

Studying the Relationship Between Crossover Features and Performance on MNK-Landscapes Using Regression Models

Teruhisa Nakashima^a, Hernán Aguirre^b and Kiyoshi Tanaka
Department of Electrical and Computer Engineering, Shinshu University, Japan
{23w2805a, ahernan, ktanaka}@shinshu-u.ac.jp

Keywords: Crossover Features, Performance, Regression Models, Evolutionary Multi-Objective Optimization, MNK-Landscapes.

Abstract: Crossover is a key component of evolutionary algorithms and has been the focus of numerous studies. Its effectiveness depends on the operator's properties to mix information, the specific characteristics of the problem, and the diversity of the population, influenced by the dynamics of the algorithm. This study focuses on binary representations and introduces a method to examine the relationship between crossover features and the performance of a multi-objective evolutionary algorithm on problem subclasses with random and neighbor patterns of variable interactions. The aim is to identify the crossover features relevant to performance in each problem subclass through regression models.

1 INTRODUCTION

In the real world, many optimization problems require optimizing multiple objectives simultaneously. Numerous types of algorithms have been proposed to address these problems, one of which is Multi-objective Evolutionary Algorithms (MOEAs). MOEAs are optimization method inspired by biological evolution, where genetic operations are repeatedly applied to individuals with solutions represented as genes to perform optimization.

To enhance the solution-searching capability of a MOEA, it is necessary to tune its components. One such component is crossover, a genetic operator that generates offspring by combining the genetic information from parent individuals. The role of crossover in evolutionary algorithms has been the subject of several studies (Spears, 2000). Various crossover methods have been proposed, and their properties for mixing information have been investigated, along with their effects on performance, particularly on single objective optimization. In general, the effectiveness of crossover depends on the properties of the operator, the characteristics of the problem, and the population diversity, which is influenced by the dynamics of the algorithm.

MOEAs evolve a population of solutions aiming to find a set of Pareto non-dominated solutions, which

are usually spread in a broad region of decision and objective space. In evolutionary multi-objective optimization, there are several studies focusing on the diversity of solutions present in the population and the disruptive nature of crossing parent individuals that could be far apart in decision space (Ishibuchi et al., 2014), (Sato et al., 2013), (Sato et al., 2007). In MOEAs, crossover methods are often compared based on the performance achieved by the algorithm on a few problems, without much consideration of the operator's properties.

This work focuses on binary representations and presents a method to study and assess the relationship between crossover features and the performance of a MOEA on subclasses of problems, for which we know some of their properties. Our aim is to identify the main crossover features that are relevant to performance in each problem subclass using regression models of the features.

We first define features that capture the information-mixing characteristics of crossover operators, and then apply a crossover feature extraction method that is independent of the characteristics of the particular problem or the dynamics of the algorithm. For a given crossover operator, we quantify these features on a sample of solutions generated by the operator. Separately, for each crossover for which features were quantified, we run several times a multi-objective evolutionary algorithm on instances of a problem subclass to collect information about the performance of the algorithm. Then, we create re-

^a <https://orcid.org/0009-0009-7637-0925>

^b <https://orcid.org/0000-0003-4480-1339>

gression models to describe the relationship between the features and the performance of an algorithm.

We study the proposed approach by solving several instances of MNK-Landscapes (Aguirre and Tanaka, 2004) (Aguirre and Tanaka, 2007) with $N = 20$ bits, $M = 2$ objectives, and $K = 2$ and 4 interacting variables with *nearest neighbor* and *random* models of epistasis. For each combination of M , N , K , and type of epistasis model, we find an optimal model that determines the important features for each subclass of problems and explains performance in terms of them. We show that the models, tested on unseen problem instances, correctly predict better performance by the top-ranked operators. The proposed method could be useful to select an appropriate crossover operator.

2 METHOD

2.1 Overview

This work presents a method to study and assess the relationship between crossover features and the performance of a MOEA on subclasses of problems. The main steps of the method are as follows:

- Step 1.** Define features of the information-mixing characteristics of crossover operators, specify a set of crossover operators and compute their features.
- Step 2.** Define a problem subclass and several instances of it. With each specified crossover, run a MOEA a number of times on several instances of a problem subclass. Then, compute for each crossover a performance value in the problem subclass.
- Step 3.** Learn a regression model to capture the relationship between the crossover features and the performance values.
- Step 4.** Validate the predictive accuracy of the regression model on unseen instances of the problem subclass.

In the following we detail the characteristics of the method.

2.2 Crossover Feature Extraction

The feature extraction approach initially generates a sample offspring population from the same pair of parent individuals using a crossover method. The genetic information of the offspring population is then used to compute certain features of the operator.

Let us denote \mathcal{X} a crossover operator, and $\mathbf{p}_1, \mathbf{p}_2$ two parent individuals that are defined as binary vectors of length N , i.e. $\mathbf{p}_i = (p_{(i,1)}, \dots, p_{(i,N)}) \in \{0, 1\}^N$ where N is the number of variables. A crossover operation on the parents is denoted $\mathcal{X}(\mathbf{p}_1, \mathbf{p}_2)$ and generates two binary vectors $\in \{0, 1\}^N$ as offspring.

To efficiently track the origin of the genes in the offspring and analyze features, we specify \mathbf{p}_1 and \mathbf{p}_2 so that their corresponding j -th variables $p_{(1,j)}$ and $p_{(2,j)}$ contain different values. Namely, we prepare parent individuals $\mathbf{p}_1 = (0, 0, 0, \dots, 0)$ and $\mathbf{p}_2 = (1, 1, 1, \dots, 1)$ with N bits, perform crossover $\mathcal{X}(\mathbf{p}_1, \mathbf{p}_2)$ M times to generate a sample offspring population $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{2M}\}$, and represent the i -th offspring as $\mathbf{x}_i = (x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,N)})$. The obtained sample population is then analyzed for “search ability” and “sequence influence,” extracting four features (Caruana et al., 1989).

“Search ability” refers to the capability to efficiently search through a multitude of solution candidates, which varies between the early and late stages of the search. In the early stage, the ability to expand the search range to discover good solutions is needed. In the later stage, the ability to refine already good solution candidates is required. Two features, “Unique Rate” and “Parent Bias,” are calculated to observe these abilities.

