Difficult Features of Combinatorial Optimization Problems and the Tunable *W-Model* Benchmark Problem for Simulating them

Thomas Weise*
Institute of Applied Optimization
Faculty of Computer Science and Technology
Hefei University
Hefei, Anhui, China
tweise@hfuu.edu.cn

Zijun Wu
Institute of Applied Optimization
Faculty of Computer Science and Technology
Hefei University
Hefei, Anhui, China

ABSTRACT

The first event of the Black-Box Discrete Optimization Benchmarking (BB-DOB) workshop series aims to establish a set of example problems for benchmarking black-box optimization algorithms for discrete or combinatorial domains. In this paper, we *1)* discuss important features that should be embodied by these benchmark functions and *2)* present the *W-Model* problem which exhibits them. The *W-Model* follows a layered approach, where each layer can either be omitted or introduce a different characteristic feature such as neutrality via redundancy, ruggedness and deceptiveness, epistasis, and multi-objectivity, in a tunable way. The model problem is defined over bit string representations, which allows for extracting some of its layers and stacking them on top of existing problems that use this representation, such as OneMax, the Maximum Satisfiability or the Set Covering tasks, and the NK landscape. The ruggedness and deceptiveness layer can be stacked on top of any problem with integer-valued objectives. We put the *W-Model* into the context of related model problems targeting ruggedness, neutrality, and epistasis. We then present the results of a series of experiments to further substantiate the utility of the *W-Model* and to give an idea about suitable configurations of it that could be included in the BB-DOB benchmark suite.

CCS CONCEPTS

• **Mathematics of computing → Optimization with randomized search heuristics**; **Evolutionary algorithms**; *Probabilistic algorithms*; Sequential Monte Carlo methods; • **Theory of computation → Random search heuristics**; *Theory of randomized search heuristics*; • **Applied computing** → *Multi-criterion optimization and decision-making*;

KEYWORDS

Discrete Optimization, Benchmarking, Epistasis, Neutrality, Multi-Objective, Ruggedness

*Corresponding Author

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208240

ACM Reference Format:
Thomas Weise and Zijun Wu. 2018. Difficult Features of Combinatorial Optimization Problems and the Tunable *W-Model* Benchmark Problem for Simulating them. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205651.3208240

1 INTRODUCTION

The aim of the first Black-Box Discrete Optimization Benchmarking Workshop (BB-DOB@GECCO) workshop is to develop a standard methodology and problem set for the benchmarking of black-box optimization algorithms for discrete and combinatorial domains. With this paper, we make two contributions to this end:

(1) The goal of benchmarking is to get a complete picture of the strengths and weaknesses of optimization methods. We discuss a set of important problem features, which therefore should be represented in the set of BB-DOB benchmark problems.
(2) We then propose the *W-Model* problem, which can simulate these features in a layered, tunable way, for inclusion into the BB-DOB benchmark set. The layers of this model can also easily be combined with classical optimization problems.

Many real-world optimization tasks can be solved very efficiently with metaheuristics like Stochastic Local Search [17] and Evolutionary Algorithms (EAs) [44]. However, some frequently occurring problem characteristics cause difficulties for such algorithms [45, 48]. Some of the most important features that influence the problem hardness are ruggedness, neutrality, and deceptiveness in the fitness landscape as well as one of their causes, epistasis. The hardness of a problem further increases with the number of involved objective functions. In real applications, the influence of these features on the optimization process and their interactions with each other are often *a priori* unknown and complicated to measure.

A comprehensive set of benchmark functions for discrete optimization should include problems which exhibit these features in different strengths and in different combinations.

Many classical problems from operations research such as the Traveling Salesman Problem (TSP) [6, 46] or the Maximum Satisfiability Problem [1, 16] are not necessarily good choices for this purpose. One reason for this is that the hardness of these problems usually does not depend only on the "obvious" problem parameters such as the number of cities, clauses, or decision variables. An instance of the TSP, for example, is not necessarily hard just because it has a large number of cities. Regardless how many cities it has, if they could be arranged in a circle or equidistant grid, we can easily find the optimal solution. For many classical problems, complex statistics need to be computed in order to get an impression on whether a problem instance will be hard before trying to solve it [27].

The requirements that a good problem for the BB-DOB benchmark set should meet can be divided into non-functional aspects that increases its usability and functional properties which allow it to produce scientifically interesting results [11]. Let us first define the non-functional requirements:

(1) The objective function(s) should
   (a) have known ranges and
   (b) be easy and fast to compute.
(2) The optimal solutions should be known.
(3) A standard representation from the discrete domain should be used, such as bit strings or permutations.
(4) It should be possible to
   - create both easy and hard problem instances at small scales of the problem and
   - derive a problem instance entirely and deterministically derived from its parameters.

