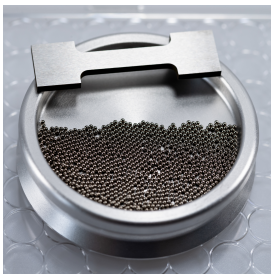
Domain-driven correlation-aware recombination and mutation operators for complex real-world applications

Christina Plump*, Bernhard J. Berger*, Rolf Drechsler*†
*Institute of Computer Science, University of Bremen, Bremen, Germany
†Cyber-Physical Systems, DFKI GmbH, Bremen, Germany
Email: {cplump, berber, drechsle}@uni-bremen.de

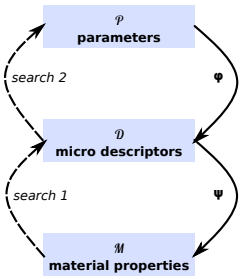
***Abstract*—Evolutionary algorithms are a very general method for optimization problems that allow adaption to many different use cases. Application to real-world problems usually comes with features as constraints, dependencies and approximations. When a multidimensional search space comes with strings attached—namely dependencies between its dimensions— an expression in two ways is possible: Restrictive—as equalities or inequalities—or vague—as correlations between dimensions, for example. Correlations between dimensions are not as easy to grasp as constraints. Therefore, well-known techniques as death penalty or penalty functions do not apply directly. We propose new mutation and recombination operators that incorporate domain knowledge to increase the offspring fraction that adheres to these correlations. We evaluate our approach with several benchmark functions and different assumptions on the dependencies of the search space. We compare the likelihood of valid (in terms of adhering correlations) outcomes of algorithms using standard mutation and recombination operators to those with the proposed operators. We find that the correlation-aware operators preserve population's features in terms of dependencies.**

I. Introduction

Technologies advance faster than ever. From crash-resistant cars to climate-neutral houses, discoveries are made every day and steadily change our lives [1], [2]. These discoveries, however, are in dire need of complex and high-performing materials. Unfortunately, the development of new structural materials with a given set of properties (e.g. tensile strength, young's modulus, hardness) is a cost-intensive and time-consuming process. Unsurprisingly, most breakthroughs in this area have been mere luck through try and error. Ellendt et al. proposed an innovative approach to constructing a guided process that significantly decreases time and cost [3]. It builds on the synthesis of material science and computer science. Material scientists develop novel testing methods (so-called micro descriptors), analogously to the established ones, but applicable on much smaller specimens [4]. Figure 1a shows an exemplary specimen for a tensile test—this is a standard test procedure for steal—and about 1300 micro samples that roughly have the same mass as the specimen. Computer scientists modify and apply machine learning techniques and optimisation algorithms to perform a guided search towards the development process necessary for a given set of material properties. Figure 1b depicts the main structure in terms of prediction and optimisation. First, two predictions need to be trained: From $\mathscr{P}$, the set of parameters producing all possible structural materials, to $\mathscr{D}$, the vector space containing the micro descriptors, and from $\mathscr{D}$ to $\mathscr{M}$, the set containing all combinations of material properties. Given a specification of material properties $m \in \mathscr{M}$, a concatenated search optimises first micro-descriptors d to match m (search 1), and finally parameters for the creation of structural materials whose corresponding micro descriptors match d (search 2). As the experiments on micro samples are faster and cheaper, the second part of the search can be supported by much more training data than the first.



(a) A specimen for a tensile test with the same mass as ∼1300 micro samples (commercial steal 100Cr6)©Jan Rathke



(b) A shortened and simplified overview about the optimisation process

Fig. 1. Information about application domain

The proposed method of Ellendt et al. then works as follows [3]: Given a set of desired material properties (potentially with error margins) $m \in \mathscr{M}$, search 1 is performed, yielding $d \in \mathscr{D}$. Then several instances of search 2 are performed, leading to several suggestions on parameter configurations for the structural material in question. The cost- and resource efficiency of the developed micro descriptors then allows a high throughput screening(see the one presented by Bader et al. [5]). Due to this high throughput screening, not only the parameter configurations found through search 2 can be tested, but also such that are *close* in either on of their dimensions. Those that fit $d \in \mathscr{D}$ best, are then chosen for review on a macro scale, leading to a significant decrease in resources and

This work was partially founded by the German Research Foundation (DFG) in subproject P02—Heuristic, Statistical and Analytical Experimental Design—of CRC1232 ('Farbige Zustände'—project number 276397488)

time needed for the discovery of a structural material with specified material properties.

There are three main areas for data scientists to come in: First, there is the classic machine learning part. Both predictive functions require machine learning techniques to be determined. With trained functions at hand, the second area comes to play: Find an input to the prediction, whose output best matches the desired target. Hence, a classical optimisation task, such that evolutionary algorithms seem like the right choice. Third, incorporating expert knowledge is a challenging but crucial task. Expert knowledge covers meta-information on training data. Additionally, it collects scientists' experience relevant to the search for new materials, e.g. whether an alloy is hazardous to health or a material behaves similarly to another. Our work deals with areas two and three.

Customising an evolutionary algorithm to a given domain and optimisation problem is almost always a challenging task. In this particular case, we encountered several intriguing issues.

The two respective search spaces are very different: On the one hand, we have $\mathscr{D} \subseteq \mathbb{R}^m$ equipped with constraints and dependencies. However, these dependencies are far from what we usually would call a correlation. They can be anything from a well reasoned, analytically defineable dependency to an experience-based gut feeling. Additionally, they do not need to be linear at all. Hence, we refrain from treating them cardinally, but rather categorically.

On the other hand, the search space $\mathscr{P}$ is a bit more complicated: An alloy follows a series of treatments (thermal and mechanical) to form a structural material. Therefore $\mathscr{P}$ is a cartesian product of $[0,1]^l$ (the percentages of used alloys), the set containing all heat treatments and the set containing all mechanical treatments. As the latter represent processes with circular subprocesses, a standard real-valued encoding is inapplicable here. Still, there are dependencies, e.g. some alloys only allow lower heating temperatures. Again, these dependencies are categorical, rather than cardinal.