“Sequence influence” refers to the impact of managing solutions as a concatenation of genes, which usually involves arranging the genes in contiguous locations in a one-dimensional array. This arrangement can cause various influences. Two features, namely “Adjacent Bias” and “Position Bias,” are calculated to observe these influences.

The next section explains in detail the features.

2.3 Features

2.3.1 Unique Rate

Unique Rate (UR) represents the extent to which new individuals can be generated from the same parent individuals by counting the number of unique individuals in the sample offspring population. A higher value indicates fewer duplicates and easier generation of new individuals during the search. Let us denote the vector containing the uniqueness status of the offspring in the sampled population as $\mathbf{u} = (u_1, \dots, u_{2M}) \in \{0, 1\}^{2M}$, where $u_i = 1$ indicates \mathbf{x}_i is unique and $u_i = 0$ otherwise,

$$u_i = \begin{cases} 1 & (i = 1) \\ 0 & (i \neq 1 \wedge \mathbf{x}_i \in \{\mathbf{x}_1, \dots, \mathbf{x}_{i-1}\}) \\ 1 & (i \neq 1 \wedge \mathbf{x}_i \notin \{\mathbf{x}_1, \dots, \mathbf{x}_{i-1}\}). \end{cases} \quad (1)$$

The number of unique offspring divided by the total number of offspring $2M$ gives the Unique Rate,

$$UR = \frac{\sum_{i=1}^{2M} u_i}{2M} \times 100. \quad (2)$$

2.3.2 Parent Bias

Parent Bias (PaB) represents the similarity of offspring to the two parent individuals. A higher value indicates greater similarity to one of the parents, suggesting an inheritance bias towards one of parent characteristics during the search. When offspring receive genes equally from both parents, the total number of genes received from each parent should be half the number of bits N . To calculate parent bias, we use the absolute value of the difference between the actual number of genes from one parent and $N/2$. In our case, it is easy to count the bits set to 1 in the offspring, which are known to come from parent \mathbf{p}_2 . The average of these differences in the sampled offspring population gives the Parent Bias,

$$PaB = \frac{\sum_{i=1}^{2M} \left| \left(\sum_{j=1}^N x_{(i,j)} \right) - \frac{N}{2} \right|}{2M} \times 100. \quad (3)$$

2.3.3 Adjacent Bias

Adjacent Bias (AB) represents the continuity of gene inheritance from the same parent, indicating the tendency of neighboring genes in the offspring to originate from the same or different parents. A higher value shows greater continuity. Given the offspring \mathbf{x}_i , we count at the j -th bit position, $j = 1, \dots, N$, the length l_j of the uninterrupted sequence $x_{(i,j-l_j)}, \dots, x_{(i,j-1)}, x_{(i,j)}$ of bits inherited from the same parent, i.e., $x_{(i,k)} \in \mathbf{p}_1 \vee x_{(i,k)} \in \mathbf{p}_2, \forall k \in \{j-l_j, \dots, j-1, j\}$. Let us denote the vector of counters l_j for the offspring \mathbf{x}_i as $\mathbf{v}_i = (v_{(i,1)}, \dots, v_{(i,N)})$, $v_{(i,j)} = l_j \in \{0, \dots, N-1\}, j = 1, \dots, N$. Since all bits from \mathbf{p}_1 are 0s and from \mathbf{p}_2 are 1s, the counter of continuous inheritance $v_{(i,j)}$ for the i -th offspring at the j -th bit position is easily computed as follows,

$$v_{(i,j)} = \begin{cases} 0 & (j=1) \\ v_{(i,j-1)} + 1 & (j \neq 1 \wedge x_{(i,j-1)} = x_{(i,j)}) \\ 0 & (j \neq 1 \wedge x_{(i,j-1)} \neq x_{(i,j)}) \end{cases} \quad (4)$$

The sum of the element of \mathbf{v}_i divided by the maximum possible sum of \mathbf{v}_i across all offspring gives the Adjacent Bias,

$$AB = \frac{\sum_{i=1}^{2M} \frac{\sum_{j=1}^N v_{(i,j)}}{\frac{N(N-1)}{2}}}{2M} \times 100. \quad (5)$$

$AB = 0$ if all $2M$ offspring are perfectly uniform. That is, $\forall i = 1, \dots, 2M$, for each bit position

$j = 2, \dots, N-1$, the bit $x_{(i,j)}$ came from one parent and the neighboring bits $x_{(i,j-1)}$ and $x_{(i,j+1)}$ from the other parent. On the other hand, $AB = 100$ if all $2M$ offspring are clones of one of the parents. That is, $\forall i = 1, \dots, 2M$, at each bit position $j = 1, \dots, N$, the bit $x_{(i,j)}$ came from the same parent.

2.3.4 Position Bias

Upon examination of a particular gene position (loci) within the sampled offspring population, if on average half of the offspring inherits its gene from parent \mathbf{p}_1 and the other half from parent \mathbf{p}_2 , we say that the operator is positionally unbiased.

Position Bias (PoB) measures the frequency of the parents' genes in their offspring at every gene position and computes the deviation from the expected positionally unbiased frequency. A higher value shows stronger positional bias. Since two offspring are generated per crossover, only one offspring is used for calculation. The gene frequency at each gene position is determined by the sum of gene values '1' from parent \mathbf{p}_2 at that position. The difference between this sum and half the number of crossover operations, averaged across gene positions is the positional bias, as shown below,

$$PoB = \frac{\sum_{j=1}^N \left| \left(\sum_{i=1}^M x_{(2i,j)} \right) - \frac{M}{2} \right|}{N} \times 100. \quad (6)$$

2.4 Performance Value

In our study we consider several crossover operators, which for convenience we identify as $\mathcal{X}_k, k = 1, \dots, \alpha$. To analyze the relationship between crossover features and performance, we calculate a performance value (PV) on a problem subclass for each crossover operator. MOEAs are stochastic algorithms. To provide an estimate of their performance on an instance of the problem subclass, they are usually run several times on the same instance. This leads to a range of values for the performance metric, where the mean performance of the MOEA lies within a certain probability. This range is expected to vary from instance to instance, though the instances belong to the same problem subclass. Thus, to compute PV we take into account the diverse ranges of the performance metric across multiple problem instances within the subclass.