Having a fast-to-compute objective function with a known range and being able to represent different degrees of hardness within small-scale problems will allow to conduct many experiments in a short time as well as storing and processing candidate solutions in an efficient way. If the range of the objective value and the optimum are known, we have both easy ways to determine success or failure as well as to compare the performance on problems of different scale/range, because we can normalize the objective values.

The last requirement aims at increasing the reproducibility of experiments, which is currently a hot topic [18]. In the combinatorial domain problem instances are often specified in form of text files following certain formats, requiring a researcher to both have the paper *and* the problem instance files described in it. The latter one is not needed if our requirement is met.

Besides such features which increase the ease of use of the model, it should have the following properties from a "functional" (research) perspective:

(1) It should be well-motivated from the theoretical perspective and allow establish connections to results already existing in theoretical research or be theoretically tractable in way that allows new theoretical results.
(2) Its fitness landscape should exhibit features that are challenging for common metaheuristics, such as those discussed at the beginning of this introduction.
(3) Ideally, it should be possible to tune these features and study them both separately and in combination.
(4) The hardness of the problem should be determined directly by tunable parameters.

A benchmark problem meeting these requirements would be highly suitable for experiments, as it would allow researchers to discover and compare the mutual strengths and weaknesses of their algorithms. In this paper, we present *W-Model*, a tunable benchmark model which fulfills all the above functional and non-functional requirements. This problem, proposed in its original form by Weise et al. [47], allows for studying several characteristic fitness landscape features in a tunable way. It can be tackled both with fixed-length and variable-length bit string representations either in a single- or a multi-objective variant.

A problem proposed for the BB-DOB benchmark set should further meet the following availability criteria:

(1) It should be specified fully and reproducibly in the submitted paper.
(2) A reference implementation in one of the major programming languages should be provided as open source software on a publicly-available repository.
(3) Comprehensive utilities should provided to show that the reference implementation is equivalent to the definition in the paper and to allow for testing whether an alternate implementation fulfills the problem specification.
(4) Example experiments and results should be available.

We provide an open source reference implementation of the model in `Java` at http://github.com/thomasWeise/BBDOB_W_Model, including unit tests that allow for verifying the correctness of (possibly different) model implementations, an automatic experiment parallel execution environment, and an example experiment setup with some simple metaheuristics applied to the *W-Model*. We suggest that a set of configurations of this problem into the black-box discrete optimization benchmark suite.

In the following text, we first discuss features that can make an optimization problem hard *together* with their representation in the *W-Model* in Section 2. We then analyze the related work, i.e., benchmark problems which try to model (subsets of) similar features in Section 3. We then conduct an experimental study showing that the *W-Model* suitable to simulate arbitrary complex optimization problems correctly in Section 4. In 5, we summarize our research on the model.

2 DIFFICULT FEATURES AND MODEL DEFINITION

The *W-Model* defined by Weise et al. [44, 47] possesses tunable neutrality and redundancy, ruggedness and deceptiveness, epistasis, and multi-objectivity features. It is divided in distinct layers as sketched in Figure 1. These layers correspond to a step-by-step transformation of a bit string x to the objective value(s).

The baseline of the model problem is to find a bit string $x^\star$ = `0101010101010...01` of a predefined length n consisting of alternating zeros and ones.

This setup is very similar to the OneMax problem. While the goal of the OneMax problem is to find a bit string x of minimal Hamming distance $h(x, x^\star_{OM})$ to $x^\star_{OM} = (1111...)$, the goal under the *W-Model* is to minimize the Hamming distance $h(x, x^\star)$ to $x^\star = (0101...)$, as sketched in layer 5 of Figure 1. It can be expected that the extensive body of research on the OneMax problem [2, 9, 28, 40] would carry over to the baseline version of the *W-Model* problem. This objective function can be computed in $O(n)$.

While suitable for the search space of bit strings of length n, search spaces of variable-length bit strings can be facilitated as follows: Overly long strings are cut off after index n and the value