For the most part, this application's constraints are derived from nature laws (e.g. a length always will be non-negative), thus unchangeable. However, as these experiments' design is still progressing, and there might be new experiments or micro-descriptors discovered, it is necessary to include this knowledge in a configurable manner. On the other hand, dependencies are pure expert knowledge, derived either from literature or experiment designing scientists. Again, as these experiments are still new research and different structural materials will be studied, knowledge about relations between micro-descriptors might change. Thus, we expect to see many changes in these dependencies.

Therefore, we derived mutation and recombination operators that include this categorical knowledge about relations between variables without being too strict about it. We ensure that dependent variables are changed according to the direction of their dependency but do not explicitly use the given value. Additionally, to ensure sustainability and the capability of adding new experiments or derived micro-descriptors, we defined a domain-specific language (DSL) and captured the necessary meta-information in DSL-compliant files.

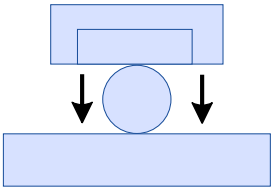
We evaluated our approach on four well-known benchmark functions comparing our newly designed operators with their standard counterparts. We compare the correlation matrix computed from the initial population, with the correlation matrix that results when adding the result to the initial population's set. We sample the initial population from different distributions. We found that our operators are more likely to produce valid, i.e. obeying the dependencies, search results than standard operators.

Incorporating correlations into recombination and mutation operators is in itself not a new idea. Several work has been done in that area, especially in evolutionary strategies (see Li et al. [6] for a survey from 2020). Great progress has been made with self-adapting algorithms that iteratively change their step sizes (and other parameters) depending on the current status of the population (see Kramer [7] as well as García-Pedrajas et al. [8]). However, our work differs in two key aspects: First, we do not only work on real-valued encodings (as is inherent to evolutionary strategies) but also on bit encodings. Second and even more importantly, we consider categorical relations, that differ twofold from standard correlations: They contain every form of relation - not only linear (as is the case for correlations) and additionally, their numeric value should not be understood cardinally as they don't adhere to a rational or even an interval scale. Furthermore, we can not allow our relations to change during one algorithm as they represent physical laws and thus must be obeyed [9]. For genetic algorithms, Kundu et al. demonstrated the effectiveness of correlation-aware selection operators [10]. They incorporated the knowledge about the correlation between the current best individual and the other indiviuals in the population into their selection process. The presented approach here, however, does not guide the selection process in itself through knowledge about relations between individuals, but the recombination and mutation phase instead. Thus, it guides the way new individuals are created. Additionally, we focus on relations between different input dimensions, that are rather categorical than cardinal.

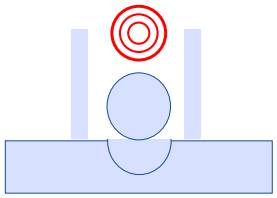
This paper's remainder is structured as follows: In Section 2, we formally describe the given optimisation problem and give an example to further illustrate the situation. Section 3 introduces the adapted operators in a reproducible manner, while Section 4 explains the practical implementation and incorporation of domain-specific knowledge. Finally, Section 5 demonstrates and discusses our evaluation's results and Section 6 concludes this article with an outlook.

II. Background and Problem Description

Drechsler et al. proposed the formal methodology to the high-throughput approach from Ellendt et al. [3] in Drechsler et al. [11], which was in part implemented by Huhn et al. [12] and Drechsler et al. [13] using a recursive least squares kernel estimation [14]. The following section defines the domain-specific terms, gives an example and specifies the optimisation problem.



(a) Schematic depiction of a micro-compression test (derived from [15])



(b) Schematic depiction of a laser shockwave indention test (derived from [16])

Fig. 2. Exemplary novel testing methods developed for a high-throughput approach

Definition 1 (Experiment): An experiment E is a procedure carried out on a testing specimen. It has parameters $p_{j,E}$ that determine the exact specification of the procedure, and descriptors $d_{i,E}$ that contain the experiment's results. Each parameter and each descriptor may have constraints of the form $g(d_{i,E}) = d$, $g(d_{i,E}) \geq d$ or even $g(d_{i,E}, d_{j,E}, ...) = d$. Furthermore, there may be bivariate relations between descriptors of the experiment or other experiments of the form $rel(d_{i,E}, d_{i,E'}) = c$, where E can be different or equal to E' and $c \in [-1, 1]$. We distinguish experiments on micro samples and such on macro specimen with the respective index. $\mathscr{E}$ is the set containing all experiments.

Let's illustrate this with a small example:

Example 1 (Micro Compression and Laser Shockwaves):

The experiment *Micro compression test* [15] presses an indenter onto a specimen and then releases the pressure. The applied force and the indention on the specimen are measured at an equidistant rate. Parameters for this experiment are pressure and specimen size. The descriptors for this experiment are derived from the force-distance-diagram and compute the indention depth and several mechanical works. Between these mechanical works, some constraints hold. The experiment *Laser Shockwave Indention* [16] pushes a shockwave onto an indenter through a laser, which then presses the indenter onto an embedded micro sample cut in half. Descriptors in this case are indention area, indention diameter, the resulting pile-up and theoretical depth from diameter. Again, there are constraints between these descriptors. Additionally, there is a negative relation between plastic work (derived from microcompression) and indention diameter(derived from laser shock wave). Two data description files capture all of this information. Listing 1 shows the data description file for laser shockwave and Listing 2 for microcompression. For brevity we do not display units here as is done in the actual data description file and leave out descriptors without constraints. One can see the singular constraints, the bivariate ones, and the relation between the abovementioned descriptors.