As mentioned above, the MOEA set with crossover \mathcal{X}_k is run several times on the same instance. To measure performance, we use the Hypervolume (HV)(Zitzler et al., 2003) computed on the set of non-dominated solutions in the population at the final generation of the MOEA's run. Let us denote the

problem instances with the index $i = 1, \dots, \beta$, and the runs of the MOEA with the index $j = 1, \dots, \rho$. Then, $HV_{(i,j)}^{X_k}$ denotes the HV achieved by the MOEA with crossover X_k in the j -th run on the i -th problem instance.

First, the average HV value of all runs on the i -th problem instance for crossover X_k was calculated by

$$\overline{HV}_i^{X_k} = \frac{1}{\gamma} \sum_{j=1}^{\gamma} HV_{(i,j)}^{X_k}. \quad (7)$$

Next, the average HV value of all crossover operators on the i -th problem instance was computed by

$$\overline{HV}_i = \frac{1}{\alpha} \sum_{k=1}^{\alpha} \overline{HV}_i^{X_k}. \quad (8)$$

To address the differences in ranges of HV value across problem instances, we standardized the HV value (SHV) for crossover X_k on the i -th problem instance by

$$SHV_i^{X_k} = \frac{\overline{HV}_i^{X_k} - \overline{HV}_i}{\sqrt{\frac{1}{\alpha} \sum_{k=1}^{\alpha} (\overline{HV}_i^{X_k} - \overline{HV}_i)^2}}. \quad (9)$$

The average SHV for each crossover across all problem instances was taken as the performance value PV for the problem subclass,

$$PV^{X_k} = \overline{SHV}^{X_k} = \frac{1}{\beta} \sum_{i=1}^{\beta} SHV_i^{X_k}. \quad (10)$$

2.5 Model Learning

The performance of the MOEA set with the crossover operator $X_k, k = 1, \dots, \alpha$, on a problem subclass and the features measured for the same operator are collected as data points. We extract the features unique rate UR^{X_k} , parent bias PaB^{X_k} , adjacent bias AB^{X_k} , and positional bias PoB^{X_k} using equations (2)-(6). We repeat this process several times using different random seeds and take their average as the feature value of the crossover, i.e., \overline{UR}^{X_k} , \overline{PaB}^{X_k} , \overline{AB}^{X_k} , and \overline{PoB}^{X_k} . Thus, for each problem subclass, we have α data points $(PV^{X_k}, \overline{UR}^{X_k}, \overline{PaB}^{X_k}, \overline{AB}^{X_k}, \overline{PoB}^{X_k})$ corresponding to the same number of crossover types used in this study.

To examine the relationship between crossover features and problem performance, PV was used as the dependent variable, and the four features UR , PaB , AB , and PoB were used as the explanatory variables. For each problem subclass, linear regression models $PV = f(UR, PaB, AB, PoB)$ were created for all 15 combinations of the explanatory variables. Models

in which the coefficients of all explanatory variables had a p-value of 0.05 or lower were retrained, splitting the data randomly into training (80%) and validation (20%) datasets to calculate the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). This process was repeated 50 times, using different random seeds for the splitting of the data, to obtain the average RMSE and MAE. The model f^* with the lowest RMSE and MAE on the validation datasets was considered the linear regression model that best captures the relationship between the problem subclass and crossover features.

2.6 Model Verification

To verify the regression model's accuracy, we use the selected model f^* to predict the performance value of each crossover, i.e., $\widehat{PV}^{X_k} = f^*(\overline{UR}^{X_k}, \overline{PaB}^{X_k}, \overline{AB}^{X_k}, \overline{PoB}^{X_k})$. We sort according to \widehat{PV}^{X_k} and determine the sets of the 5 best and worst performing crossovers, defined as follows

$$X^{best} = \{X_{k_i} \mid 1 \leq i \leq 5\}$$

$$X^{worst} = \{X_{k_i} \mid \alpha - 5 \leq i \leq \alpha\},$$

where (k_1, \dots, k_α) is the the ordered list of crossover indexes such that $\widehat{PV}^{X_{k_1}} > \widehat{PV}^{X_{k_2}} > \dots > \widehat{PV}^{X_{k_\alpha}}$. In addition, we also determine the mean performing crossover, i.e., those with the predicted performance value \widehat{PV}^{X_k} closest to zero. We sort according to the absolute value $|\widehat{PV}^{X_k}|$ and determine the 5 mean performing crossovers as follows

$$X^{mean} = \{X_{k_i} \mid 1 \leq i \leq 5\},$$

where (k_1, \dots, k_α) is the the ordered list of crossover indexes such that $|\widehat{PV}^{X_{k_1}}| < |\widehat{PV}^{X_{k_2}}| < \dots < |\widehat{PV}^{X_{k_\alpha}}|$.

For each crossover operator $X_b \in X^{best}$, $X_m \in X^{mean}$ and $X_w \in X^{worst}$, we optimize 10 different unseen instances 50 times each with different random seeds using the same MOEA configuration. For the i -th unseen instance, we compute the HV for each run of the MOEA with crossover X , $HV_i^X = \{HV_{(i,1)}^X, \dots, HV_{(i,50)}^X\}$, and perform pairwise t-tests between the HV results by each crossover $X_b \in X^{best}$ and each crossover $X_w \in X^{worst}$. Thus, in total we perform 25 t-tests per instance and summarize their results by counting the number of times the average

hypervolume $\overline{HV}_i^{X_b}$ by $X_b \in \mathcal{X}^{best}$ is better, no significantly different, or worse than $\overline{HV}_i^{X_w}$ by $X_w \in \mathcal{X}^{worst}$. Similarly, we perform pairwise t -tests for each $X_b \in \mathcal{X}^{best}$ and each $X_m \in \mathcal{X}^{mean}$ and summarize their results.