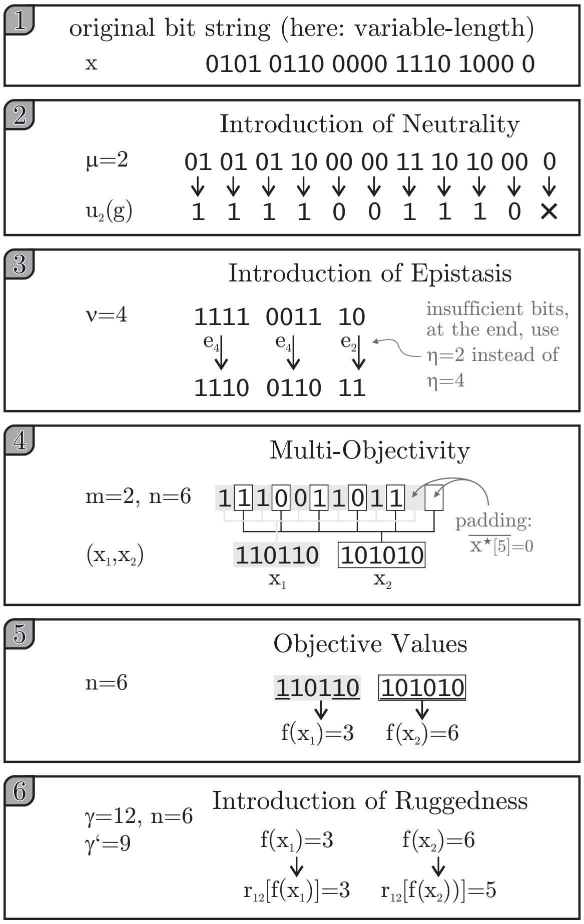


Figure 1: An example evaluation of a candidate solution for the *W-Model*.

$n - l(x)$ is added to the objective values for strings x whose length $l(x)$ is too short ($l(x) < n$).

2.1 Neutrality (layer 2)

The application of a search operator is *neutral* if it yields no change in objective space [7, 37]. It is challenging for optimization algorithms if the best candidate solution currently known is situated on a plane of the fitness landscape, i.e., if all adjacent solutions have the same objective values. The optimization algorithm then cannot find any gradient information[1] and thus there is no direction into which to proceed in a systematic manner. From the black-box point of view, each search operation will yield identical results.

Researchers in the late 1990s and early 2000s hoped that neutrality could increase the "evolvability" in an optimization process and may hence lead to better performance [7, 39, 41]. However, other works indicate that there may not be an advantage in random redundancy [23, 38], so especially uniform redundancy should always be avoided in representation design – but testing its impact may show how the optimization algorithms can deal with problems where redundancy is unavoidable.

A well-defined amount of neutrality can be generated in the *W-Model* through uniform redundancy in the search space, as sketched in layer **2** of 1. We therefore apply a trivial transformation u_μ that shortens the original bit string x by an integer factor $\mu \in 1 \ldots n$. The i^{th} bit in $u_\mu(x)$ is defined as 0 if and only if the majority of the μ bits starting at locus $i \cdot \mu$ in x is also 0, and as 1 otherwise. The default value 1 set in draw situations has (in average) no effect on the fitness, because the target solution $x^\star$ is defined as a sequence of alternating zeros and ones. If the length $l(x)$ of the bit string x is not a multiple of μ, the remaining $l(x) \bmod \mu$ bits are ignored. If $\mu = 1$, no neutrality as introduced. This transformation could be plugged on top of any bit-string based optimization problem and requires $O(n\mu)$ steps.

2.2 Epistasis (layer 3)

According to Lush [4, 26], the interaction between biological genes is epistatic if the effect on the fitness from altering one gene depends on the allelic state of other genes. Transposed to optimization, two decision variables (here: bits) can be said to interact epistatically, if the contribution of one of these variables to the objective value depends on the value of the other variable [4, 12, 31, 44].

Explicit epistasis is introduced in the *W-Model* as second transformation after the neutrality layer [44, 48]. A bijective function e_ν is defined, which translates a bit string x of length ν to a bit string $e_\nu(x)$ of the same length in $O\left(\nu^2\right)$ steps. Assume that we have two bit strings x_1 and x_2 which only differ in one single location, i.e., their Hamming distance $h(x_1, x_2)$ is one. e_ν leads to epistasis by exhibiting the following property:

$$h(x_1, x_2) = 1 \Rightarrow h(e_\nu(x_1), e_\nu(x_2)) \geq \nu - 1 \;\forall x_1, x_2 \in \{0,1\}^\nu \quad (1)$$

The meaning of Equation 1 is that a change of one bit in a bit string x leads to the change of at least $\nu-1$ bits in the corresponding mapping $e_\nu(x)$. This, as well as the demand for bijectivity, is provided if we define e_ν as in Equation 2, where we use both the binary and the two's complement natural number representation of the string x for simplicity:

$$e_\nu(x) = \begin{cases} e_\nu(x)[i] = \bigotimes_{\substack{\forall\, j \in \mathbb{N}_0 : 0 \leq j < \nu, \\ j \neq (i-1) \bmod \nu}} x[j] & \forall\; x : 0 \leq x < 2^{\nu-1} \\ \overline{e_\nu(x - 2^{\nu-1})} & \textit{otherwise} \end{cases} \quad (2)$$

In other words, for all strings $c \in \{0,1\}^\nu$ which have the most significant bit (MSB) not set, the e_ν transformation is performed bitwise. The i^{th} bit in $e_\nu(x)$ equals the exclusive-or combination of all but one bit in x. Hence, each bit in x influences the value of $\nu - 1$ bits in $e_\nu(x)$. For all strings x with 1 in the MSB, $e_\nu(x)$ is simply set to the negated e_ν transformation of x with the MSB cleared (the value of the MSB is $2^{\nu-1}$). This differentiation in e is needed in order to ensure its bijectiveness for even ν.