Now we're ready to formally define $\mathscr{D}$:

Definition 2 (Search Space of Micro Descriptors): The search space of all micro descriptors $\mathscr{D}$ consists of all available micro-descriptors, i.e.

$$\mathscr{D} = \bigoplus_{E \in \mathscr{E}_{micro}} \mathscr{D}_E \tag{1}$$

Listing 1. Data Description File for Laser Shockwave

```
experiment "laser shockwave" {
  free quotient parameter "specimen diameter"
    § "specimen diameter" >= 0;
  measuringmethod LiSe {
    quotient descriptor "indention area"
      § "indention depth" >= 0;
    quotient descriptor "indention diameter"
      § "indention diameter" >= 0;
    quotient descriptor "measured depth"
      § "measured depth" >= 0;
    § "indention area" = "indention diameter" /2 * 3.14;
  }
  /* relation between extra-experimental descriptor */
  § rel("microcompression.plastic work",
      "indention diameter") = -0.6;
}
```

Listing 2. Data Description File for Micro Compression

```
experiment "microcompression" {
  free quotient parameter "force"
    § "force" >= 0;
  free quotient parameter "specimen diameter"
    § "specimen diameter" >= 0;
  measuringmethod ForceDistance {
    quotient descriptor "indention depth"
      § "indention depth" >= 0;
    quotient descriptor "mechanical work"
      § "mechanical work" >= 0;
    quotient descriptor "plastic work"
      § "plastic work" >= 0;
    [...]
  }
  /* relation between extra-experimental descriptor */
  § rel("laser shockwave.indention diameter",
      "plastic work") = -0.6;
}
```

where $\mathscr{D}_E = \left\{ \boldsymbol{d} = (d_{1,E}, ..., d_{n_E,E})^\top \,|\boldsymbol{d} \vdash \mathscr{C}_E \right\}$, and $\mathscr{C}$ contains all constraints and relations. We basically flatten all descriptors in one vector of dimension $\sum_{E \in \mathscr{E}_{micro}} n_E$.

This search space is the origin of the second significant mapping, i.e. the predictive function:

Definition 3 (Predictive Function): The predictive function $\psi : \mathscr{D} \longrightarrow \mathscr{M}$ maps a collection of micro descriptors to their corresponding material properties. That is: If the same structural material is tested on a micro level with the newly developed testing methods and is tested with standard procedures, the predictive function maps these results to one another.

The optimisation problem (search 1) is strongly related to the predictive function and can then be defined as follows:

Definition 4 (Optimization problem): Given $\boldsymbol{m}^* \in \mathscr{M}$, find $\boldsymbol{d} \in \mathscr{D}$, s.t. $||\psi(\boldsymbol{d}) - \boldsymbol{m^*}|| \rightarrow min!$ with all constraints g from the experiments' definition satistfied and rel obeyed.

III. Methodology

We transform the given search space in the problem domain to a population by representing a $\boldsymbol{d} \in \mathscr{D}$ with a genotype of n_{total} chromosomes, i.e.

$$genotype \equiv \boldsymbol{d}$$
$$chromosome \equiv d_i$$

We adapt recombination and mutation operators for both real-valued and bit encoding. As we argued in Section 1 the search space $\mathscr{U}$ is not suitable for real-valued encoding. To

obtain operators for both searches, we carry out the adaption for both encodings.

Our main methodology follows the principle (assuming that each chromosome of a genotype is altered with a given probability):

> *If a chromosome is altered, all related chromosomes are altered according to their respective rel.*

Please keep in mind, that the relations should not be taken at face value, but rather for their general direction, i.e. a $rel(indentionDepth, plasticDeformation) = 0.8$ must not be interpreted like a correlation coefficient, but as *There is a strong, positive relation*. Especially, it is not restricted to linear dependencies.

A. Mutation operators

We adapted two standard mutation operators: One, the `GaussianMutator` as standard mutator for a real-valued encoding, and second, the `SwapMutator` as standard mutator for a bit encoding. For both instances, we reciprocate the mutation carried out on one chromosome to all dependent chromosomes, either parallely ($c \geq 0$) or inversely ($c < 0$).

1) GaussianMutator: In its standard (with fixed step size) setting, the `GaussianMutator` takes a random value $r \sim \mathcal{N}(0, \sigma)$ and adds it to the current value, cropping it to its minimal and maximal values, if necessary. Please note, that the following algorithms do not show the cropping process. The adapted `GaussianMutator` works similarly, as shown in Algorithm 1 and 2. If there is a strong correlation between two chromosomes, the random value propagates to the correlated chromosome (see line 5, Algorithm 2), respecting the sign of the correlation factor. If the correlation is a weak one, the random value adds to a new one to take it into account (see line 7, Algorithm 2)).

Algorithm 1 Correlated Gaussian Mutator

Require: $|gt^p| = n$ {genotype of length n}
Require: v {gaussian random value}
Require: $1 \leq c \leq n$ {randomly chosen chromosome}
Require: t {category threshold}
Require: σ {standard deviation}
1: $gt^r \leftarrow gt^p$
2: $propagate(i \leftarrow c, r \leftarrow v)$
3: **return** gt^r

2) SwapMutator: The `SwapMutator` works on bit encodings. It randomly chooses chromosomes from a genotype. For each chromosome, it selects a bit of that chromosome and flips it. Algorithms 3 and 4 show the correlation-aware swap mutator. It flips the initially selected bit for a chromosome by calling the propagate algorithm with the $\perp$ parameter. The propagate algorithm has five different propagation modes. First, the $\perp$-mode negates the given bit. Second, the $\downarrow$- and $\uparrow$-mode sets the bit to $false$ or $true$ (see lines 4–7, Algorithm 4), depending of the mode. This corresponds to a weak correlation where the correlated chromosomes are changed the same way or stay unchanged. The strong correlation is covered by the

Algorithm 2 Correlation Propagation of the Gaussian Mutator

Require: $1 \leq i \leq n$ {chromosome}
Require: $r \sim \mathcal{N}(0, 1)$
1: $gt^r_i \leftarrow r \cdot \sigma + gt^p_i$
2: **for** $(j, f) \in correlationsOf(i)$ **do**
3: $g \sim \mathcal{N}(0, 1)$
4: **if** $f \geq t$ **or** $f \leq -t$ **then**
5: $propagate(i \leftarrow j, r \leftarrow sign(f) \cdot r)$
6: **else**
7: $propagate(i \leftarrow j, r \leftarrow sign(f) \cdot (r + g)/2)$
8: **end if**
9: **end for**

$\downdownarrows$- and $\uparrow\uparrow$-mode. It looks for the next bit that can be flipped into the given direction (see lines 8–13, Algorithm 4). The propagation mode is chosen based on the correlation factor (f) and the change kind (see lines 17–25, Algorithm 4).