3 MNK-LANDSCAPES

NK-Landscapes (Kauffman, 1993) are well-known mathematical models of rugged fitness landscapes with a configurable number of variables and landscape ruggedness, specified by the parameters N and K , respectively. They were extended to the multi-objective domain by adding another parameter M , which defines the number of objectives, and considering different variable epistasis models for each objective. This multi-objective version is known as MNK-Landscapes (Aguirre and Tanaka, 2004) and is mathematically represented as:

$$f_i(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^N f_{i,j}(x_j, \underbrace{z_1^{(i,j)}, z_2^{(i,j)}, \dots, z_{K_i}^{(i,j)}}_{K_i \text{ bits interacting with } x_j}), \quad (11)$$

where f_i denotes the i -th fitness function, $i = 1, 2, \dots, M$, $\mathbf{x} = (x_1, \dots, x_N) \in \{0, 1\}^N$, $f_{i,j} : B^{K_i+1} \rightarrow \mathbb{R}$ gives the fitness contribution of bit x_j to f_i , and $z_1^{(i,j)}, \dots, z_{K_i}^{(i,j)}$ are the K_i bits interacting with bit x_j in the string \mathbf{x} . This means that the fitness value of the bit x_j depends not only on its own value but also on the other K_i bits that it interacts with. In other words, the subfunction $f_{i,j}(x_j, z_1^{(i,j)}, z_2^{(i,j)}, \dots, z_{K_i}^{(i,j)})$ is a lookup table of 2^{K_i+1} fitness values, one per each combination of the $K_i + 1$ variables. These fitness values are random real numbers in the range $[0.0, 1.0]$ drawn from a uniform distribution $U(0, 1)$. A larger K_i creates a more rugged landscape, which makes the benchmark problem more difficult to solve (Aguirre and Tanaka, 2004)(Aguirre and Tanaka, 2007)(Pelikan, 2010)(Daolio et al., 2015)(Martins et al., 2021).

The K_i bits that interact with bit x_j define an epistasis model of variable interactions, which is specified either by the *nearest neighbor* bits of x_j or *random* bits, i.e. $z_k^{(i,j)} = x_{\hat{j}_k} \in \{x_1, \dots, x_N\}$ and $\hat{j} = U(1, \dots, N)$. Note that when the epistasis model is random, the subset of bits that interact with x_j is different for each fitness function. However, even if the epistasis model for bit x_j is the same in all objectives, as is the case in the nearest neighbor model, the fitness contribution of x_j is different for each objective function due to the random generation of $f_{i,j}$.

The properties of MNK-Landscapes in regards to each configurable parameter, number of variables, ob-

Table 1: Crossover Operators.

Crossover	Parameter
One Point	
Two Point	
Multi Point	$\{3, 4, \dots, 19\}$
Uniform	
Half Uniform	
Exponential	$\{0.05, 0.1, \dots, 0.9\}$
Binomial	$\{0.05, 0.1, \dots, 0.9\}$

Table 2: Crossover Feature Extraction.

Number of Bits N	20
Crossover Executions M	50
Experiments	100
Crossover Types α	82

Table 3: MNK-Landscapes.

Objectives O	2
Variables N	20
Interacting Variables K	2, 4
Epistasis Model	Random, Near
Instances for Training β	30
Instances for Verification	10

Table 4: MOEA.

Generations	500
Population Size	100
Crossover	See Table1
Crossover Probability	1.0
Mutation	Bit flip
Bit Mutation Rate	0.1
Runs for Training γ	10
Runs for Validation	50

jectives, and epistasis, and their effect on the performance of MOEA have been studied by Aguirre et al in (Aguirre and Tanaka, 2004) and (Aguirre and Tanaka, 2007).

In this work, we analyze MNK-Landscapes with *random* and *nearest neighbor* models of epistasis with the same number K of interacting variables in all objective functions, i.e. $K_i = K, i = 1, \dots, M$.

4 EXPERIMENTAL DETAILS

We investigate the relationship between crossover features and performance on subclasses of problems with interacting variables under *nearest neighbor* and *random* models of epistasis.

The crossover used in this study are listed in Table 1. They include the commonly used One and

Two Point crossovers. Uniform crossover (Syswerda, 1989) clones the parents and swaps bits with a probability of 0.5 to create offspring. Half Uniform crossover swaps exactly half of the non matching bits between the parents. Multi Point crossover uses the number of crossing points as a parameter, whereas Exponential and Binomial crossover operators (Storn and Price, 1995)(Price et al., 2005) use a crossover rate per variable as a parameter. The Binomial crossover applies the specified crossover rate to all variables, similar to Uniform crossover. On the other hand, the Exponential crossover applies the specified crossover rate to the first variable and then reduces it exponentially for the subsequent variables. Variations of these latter three operators are considered by setting them with different values of their parameters, as shown in Table 1. Considering the 7 basic operators in Table 1 and their parameters, we have 41 types of crossover operators. Additionally, we also considered shuffled (Caruana et al., 1989) versions of these operators. Thus, in total, we have $\alpha = 82$ types of crossover operators. For convenience, we denote these crossover operators as X_k , $k = 1, \dots, \alpha$.

Feature extraction was performed for these crossovers assuming individuals have $N=20$ variables. With each crossover operator X_k and the predefined parents \mathbf{p}_1 and \mathbf{p}_2 set as explained in Section 2, we create a sample offspring population of size $2M$ applying $X_k(\mathbf{p}_1, \mathbf{p}_2)$ $M = 50$ times and compute the features. We repeat this process 100 times, each time with a distinct sample offspring population, to calculate the average feature values for each crossover. The parameters used for feature extraction are summarized in Table 2.

The benchmark problem for performance evaluation was MNK-Landscapes(Aguirre and Tanaka, 2007). In this study, we use landscapes with $M = 2$ objectives, $N = 20$ bits, and $K = 2$ and 4 interacting variables with *nearest neighbor* (Near) and *random* (Random) models of epistasis. To compute the performance value of the crossovers and train the models, we solve $\beta = 30$ problem instances for each of the four problem subclasses determined by the combinations of K and epistasis models. To verify the models we use other 10 instances per problem subclass. Overall we use 160 problem instances in this study. The parameters of the problems used are summarized in Table 3.

The MOEA configuration used for optimizing these problems employed a NSGA-II algorithm (Deb et al., 2002) with a population size of 100 running for 500 generations. The crossover probability was set to 1.0, and the mutation used was Bit Flip Mutation with a mutation rate of 0.1. We add to this

configuration one of the above X_k crossover operators and compile results on the algorithm's performance for each crossover, $k = 1, \dots, 82$. Our implementation uses the framework proposed by (Blank and Deb, 2020). The MOEAs are run 10 times in each instance to compute the performance values of the crossovers and train the models. On the other hand, the MOEAs are run 50 times in each instance to validate the models. Table 4 summarizes the parameters used for the multi-objective optimizer.

The reference point for the HV is (0.0, 0.0).

