Myburgh. 2016. Breaking the Billion-Variable Barrier in real-world optimization using a customized evolutionary algorithm. In *Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, 653–660.
- [14] L. R. M. Dickhoff, E. M. Kerkhof, H. H. Deuzeman, C. L. Creutzberg, T. Alderliesten, and P. A. N. Bosman. 2022. Adaptive objective configuration in bi-objective evolutionary optimization for cervical cancer brachytherapy treatment planning. In *Proceedings of the 2022 Genetic and Evolutionary Computation Conference*, 1173–1181.
- [15] C. Echegoyen, J. A. Lozano, R. Santana, and P. Larrañaga. 2007. Exact Bayesian Network Learning in Estimation of Distribution Algorithms. In *IEEE Congress on Evolutionary Computation*, 1051–1058.
- [16] I. Gronau and S. Moran. 2007. Optimal implementations of UPGMA and other common clustering algorithms. *Inform. Process. Lett.* 104, 6 (2007), 205–210.
- [17] N. Hansen, S. D. Müller, and P. Koumoutsakos. 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation* 11, 1 (2003), 1–18.
- [18] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga. 2012. Continuous Estimation of Distribution Algorithms Based on Factorized Gaussian Markov Networks. In *Markov Networks in Evolutionary Computation*. Springer, 157–173.
- [19] M. M. Komarnicki, M. W. Przewozniczek, H. Kwasnicka, and K. Walkowiak. 2023. Incremental Recursive Ranking Grouping for Large-Scale Global Optimization. *IEEE Transactions on Evolutionary Computation* 27, 5 (2023), 1498–1513.
- [20] C. Olieman, A. Bouter, and P. A. N. Bosman. 2021. Fitness-Based Linkage Learning in the Real-Valued Gene-Pool Optimal Mixing Evolutionary Algorithm. *IEEE Transactions on Evolutionary Computation* 25, 2 (2021), 358–370.
- [21] M. Pelikan and D. E. Goldberg. 2006. Hierarchical Bayesian Optimization Algorithm. In *Studies in Computational Intelligence*, Vol. 33. Springer, 63–90.
- [22] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. 1999. BOA: The Bayesian Optimization Algorithm. In *Proceedings of the 1999 Conference on Genetic and Evolutionary Computation*, 525–532.
- [23] M. W. Przewozniczek, R. Tinós, B. Frej, and M. M. Komarnicki. 2022. On turning black - into dark gray-optimization with the direct empirical linkage discovery and partition crossover. In *Proceedings of the 2022 Genetic and Evolutionary Computation Conference*, 269–277.
- [24] O. Quevedo De Carvalho, R. Tinos, D. Whitley, and D. Sipoli Sanches. 2019. A new method for identification of recombining components in the generalized partition crossover. In *Proceedings of the 2019 Brazilian Conference on Intelligent Systems*, 36–41.
- [25] S. Shakya, A. Brownlee, J. McCall, F. Fournier, and G. Owusu. 2010. DEUM - A Fully Multivariate EDA Based on Markov Networks. In *Exploitation of Linkage Learning in Evolutionary Algorithms*, Vol. 3. Springer, 71–93.
- [26] S. Shakya and R. Santana. 2008. An EDA Based on Local Markov Property and Gibbs Sampling. In *Proceedings of the 2008 Conference on Genetic and Evolutionary Computation*, 475–476.
- [27] S. Shakya and R. Santana. 2012. A Review of Estimation of Distribution Algorithms and Markov Networks. In *Markov Networks in Evolutionary Computation*. Springer, 21–37.
- [28] D. Thierens and P. A. N. Bosman. 2011. Optimal mixing evolutionary algorithms. In *Proceedings of the 2011 Genetic and Evolutionary Computation Conference*, 617–624.
- [29] R. Tinós, D. Whitley, and F. Chicano. 2015. Partition crossover for Pseudo-Boolean optimization. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*. Association for Computing Machinery, Inc, 137–149.