Algorithm 3 Correlated Swap Mutator

Require: $|gt^p| = n$ {genotype of length n}
Require: $1 \leq c \leq n$ {randomly chosen chromosome}
Require: $1 \leq g \leq |gt^p_c|$ {randomly chosen gene}
Require: t {category threshold}
1: $gt^r \leftarrow gt^p$
2: $propagate(i \leftarrow g, d \leftarrow \perp)$
3: **return** gt^r

B. Recombination Operator

We adapted two standard recombination operators for each encoding: For bit encodings, we adapted `SinglePointCrossover` and `UniformCrossover`. For real-valued encodings, we adapted `MeanAlterer` and `LineCrossover`, where one actually is a special case of the other. For recombination operators, we enforce the same crossover for all dependent chromosomes. In this case we do not need to explicitly encode the direction of dependence, it is inherent to the recombination.

1) MeanAlterer: The `MeanAlterer` computes the average between two chromosomes and uses this as offspring. Again, we simply copy this behaviour to all dependent chromosomes (see Algorithm 5)

2) LineCrossover: The standard procedure of an `LineCrossover` is drawing a line through both parental chromosomes and choosing a point on that line as offspring. In our case, chromosomes are one-dimensional, therefore it is basically a weighted average. Again, we remember the random value deciding the position of the point on the drawn line, and pass it to all dependent chromosomes. Algorithm 6 shows the idea in pseudo code.

3) SinglePointCrossover: A `SinglePointCrossover` takes a random number g, cuts the chromosome at this bit's position into halves, and repeats this with the other parent. Tails are switched and two offsprings are generated. The correlation-aware single point crossover works slightly different

Algorithm 4 Correlation Propagation of the Swap Mutator

Require: $1 \leq i \leq |gt_c^p|$ {gene}
Require: $d \in \{\downdownarrows, \downarrow, \perp, \uparrow, \upuparrows\}$ {propagation rule}
1: $k \leftarrow i$
2: **if** $d = \perp$ **then**
3: $new \leftarrow \neg gt_{c_i}^p$
4: **else if** $d = \downarrow$ **then**
5: $new \leftarrow false$
6: **else if** $d = \uparrow$ **then**
7: $new \leftarrow true$
8: **else if** $d = \upuparrows$ **then**
9: $k \leftarrow findNearestIndex(i, false)$
10: $new \leftarrow true$
11: **else if** $d = \downdownarrows$ **then**
12: $k \leftarrow findNearestIndex(i, true)$
13: $new \leftarrow false$
14: **end if**
15: $gt_{c_k}^p \leftarrow new$
16: **for** $(j, f) \in correlationsOf(i)$ **do**
17: **if** $f \geq t$ **and** new **or** $f \leq -t$ **and** $\neg new$ **then**
18: $propagate(i \leftarrow j, d \leftarrow \upuparrows)$
19: **else if** $f \geq t$ **and** $\neg new$ **or** $f \leq -t$ **and** new **then**
20: $propagate(i \leftarrow j, d \leftarrow \downdownarrows)$
21: **else if** $f > 0$ **and** new **or** $f > -t$ **and** $\neg new$ **then**
22: $propagate(i \leftarrow j, d \leftarrow \uparrow)$
23: **else if** $f > 0$ **and** $\neg new$ **or** $f > -t$ **and** new **then**
24: $propagate(i \leftarrow j, d \leftarrow \downarrow)$
25: **end if**
26: **end for**

Algorithm 5 CorrelatedMeanAlterer

Require: $|gt^{p_1}| = |gt^{p_2}| = n$ {equal-sized genotypes}
Require: $1 \leq c \leq |gt^{p_1}|$ {randomly chosen chromosome}
for $i = 1$ **to** $|gt^{p_1}|$ **do**
if $(i, \_) \in correlationsOf(c)$ **or** $i = c$ **then**
$gt_i^c \leftarrow mean(gt_i^{p_1}, gt_i^{p_2})$
else
$gt_i^c \leftarrow gt_i^{p_1}$
end if
end for
return gt^c, gt^{p_2}

(see Algorithm 7 and 8). Only one of the chromosomes is altered by receiving the tail of the other chromosome (compare lines 2–6, Algorithm 8). The surviving parent of correlated chromosomes is chosen depending on the correlation factor's sign. If two chromosomes are positivley correlated the same parent survives.

4) UniformCrossover: The `UniformCrossover` swaps randomly choosen genes between two parental chromosomes. We implemented the correlation-aware uniform crossover similar to the correlation-aware single point crossover. Algorithm 9 and 10 shows its working. The correlation-aware uniform crossover copies a gene according to the sign of the correlation

Algorithm 6 Correlated Line Crossover

Require: $|gt^{p_1}| = |gt^{p_2}|$ {equal-sized genotypes}
Require: $0 \leq p \leq 1$ {probability}
Require: $1 \leq c \leq |gt^{p_1}|$ {randomly chosen chromosome}
Require: $-p \leq f_1 \leq 1 + p$ {random value}
Require: $-p \leq f_2 \leq 1 + p$ {random value}
for $i = 1$ **to** $|gt^{p_1}|$ **do**
if $(i, \_) \in correlationsOf(c)$ **or** $i = c$ **then**
$gt_i^{c_1} \leftarrow f_1 \cdot gt_i^{p_1} + (1 - f_1) \cdot gt_i^{p_2}$
$gt_i^{c_2} \leftarrow f_2 \cdot gt_i^{p_2} + (1 - f_2) \cdot gt_i^{p_1}$
else
$gt_i^{c_1} \leftarrow gt_i^{p_1}$
$gt_i^{c_2} \leftarrow gt_i^{p_2}$
end if
end for
return gt^{c_1}, gt^{c_2}

Algorithm 7 Single Point Crossover

Require: $|gt^{p_1}| = |gt^{p_2}|$ {equal-sized genotypes}
Require: $1 \leq c \leq |gt^{p_1}|$ {randomly chosen chromosome}
Require: $1 \leq g \leq |gt_c^{p_1}|$ {randomly chosen gene}
1: $gt^{c_1} \leftarrow gt^{p_1}$
2: $gt^{c_2} \leftarrow gt^{p_2}$
3: $propagate(i \leftarrow c, d \leftarrow 1)$
4: **return** gt^{c_1}, gt^{c_2}

factor (see lines 2–6, Algorithm 10). As long as the correlation is positive the swap direction stays the same for correlated chromosomes (see line 9, Algorithm 10).