5 RESULTS AND DISCUSSION

5.1 Selected Models

We first examine the general characteristics of the selected models. The coefficients, standard errors, t-values, p-values, confidence intervals, coefficient of determination R^2 , and adjusted R^2 for the selected models f^* with the lowest RMSE and MAE for each problem subclass are shown in Tables 5 - 8.

Looking at the explanatory variables selected in the models, we find that for problems with random variable interaction, Parent Bias alone can predict performance for $K=2$ and $K=4$. On the other hand, for problems with near variable interaction, Unique Rate, and Adjacent Bias are selected for $K=2$ and $K=4$, with Parent Bias also being included for $K = 4$. Since problems with near variable interaction have objective function values influenced by adjacent genes, the selection of Adjacent Bias in the regression model indicates that problem characteristics are reflected in the crossover features.

The R^2 and adjusted R^2 values are in the range 0.83-0.94, indicating that in all problem subclasses a large proportion of the variation in the performance value PV is predictable from the explanatory variables.

Below, we analyze the regression models in detail, grouping by type of variable interaction.

5.1.1 Random Variable Interaction

The models for the two problem subclasses with random variable interaction are simple linear regression models with Parent Bias as the explanatory variable. Comparing the two models in Tables 5, and 6, the model for $K = 4$ has a higher R^2 and adjusted R^2 with lower RMSE and MAE than the model for $K=2$, indicating higher prediction accuracy and fewer outliers.

Plots showing the regression lines and data for each crossover are presented in Figures 1, and 2. In

Table 5: Best Regression Model for $M = 2$, $N = 20$, $K = 2$, and random variable interactions on 30 instances.

	coef	std error	t-value	p-value	95%CI	R^2	Adj. R^2	RMSE	MAE
Const	-0.7548	0.040	-19.029	0.000	-0.834, -0.676	0.875	0.873	0.202	0.134
PaB	0.0181	0.001	23.640	0.000	0.017, 0.020				

Table 6: Best Regression Model for $M = 2$, $N = 20$, $K = 4$, and random variable interactions on 30 instances.

	coef	std error	t-value	p-value	95%CI	R^2	Adj. R^2	RMSE	MAE
Const	-0.9904	0.034	-29.487	0.000	-1.057, -0.924	0.944	0.943	0.174	0.128
PaB	0.0237	0.001	36.632	0.000	0.022, 0.025				

Table 7: Best Regression Model for $M = 2$, $N = 20$, $K = 2$, and near variable interactions on 30 instances.

	coef	std error	t-value	p-value	95%CI	R^2	Adj. R^2	RMSE	MAE
Const	-2.1858	0.158	-13.844	0.000	-2.500, -1.872	0.894	0.891	0.220	0.150
UR	0.0137	0.001	9.457	0.000	0.011, 0.017				
AB	0.0396	0.002	23.665	0.000	0.036, 0.043				

Table 8: Best Regression Model for $M = 2$, $N = 20$, $K = 4$, and near variable interactions on 30 instances.

	coef	std error	t-value	p-value	95%CI	R^2	Adj. R^2	RMSE	MAE
Const	-2.2277	0.151	-14.738	0.000	-2.529, -1.927	0.923	0.920	0.220	0.147
UR	0.0127	0.001	9.019	0.000	0.010, 0.015				
PaB	0.0073	0.002	3.057	0.003	0.003, 0.012				
AB	0.0330	0.004	8.705	0.000	0.025, 0.041				

both figures, the data and the models indicate that higher Parent Bias corresponds to better performance. The regression coefficient for Parent Bias is larger for $K = 4$ and the slope of the regression line is steeper, suggesting a greater influence of the explanatory variable in problems with larger K . Note that the best-performing crossovers have Parent Bias around 90%. This suggests that the most successful crossovers are those that produce offspring that are one or two bits different from one of the parents. To verify this, column Random in Table 9 shows the features and performance values of the 5 crossovers predicted to be the best by the models (higher \widehat{PV}) for these problem subclasses. Since in both models the only explanatory variable is Parent Bias, the same crossovers are predicted to perform better both in $K=2$ and $K=4$. Note that 4 of the 5 crossovers share the same highest Parent Bias value, i.e., Exponential (E), Binomial (B) and their shuffled versions (SE and SB) with the smallest rate of 0.05, which is equivalent to accepting, on average, only one bit from the second parent. The fifth crossover is shuffled exponential (SE) with parameter 0.10, which, on average, accepts 2 bits from the second parent.

From the same figures, note that the error between the predicted \widehat{PV} and actual PV is larger for crossovers with smaller Parent Bias, particularly when Parent Bias is zero. Crossovers with a Parent

Bias of zero include Half Uniform (HU) and Multi Point with 19 crossover points (MP19), along with their shuffled versions (SHU and SMP19). Table 10 summarizes the features, actual PV , and predicted \widehat{PV} performance values for these crossovers. Note that HU and MP19 achieve a PV well below the mean ($PV < 0.0$), and their shuffled versions improve their actual PV . However, only SHU performs significantly better than the mean for $K=2$, i.e. $PV = 0.422 > 0.0$, and slightly better than the mean for $K=4$, i.e. $PV = 0.081 > 0.0$. HU, SHU, and SMP19 share similar features. However, their performance is very different and cannot be explained neither by Parent Bias nor by the other features of these crossovers. MP19 is a special case where its extremely poor performance, $PV = -1.399$ and $PV = -1.174$, can be explained by its low Unique Rate.

5.1.2 Near Variable Interaction

The models for the two problem subclasses with near variable interaction are multiple regression models with Unique Rate and Adjacent Bias as explanatory variables for $K = 2$, and Parent Bias added for $K = 4$.

From Tables 7 and 8, note that the regression coefficient for Adjacent Bias is three times larger than the coefficient for Unique Rate. The models predict better performance for crossovers with a high Adjacent Bias that can pass adjacent bits from the same parent to

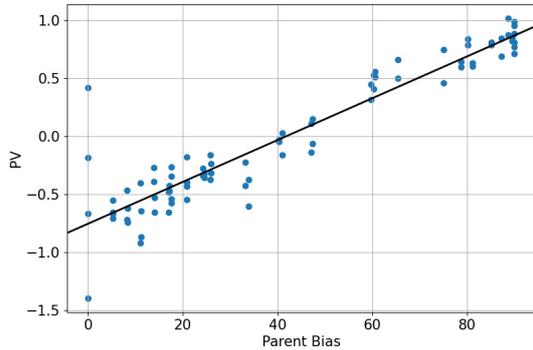


Figure 1: Regression Model for $K = 2$, Random.

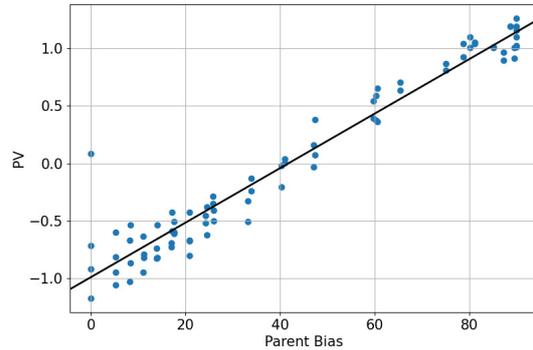


Figure 2: Regression Model for $K = 4$, Random.

Table 9: The 5 best performing crossovers $K = 2, 4$ with Random and Near variable interactions.

		Random					Near				
		SE0.05	B0.05	SB0.05	E0.05	E0.1	E0.9	E0.8	E0.7	E0.6	E0.5
Unique Rate		37.04	37.08	37.08	36.68	37.82	79.40	82.64	73.76	64.98	56.32
Parent Bias		90.0	90.0	90.0	90.0	89.9	60.54	65.41	75.03	81.21	85.08
Adjacent Bias		59.81	60.18	60.16	60.50	60.00	54.90	52.37	54.65	56.91	58.16
Position Bias		90.0	90.0	90.0	90.0	89.9	19.86	57.81	73.57	80.98	85.03
K=2	PV	0.717	0.958	0.887	0.772	0.814	0.939	0.0.868	0.923	0.779	0.901
	\widehat{PV}	0.871	0.871	0.871	0.871	0.869	0.927	0.879	0.853	0.829	0.771
K=4	PV	1.011	1.100	1.158	1.261	1.025	0.955	0.949	1.089	1.101	0.850
	\widehat{PV}	1.143	1.143	1.143	1.143	1.141	0.960	0.938	0.971	0.987	0.960

Table 10: Crossover with Parent Bias of 0.

		HU	SHU	MP19	SMP19
Unique Rate		99.94	99.98	2	99.96
Parent Bias		0	0	0	0
Adjacent Bias		7.90	7.88	0	7.76
Position Bias		11.32	11.23	0	11.69
K=2	PV	-0.182	0.422	-1.399	-0.666
	\widehat{PV}	-0.755	-0.755	-0.755	-0.755
K=4	PV	-0.922	0.081	-1.174	-0.718
	\widehat{PV}	-0.990	-0.990	-0.990	-0.990

the offspring while maintaining a good Unique Rate to avoid creating the same offspring several times.

Similar to random variable interactions, comparing the two models for near variable interactions, the model for $K = 4$ has a higher R^2 and adjusted R^2 with lower RMSE and MAE than the model for $K=2$. To further investigate the better prediction accuracy for larger K , Table 11 shows the results of a suboptimal model for $K=4$, which excludes Parent Bias and includes only Unique Rate and Adjacent Bias, the explanatory variables included in the best model for $K=2$. From Table 11, note that R^2 of the suboptimal model reduces to 0.914 compared to $R^2 = 0.923$ of the optimal model for $K=4$ shown in Table 8. How-

ever, the R^2 of the suboptimal model for $K=4$ it is still better than the $R^2 = 0.894$ of the optimal model for $K=2$ shown in Table 7, although both use the same explanatory variables. Summarizing, prediction accuracy is better for $K=4$ than for $K=2$, whether the variable interactions are random or near.

Since these are multiple linear regression models, the models are visualized using residual plots, shown in Figures 3, and 4. In Figure 3, as the predicted performance \widehat{PV} exceeds -0.5, the range of possible residuals decreases, indicating higher accuracy for higher \widehat{PV} values. Conversely, around $\widehat{PV} = -0.5$, the range of possible residuals is larger, with both positive

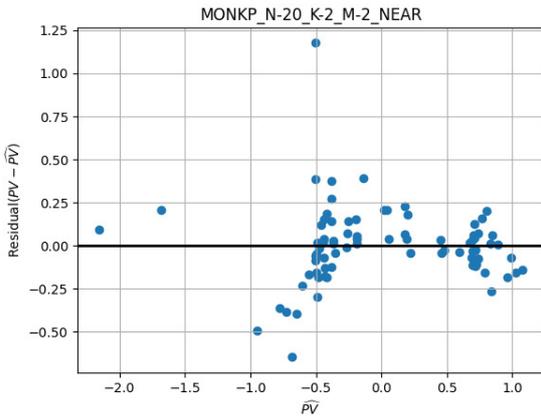


Figure 3: Residual Plot for $K = 2$, Near.

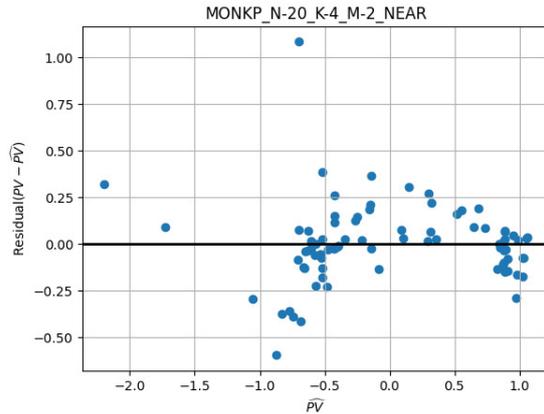


Figure 4: Residual Plot for $K = 4$, Near.

and negative distributions, indicating both underpredictions and overpredictions. In Figure 4, the range of possible residuals is smaller than for $K = 2$, indicating higher prediction accuracy. This result is also reflected in the R^2 values in Tables 7, and 8. However, for $K = 4$, there is a data point with residual exceeding 1.0.