IV. Implementation

The implementation consists of two essential parts—first, the data-description language and second, the correlated operators. The data-description language is a domain-specific language that trained domain-experts can write and is more appealing to them than several configuration files. Domain-specific languages require a well-defined grammar, a scanner and a parser that turns a document conforming to the grammar into an intermediate representation which then can be interpreted.

Algorithm 8 Propagation of the Single Point Crossover

Require: $1 \leq i \leq |gt^{p_1}|$
Require: d
1: $s \leftarrow |gt_i^{c_1}|$ {number of genes in chromosome}
2: **if** $d \geq 0$ **then**
3: $gt_{i_{g\ldots s}}^{c_2} \leftarrow gt_{i_{g\ldots s}}^{p_1}$
4: **else**
5: $gt_{i_{g\ldots s}}^{c_1} \leftarrow gt_{i_{g\ldots s}}^{p_2}$
6: **end if**
7: **for** $(j, f) \in correlationsOf(i)$ **do**
8: $propagate(i \leftarrow j, d \leftarrow d \cdot f)$
9: **end for**

Algorithm 9 Correlated Uniform Crossover

Require: $|gt^{p_1}| = |gt^{p_2}|$ {equal-sized genotypes}
Require: $1 \leq c \leq |gt^{p_1}|$ {randomly chosen chromosome}
Require: G {set of randomly chosen genes}
Require: $1 \leq g_l \in G \leq |gt^{p_1}_c|$
1: $gt^{c_1} \leftarrow gt^{p_1}$
2: $gt^{c_2} \leftarrow gt^{p_2}$
3: $crossover(i \leftarrow c, d \leftarrow 1)$
4: **return** gt^{c_1}, gt^{c_2}

Algorithm 10 Propagation of the Correlated Uniform Crossover

Require: $1 \leq i \leq |gt^{p_1}|$
Require: $-1 \leq d \leq 1$
1: **for** $k \in g$ **do**
2: **if** $d \geq 0$ **then**
3: $gt^{c_2}_{i_k} \leftarrow gt^{p_1}_{i_k}$
4: **else**
5: $gt^{c_1}_{i_k} \leftarrow gt^{p_2}_{i_k}$
6: **end if**
7: **end for**
8: **for** $(j, f) \in correlationsOf(i)$ **do**
9: $crossover(i \leftarrow j, d \leftarrow d \cdot f)$
10: **end for**

We implemented the data-description language using Eclipse Xtext [17].

Eclipse Xtext belongs to the Eclipse modelling eco-system which offers a wide range of tool support for designing and creating models, languages, and corresponding graphical representations. We integrated the correlation operators into Jenetics, an extensible Java-library for evolutionary algorithms [18]. Figure 3 depicts the underlying principle of our implementation. The evolutionary algorithm uses the data description file to customize its operation and potentially initial population. It then optimizes towards a given requirement profile (targets). Its result is a correlation-aware individual, i.e. an individual, which obeys the relations from the data description file. This individual produces an predicted profile, which hopefully is close to the given requirement profile.

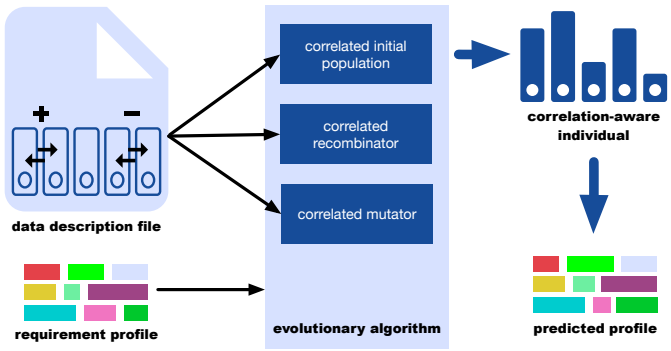


Fig. 3. Schematic overview of methodology's implementation

Algorithm 11 Standard Genetic Algorithm

$pop_0 \leftarrow pop_{initial}$
for $i = 1$ **to** $i \leq generations$ **do**
$pop^s_i \leftarrow select(pop_{i-1}, f(pop_{i-1}))$
$pop^p_i \leftarrow select(pop_{i-1}, f(pop_{i-1}))$
$pop^o_i \leftarrow mutate(recombinate(pop^p_i))$
$pop_i \leftarrow pop^s_i \cup pop^o_i$
end for

Algorithm 11 shows the standard genetic algorithm, Jenetics provides. Survivors and parents are selected based on their fitness, parents are recombinated and mutated to generate offsprings. Finally, offsprings and survivors are combined to form the next generation's population. Then, the fitness is computed for each individual, and the loop is repeated until the specified number of generations is completed. We customized the initial population selection, the recombination operator, the mutation operator, and the fitness function in our implementation. The complete implementation, including the parser for the domain-specific language, consists of 30k source lines of Java code.

V. Evaluation

We investigate the following research question:

RQ 1: The population (and thus the result) of an evolutionary algorithm are more likely to adhere to the relations specified in a data description file when using correlation-aware operators over standard operators.

We research this question on benchmark functions as well as the real-time scenario posing the inspiration for this method.

We identify several possibly influencing parameters: The target used for the fitness function (target), the initial population's features regarding dependencies (initial), the encoding, the handling of strict constraints (bounded), and the relations specified in the data description file. We also varied the configuration of the applied evolutionary algorithm and the benchmark function used for the fitness evaluation to widen the data we base our statistical decision on. We give an overview of our evaluation setup in Figure 4.

We plan to determine the relation preserving effect by comparing two covariance matrices computed on the initial and the final population. To do so, we calculate the sum of absolute distances between both matrices. The smaller this distance, the higher the relation preserving effect. We then compare the distance for each setup with correlation-preserving operators to its counterpart with standard operators, where results above 0 favor the correlated operators and vice versa. We use a Welsh's t-test [19], comparing the means over 50 runs using Satterthwaite's degrees of freedom. Following up, we compute *wins* to compare which method has a higher number of statistically significant results [20]. That is, a method gains a *win* if it significantly ($\alpha < 0.05$) outperforms the other. For setups, where none of both methods is significantly better, we assign a *win* to the category *borderline*, i.e. for this setup no method can be said to be outperforming the other.

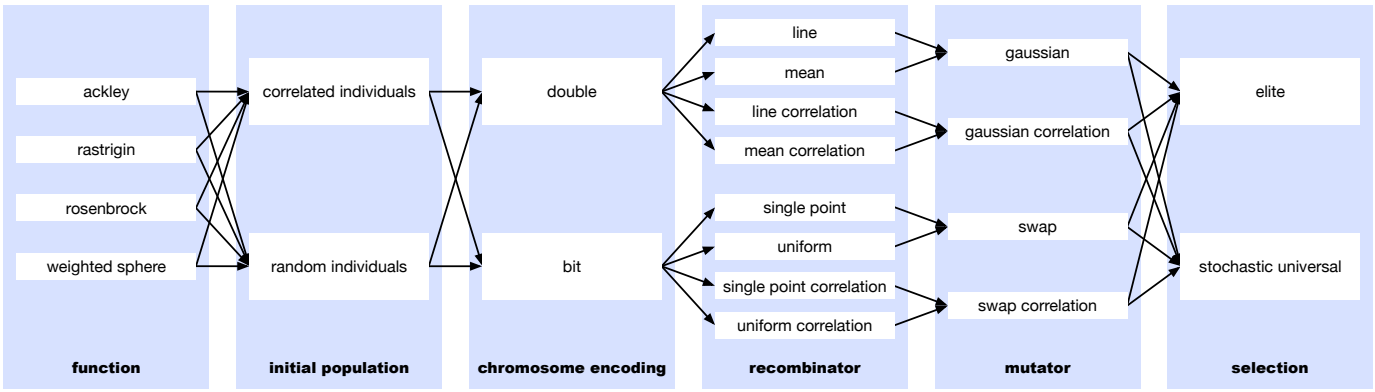


Fig. 4. Graphical representation of the evaluation variation

A. Benchmark Functions

All in all, we had four benchmark functions, three data description files, eight EA-configurations (encoding included and please compare Figure 4 for an overview of configurations) and 50 targets. Factoring in the variations on constraint handling and initial population, we obtain 19.200 distinct configurations. All of them were evaluated once with correlation-preserving operators and again with standard operators, making a total of 38.400 setups. Finally, we ran each setup 50 times, adding up to 1.920.000 evaluations.

a) Functions: We used for different benchmark functions following a classification of the properties modality and separability. These are the *WeightedSphere Function* for unimodality and separability (ranges: $[-5.12, 5.12]$), the *Rastrigin Function* for multimodality and separability (ranges: $[-5.12, 5.12]$ and $a = 10$), the *Rosenbrock Function* for unimodality and inseparability (ranges: $[-2.048, 2.048]$), and the *Ackley-Function* for multimodality and inseparability (ranges: $[-20, 30]$). We work with $n = 10$.

b) Relations and Initial Populations: We used random initial populations and pre-generated ones. The pre-generated ones follow a multivariate normal distribution. The mean is adjusted to the center of the benchmark function's range and the variances are determined to match the given range in a 3σ deviation. We chose different covariances. One, with paired correlations, varying between strong and weak, as well as negative and positive correlations (type A). Another, with a strongly correlated block for the first variables and none else (type B). Last but not least, two correlated blocks for the first and last variables, the latter being weaker correlated than the first (type C).

We mirrored these correlations in the corresponding data description files.

1) Results: For all 19.200 distinct configurations, we computed the p-value of a Welsh's t-test. Figure 5 shows the number of setups where the correlated operators win the hypothesis test (correlated), where the non-correlated operators win the hypothesis test (non-correlated), and where no significant decision can be made (borderline). Note, we have a confidence

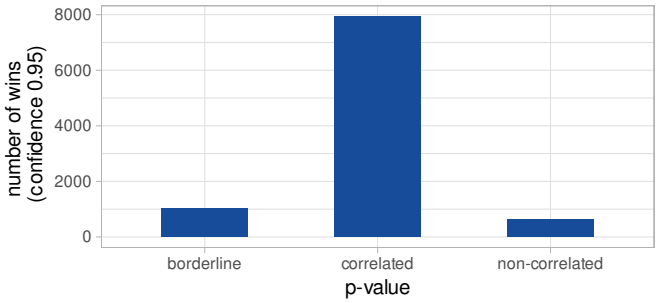


Fig. 5. Number of wins for correlated, non-correlated and borderline

level of 0.95. We see that the overwhelming majority of cases is won by correlation-aware operators.

Figure 6 depicts the wins differentiated by the relation specifications - configured through the data description file as well as the initial population (when following a multivariate normal distribution). There seems to be no notable difference, only relation configuration B produces a slight improvement for *correlated*. Next, we analyse the influence of encoding and thus, the singular performance of our correlation-aware operators. Figure 7 shows the number of wins, deviated for encoding and selection operator. Since the left bar chart does not differ much from the right bar chart, we can conclude that both selection configuration perform more or less equally. However, the bit-encoding (bit) seems to perform comparatively better than the real-valued encoding (double). Nevertheless, for both encodings the correlation-aware operators outperform the standard ones. Figure 9 depicts the number of wins depending on whether or not constraints were enforced and differentiated for their encoding. We see no difference for real-valued encoding (double), but a slight tendency towards unbounded for bit-encoding. Again, this bar chart shows the comparatively better response of bit-encoding to the correlation-aware operators. Last, but not least, we compared the factors for the inital population: In Figure 8 one can see, that reciprocating correlated input data (training) is more complicated than with random data.