In Figures 3 and 4, the two leftmost points correspond to 18 and 19 point crossovers. Their predicted performance values \widehat{PV} are underestimated, but looking at the actual PV are two of the worst performing operators. Similarly, the point with the largest positive residual, > 0.6 in $K=2$ and > 1.0 in $K=4$, correspond to shuffle half uniform crossover. The predicted performance value \widehat{PV} of this operator is also largely underestimated. However, looking at the actual PV it is above average (> 0.0) but far from being among the best. The most overestimated crossovers, those that appear with negative residuals, correspond to multi point crossovers with 10 to 17 crossing points. These crossovers are also in the group of worst-performing operators.

Column Near in Table 9 shows the features and performance values of the 5 crossovers predicted to be the best (higher \widehat{PV}) by the models for the near variable interaction problem subclasses. Although the number of explanatory variables for $K=2$ and $K=4$ is different, the same crossovers are predicted as the best in both models. Note that all 5 operators are exponential crossovers, with the parameter ≥ 0.5 . However, their rank order is different. For $K=4$, to compute \widehat{PV} the model also takes into account the value of the Parent Bias feature, in addition to Unique Rate and Adjacent Bias used for $K=2$.

Table 11: Suboptimal Regression Model for $K=4$, Near 30 instances.

	coefficient	p-value
Constant	-2.2529	0.000
Unique Rate	0.0133	0.000
Adjacent Bias	0.0436	0.000
R^2	0.914	
Adj. R^2	0.911	
RMSE	0.233	
MAE	0.169	

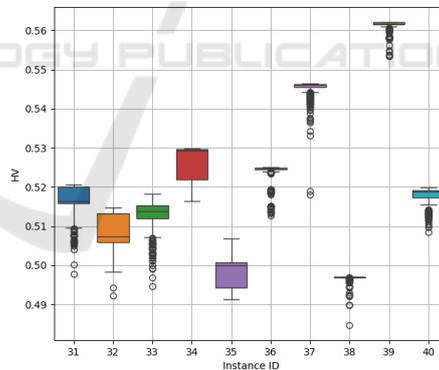


Figure 5: HV Boxplot on validation instances for $K = 2$, Random.

5.2 Model Validation with Unseen Data

Each regression model was learned to estimate the performance of crossover operators in a problem subclass. In this section, we examine the predictive power of the models for individual unseen instances in each problem subclass. As explained in Section 2.6, we identify three groups of crossovers, namely the best 5, worst 5, and mean 5 performing crossovers,

Table 12: X_b and X_w t-Test Results $K = 2$, Random.

ID	$X_b > X_w$	$X_b \sim X_w$	$X_b < X_w$
31	24	1	0
32	21	4	0
33	25	0	0
34	21	4	0
35	25	0	0
36	4	19	2
37	20	5	0
38	0	22	3
39	16	9	0
40	17	8	0

Table 16: X_b and X_m t-Test Results $K = 2$, Random.

ID	$X_b > X_m$	$X_b \sim X_m$	$X_b < X_m$
31	25	0	0
32	21	4	0
33	25	0	0
34	25	0	0
35	22	3	0
36	1	23	1
37	17	8	0
38	0	19	6
39	10	15	0
40	11	14	0

Table 13: X_b and X_w t-Test Results $K = 4$, Random.

ID	$X_b > X_w$	$X_b \sim X_w$	$X_b < X_w$
31	23	2	0
32	25	0	0
33	21	4	0
34	23	2	0
35	25	0	0
36	23	2	0
37	23	2	0
38	22	3	0
39	25	0	0
40	21	4	0

Table 17: X_b and X_m t-Test Results $K = 4$, Random.

ID	$X_b > X_m$	$X_b \sim X_m$	$X_b < X_m$
31	18	7	0
32	25	0	0
33	15	10	0
34	23	2	0
35	25	0	0
36	21	4	0
37	19	6	0
38	25	0	0
39	18	7	0
40	15	10	0

Table 14: X_b and X_w t-Test Results $K = 2$, Near.

ID	$X_b > X_w$	$X_b \sim X_w$	$X_b < X_w$
31	25	0	0
32	23	2	0
33	25	0	0
34	25	0	0
35	24	1	0
36	25	0	0
37	25	0	0
38	25	0	0
39	25	0	0
40	25	0	0

Table 18: X_b and X_m t-Test Results $K = 2$, Near.

ID	$X_b > X_m$	$X_b \sim X_m$	$X_b < X_m$
31	25	0	0
32	6	19	0
33	11	14	0
34	18	7	0
35	14	11	0
36	23	2	0
37	10	15	0
38	19	6	0
39	19	6	0
40	13	12	0

Table 15: X_b and X_w t-Test Results $K = 4$, Near.

ID	$X_b > X_w$	$X_b \sim X_w$	$X_b < X_w$
31	25	0	0
32	25	0	0
33	25	0	0
34	25	0	0
35	25	0	0
36	25	0	0
37	25	0	0
38	25	0	0
39	25	0	0
40	25	0	0

Table 19: X_b and X_m t-Test Results $K = 4$, Near.

ID	$X_b > X_m$	$X_b \sim X_m$	$X_b < X_m$
31	22	3	0
32	25	0	0
33	16	9	0
34	22	3	0
35	24	1	0
36	25	0	0
37	25	0	0
38	22	3	0
39	23	2	0
40	22	3	0

and perform pairwise t-tests between the best and the other crossovers. To illustrate the variance among instances, Figure 5 shows box plots of the HV achieved in the 50 runs of the 5 best and 5 worst crossover operators, with 500 data points in total, in each unseen instance of problem subclass $K = 2$ with random variable interaction. Note that the distributions of the points and their medium values vary greatly from instance to instance.

The summary of the Welch's t-test performed to compare the mean hypervolume on 50 runs achieved by the 5 best $X_b \in \mathcal{X}^{best}$ and 5 worst $X_w \in \mathcal{X}^{worst}$ crossovers is shown in Tables 12-15. Results are shown for 10 unseen instances in each problem subclass, i.e., instances not used to train the models. The unseen instances are identified with a number between 31 and 40.