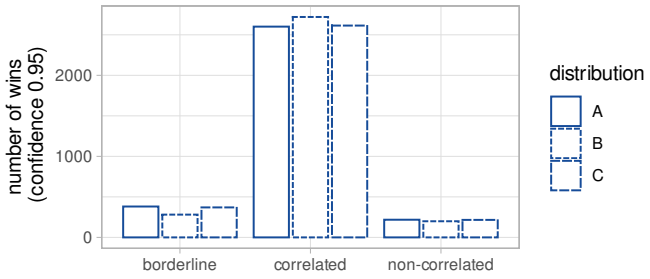


Fig. 6. Number of wins differentiated by relation specifications

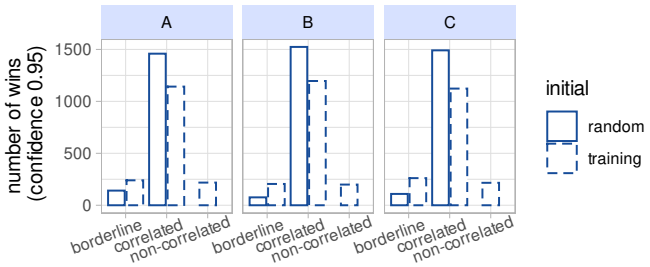


Fig. 8. Number of wins depending on initial population

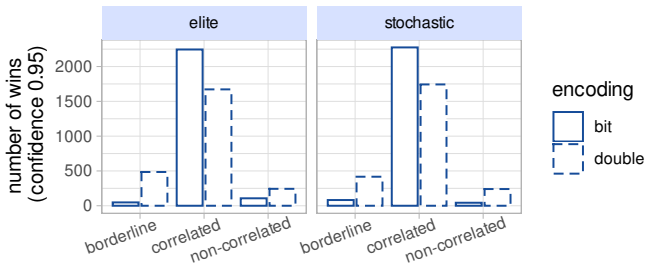


Fig. 7. Number of wins by encoding types

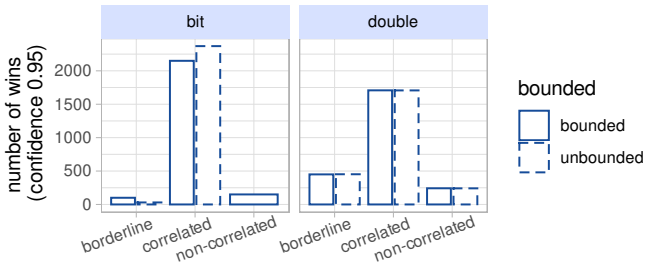


Fig. 9. Number of wins by constraint enforcement

2) Discussion and threats-to-validity: Our results show that the proposed correlation-aware operators significantly outperform standard operators in preserving specified relations of the input domain. They are independent of the given relations. Interestingly, they seem to have a higher effect when employed bit-encoding than with real-valued encoding. As we do not expect this to be an actual effect of the encoding in itself, it could be a hint, that our bit-encoding based operators work better. The chosen selection operator did not seem to make a difference. However, as we worked with a fixed parameter setting in this case, we can not definitely draw this conclusion as might see different responses with varying parameters of those selection operators. Enforcing constraints on the other hand seemed to hinder preserving the correlations. We assume, this is due to the fact, that we crop constraint-violating individuals to the closest allowed values, therefore possibly violating the relation specifictions. All in all, we could positively answer our research question and successfully investigated several influencing parameters.

B. Application Domain

We additionally evaluated the proposed operators on data from the given application from material sciences. We chose three different setups to evaluate: One to find a set of micro descriptors to yield a specified hardness (input dimension was 13), one to yield a given tensile strength (input dimension was 18) and one to yield an indention modulus (input dimension was 10). We trained the predictive function ψ (compare Figure 1b) using a kernel-based Support Vector Regression [14] on supplied training data from experiments of material scientists. We kept the above setup but only used one relation specification for each application setup, according to information supplied by material scientist responsible for the experiments. Compare Listing 1 and Listing 2 for an example of data description files covering this domain specific knowledge.

1) Results: Figure 10 shows the results for the application domain. Again, we can see that for the majority of cases, the correlation-aware operators outperform the standard ones. Please note that the chart displays the distribution of p-values for all performed setup evaluations. That is, the total count of setups yielding a p-value higher than 0.95 corresponds to the number of wins for *correlated* displayed for the benchmark functions, the total count setups yielding a p-value lower than $1 - 0.95 = 0.05$ corresponds to the number of wins for *non-correlated* and all the ones in between correspond the number of wins for *borderline*. We cross-checked the returned individuals with material scientist to review whether this might be valid combinations of micro-descriptors and this was positively answered.

2) Discussion and threats-to-validity: Our evalution showed for the application domain, that the correlation-aware operators perform better than standard ones, however, not as good as for the benchmark cases. This can be seen in the higher number of wins for *borderline*, i.e setups where neither the standard method nor our approach was significantly better than the other one. We assume this is due to three reasons: First, we work with an approximative fitness function that is not yet trained to incorporate relations of input dimensions. Therefore, data might tend to unwanted regions. Second, the specified constraints are wide-kept. They are intentionally only based on nature laws.

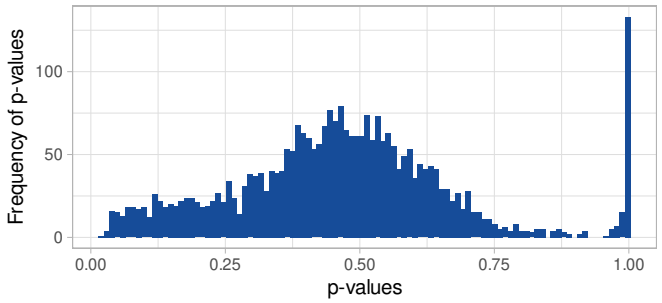


Fig. 10. Distribution of p-values for application case

That is, even if a certain range is highly likely for a given input variable, we nevertheless only enforce nature laws. Third, as nature does not always behave in a simple manner, there are relations that are only valid for given ranges of data, while for ranges another relation is valid. This could be the reason for the higher of borderline wins, as our correlation-aware operators with range-insensitive data description files would not perform as well in these cases. We plan to expand our approach to incorporate this possibility in future work.

VI. Conclusion and Further Research

We developed two new correlation-aware mutation operators and four new recombination operators to deal with categorical relations between variables in the search space of evolutionary algorithms. Additionally, we defined a data description language to accomodate the need for configurability. Information in data description files can be processed to adjust the evolutionary algorithm. Our work is highly relevant for applied research in material sciences as it paves the way to completing the method introduced by Ellendt et al. [3]. Additionally, it is straightforward to apply this to other real-world applications that are based on experiments. The only necessary adaption is the creation of different data description files in the specified grammar. Classical experiences from the natural sciences come to mind. We evaluated our approach on four well-known benchmark functions in 19.200 distinct configuration setups as well as in the given application domain. We saw overall astoundingly good results, validating that our adjusted operators can indeed preserve the structure of the search space. We plan to extend our work towards range-based relations, i.e. when relations only apply for a certain range, as well as input-output and output-output relations.

VII. Acknowledgments

We thank Heike Sonnenberg and Tobias Valentino for supplying data and expert knowledge for our application domain evaluation as well as Dr. Nils Ellendt for providing a general insight to the domain.

References

[1] M. Weber and J. Weisbrod, "Requirements engineering in automotive development-experiences and challenges," in *IEEE Joint International Conference on Requirements Engineering (RE)*, 2002, pp. 331–340.

[2] J. García, G. Jones, K. Virwani, B. McCloskey, D. Boday, G. ter Huurne, H. Horn, D. Coady, A. Bintaleb, A. Alabdulrahman, F. Alsewailem, H. Almegren, and J. Hedrick, "Recyclable, strong thermosets and organogels via paraformaldehyde condensation with diamines," *Science*, vol. 344, no. 6185, pp. 732–735, 2014.

[3] N. Ellendt and L. Mädler, "High-throughput exploration of evolutionary structural materials," *HTM Journal of Heat Treatment and Materials*, vol. 73, pp. 3–12, 2018.

[4] M. Steinbacher, G. Alexe, M. Baune, I. Bobrov, I. Bösing, B. Clausen, T. Czotscher, J. Epp, A. Fischer, L. Langstädtler, D. Meyer, S. Raj Menon, O. Riemer, H. Sonnenberg, A. Thomann, A. Toenjes, F. Vollertsen, N. Wielki, and N. Ellendt, "Descriptors for high throughput in structural materials development," *High-Throughput*, vol. 8, no. 4, 2019.

[5] A. Bader, A. Toenjes, N. Wielki, A. Mändle, A.-K. Onken, A. v. Hehl, D. Meyer, W. Brannath, and K. Tracht, "Parameter optimization in high-throughput testing for structural materials," *Materials*, vol. 12, no. 20, 2019. [Online]. Available: https://www.mdpi.com/1996-1944/12/20/3439

[6] Z. Li, X. Lin, Q. Zhang, and H. Liu, "Evolution strategies for continuous optimization: A survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 56, p. 100694, 04 2020.

[7] O. Kramer, "Evolutionary self-adaptation: A survey of operators and strategy parameters," *Evolutionary Intelligence*, vol. 3, pp. 51–65, 08 2010.

[8] N. García-Pedrajas, C. Martínez, and D. Ortiz-Boyer, "Cixl2: A crossover operator for evolutionary algorithms based on population features," *Journal of Global Optimization*, vol. 24, 09 2011.

[9] B. Craenen, A. Eiben, and E. Marchiori, *How to Handle Constraints with Evolutionary Algorithms*, 01 2001, pp. 341–361.

[10] A. Kundu, S. Laha, and A. V. Vasilakos, "Correlation-based genetic algorithm for real-parameter optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 4804–4809.

[11] R. Drechsler, S. Eggersglüß, N. Ellendt, S. Huhn, and L. Mädler, "Exploring superior structural materials using multi-objective optimization and formal techniques," in *International Symposium on Embedded Computing and System Design (ISED)*, 2016, pp. 13–17.

[12] S. Huhn, H. Sonnenberg, S. Eggersglüß, B. Clausen, and R. Drechsler, "Revealing properties of structural materials by combining regression-based algorithms and nano indentation measurements," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–6.

[13] R. Drechsler, S. Huhn, and C. Plump, "Combining machine learning and formal techniques for small data applications - A framework to explore new structural materials," in *23rd Euromicro Conference on Digital System Design, DSD 2020, Kranj, Slovenia, August 26-28, 2020*. IEEE, 2020, pp. 518–525.

[14] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing (TSP)*, vol. 52, no. 8, pp. 2275–2285, 2004.

[15] H. Sonnenberg and B. Clausen, "Short-term characterization of spherical 100cr6 steel samples using micro compression test," *Materials*, vol. 13, no. 3, 2020. [Online]. Available: https://www.mdpi.com/1996-1944/13/3/733

[16] T. Valentino, "Material characterisation with new indentation technique based on laser-induced shockwaves," *Lasers in Manufacturing and Materials Processing*, vol. 5, 12 2018.

[17] M. Eysholdt and H. Behrens, "Xtext: Implement your language faster than the quick and dirty way," in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, ser. OOPSLA '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 307–309.

[18] F. Wilhelmstötter, "Jenetics," https://jenetics.io, 2021.

[19] B. L. Welch, "The generalization of 'student's' problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947. [Online]. Available: http://www.jstor.org/stable/2332510

[20] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodologyfor comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3–18, 2011.