The t-tests try to assess whether the models could be useful to classify good crossovers from bad ones in particular instances of the problem subclasses. If there is a clear differentiation between the best and worst crossovers for a particular instance, $X_b > X_w$ should approach 25 and $X_b < X_w$ should approach 0, indicating that the mean hypervolume by most $X_b \in \mathcal{X}^{best}$ is better than by $X_w \in \mathcal{X}^{worst}$. As shown in Tables 13,14, and 15, $X_b < X_w$ is zero for all instances, indicating the model is useful. However, in Table 12, instances 36 and 38 have non-zero values for $X_b < X_w$. To determine if this result is due to the model or the instances, Figure 6 shows separate box plots for the HV achieved by the best 5 and the worst 5 crossovers for instances 33, 36, 38, and 40. In instance 33, there is a perfect score of $X_b > X_w = 25$ in favor of the best crossover operators, which is in accordance with the box plots that show distributions of HV values clearly centered around different means. In instance 40, the scores $X_b > X_w = 17$ and $X_b \sim X_w = 8$ indicate that the best crossovers are mostly better than and sometimes similar to the worst crossovers. The box plots show that the distributions of the two groups are different.

In instances 36 and 38, the scores $X_b \sim X_w = 19$ and 22, respectively, show that, mostly, there is no statistical difference in the HV means between the best and worst crossovers. The box plots confirm this, and also show that the distributions of HV values are very similar, with a small standard deviation around the mean. This suggests that these instances are simpler, where crossover features make no difference in performance and therefore the majority of crossovers will achieve a similar HV.

Tables 16-19 show the results of the pairwise t-tests between the best $X_b \in \mathcal{X}^{best}$ and the mean $X_m \in \mathcal{X}^{mean}$ performing crossovers. Note that in almost all problem subclasses and instances, X_b 's perfor-

mance is better or similar to X_m . As expected, due to the variability in the instances, there are more cases where the performance of the predicted best and mean crossovers in the subclass perform similarly in a given instance.

A summary of the mean performing crossovers in each problem subclass is shown in Table 20. It can be seen that in random variable interactions, the group of crossovers considered as mean performing is the same for $K=2$ and $K=4$. In near variable interactions, three crossovers are common for $K=2$ and $K=4$. Also note that the groups of mean performing crossovers are different for random and near variable interactions.

These results show that the models are useful for classifying good performing crossovers from average and bad performing crossover in particular instances.

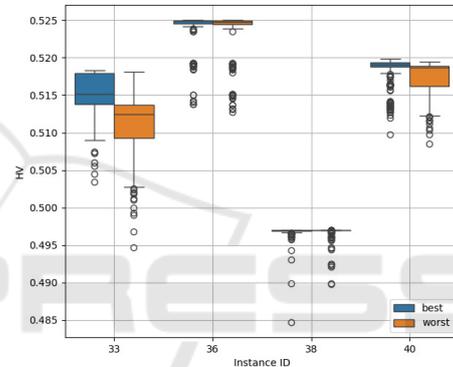


Figure 6: HV Boxplot by best and worst crossovers on selected instances for $K = 2$, Random.

Table 20: Mean performing crossovers.

	Random		Near	
	K=2	K=4	K=2	K=4
B0.7	✓	✓		✓
SB0.7	✓	✓		
B0.3	✓	✓		
SB0.3	✓	✓		
OP	✓	✓		
MP4				✓
MP5			✓	✓
MP6			✓	
SOP			✓	✓
STP			✓	✓
SB0.2			✓	

6 CONCLUSION

This work presented a method to study the relationship between crossover features and the performance

of a multi-objective evolutionary algorithm on subclasses of problems by using regression models. Several crossover operators were studied, extracting their features with a method independent of the characteristics of the problem instances or the dynamics of an evolutionary algorithm. A performance value for a problem subclass, relative to other operators, was computed for each crossover running a multi-objective evolutionary algorithm on several instances. We used MNK-Landscapes with random and near-variable interactions to define problem subclasses. We verified that the models identified relevant features for each problem subclass and can explain a large proportion of the performance variance.

In the future, we would like to validate the method with other multi-objective evolutionary algorithm configurations and use problems with more variables and objectives. Also, it would be interesting to add problem features to obtain more general models.

REFERENCES

- Aguirre, H. E. and Tanaka, K. (2004). Insights on properties of multiobjective MNK-landscapes. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 1, pages 196–203 Vol.1. IEEE.
- Aguirre, H. E. and Tanaka, K. (2007). Working principles, behavior, and performance of moeas on mnk-landscapes. *European Journal of Operational Research*, 181(3):1670–1690.
- Blank, J. and Deb, K. (2020). pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509.
- Caruana, R., Eshelman, L., and Schaffer, J. (1989). Representation and hidden bias ii: Eliminating defining length bias in genetic search via shuffle crossover. volume 1, pages 750–755.
- Daolio, F., Liefoghe, A., Verel, S., Aguirre, H., and Tanaka, K. (2015). Global vs local search on multi-objective nk-landscapes: Contrasting the impact of problem features. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15*, pages 369–376, New York, NY, USA. Association for Computing Machinery.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Ishibuchi, H., Tanigaki, Y., Masuda, H., and Nojima, Y. (2014). Distance-based analysis of crossover operators for many-objective knapsack problems. In Bartz-Beielstein, T., Branke, J., Filipič, B., and Smith, J., editors, *Parallel Problem Solving from Nature – PPSN XIII*, pages 600–610, Cham. Springer International Publishing.
- Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.
- Martins, M. S. R., Yafrani, M. E., Delgado, M., Lüders, R., Santana, R., Siqueira, H. V., Akcay, H. G., and Ahiod, B. (2021). Analysis of bayesian network learning techniques for a hybrid multi-objective bayesian estimation of distribution algorithm: a case study on mnk landscape. *Journal of Heuristics*, 27(4):549–573.
- Pelikan, M. (2010). Nk landscapes, problem difficulty, and hybrid evolutionary algorithms. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10*, pages 665–672, New York, NY, USA. Association for Computing Machinery.
- Price, K., Storn, R. M., and Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag, Berlin, Heidelberg.
- Sato, H., Aguirre, H., and Tanaka, K. (2013). Variable space diversity, crossover and mutation in moea solving many-objective knapsack problems. *Annals of Mathematics and Artificial Intelligence*, 68(4):197–224.
- Sato, H., Aguirre, H. E., and Tanaka, K. (2007). Local dominance and local recombination in moeas on 0/1 multiobjective knapsack problems. *European Journal of Operational Research*, 181(3):1708 – 1723.
- Spears, W. (2000). *Evolutionary Algorithms: The Role of Mutation and Recombination*. Natural Computing Series. Springer Berlin Heidelberg.
- Storn, R. and Price, K. (1995). Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 23.
- Syswerda, G. (1989). Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., and da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.