Bit strings of arbitrary length can be divided into consecutive blocks of the length ν and each of them is transformed separately with e_ν. If the length $l(x)$ of a given bit string x is no multiple of ν, the remaining $l(x) \bmod \nu$ bits at the end will be transformed

[1] The term "gradient" is a concept from continuous domains and we adopt it in a very loose way to discrete domains as a compact way of stating "direction in the search space where the objective values change (ideally improve)."

	z	e_4(z)		z	e_4(z)	z	e_4(z)
	0000	0000		1111	1110	0011	0110
h=1	0001	1101	h≥3	0111	0001	0101	1010
	0010	1011		1011	1001	0110	1100
	0100	0111		1101	0101	1001	0010
	1000	1111		1110	0011	1010	0100
						1100	1000

Figure 2: An example for the epistasis mapping $z \to e_4(z)$.

with the function $e_{l(x) \bmod \nu}$ instead of e_ν, as outlined in layer **3** of Figure 1.

The tunable parameter ν for the epistasis ranges from 2 to n leading to a complexity between $O(n)$ and $O(n^2)$. If m objective functions are specified (see next section), the string length grows to $n \cdot m$ and so does the valid range for ν. If ν is set to a value smaller than 3, no additional epistasis is introduced. Figure 2 outlines the mapping for $\nu = 4$.

This setup means that the interacting variables are all adjacent, which may or may not be a feature present in real-world problems. This property allows operations like single-point crossover to be functional. One could increase the hardness by first exchanging all bits according to a fixed permutation, which should be randomly selected before the experiment. Here we vote against this measure, since black-box optimization algorithms should not make assumptions about the relationship of decision variables based on their location in the representation anyway.

Besides the explicit epistasis introduced here, implicit epistasis can occur through the neutrality and ruggedness (see 2.4) mappings. To the best of our knowledge, it may not be possible to study these three effects completely separately, but with our model, well-dosed degrees of epistasis, neutrality, and ruggedness can separately or jointly generated. Of course, this epistasis transformation can again be plugged on top of any problem using binary string representations.

2.3 Multi-Objectivity (layer 4)

Many optimization problems are multi-objective, i.e., involve multiple, possible conflicting criteria [10, 13, 14]. A task with m objective functions is created in the original *W-Model* by interleaving m instances of the benchmark problem with each other and defining separate objective functions for each of them.

Instead of just dividing the candidate solution x in m blocks of length n, each standing for one objective, we scatter the objectives as illustrated in layer **4** of Figure 1. There, the bits for the first objective function comprise $x_1 = (x[0], x[m], x[2m], \dots)$, those used by the second objective $x_2 = (x[1], x[m+1], x[2m+1], \dots)$.

If a variable-length representation is used, superfluous bits (beyond the index range $0 \dots nm-1$) are ignored. If x is too short, the missing bits in the phenotypes are replaced with the complement from $x^\star$, i.e., if one objective misses the last bit (index $n-1$), it is padded with $\overline{x^\star[n-1]}$ which will worsen the objective by 1 on average.

No bit in x is used in more than one objective, so the optimization goals are orthogonal and unrelated. The objective functions of the *W-Model* will begin to conflict if epistasis ($\nu > 2$) is applied. Changing one bit in the candidate solution will then change the outcome of at most $\min\{\nu, m\}$ objectives. Some of them may improve while others may worsen.

2.4 Ruggedness and Deceptiveness (layer 6)

It is a general rule for representation design that it should exhibit (strong) causality [34, 35]. Small search steps should lead to small changes in the objective values. In rugged fitness landscapes, this is not the case: small changes in a candidate solution often cause large changes in its objective values. This makes it harder for an optimization algorithm to find and climb a gradient in objective space. Hand in hand with ruggedness goes *deceptiveness.* A region of the fitness landscape is deceptive if performing a gradient descend does not lead towards the optimal solution but instead away from it.

There are two (possibly interacting) sources of ruggedness and deceptiveness in a fitness landscape. The first one is the epistasis already modeled, since it generally violates causality. The other concerns the objective functions themselves, it lies in the nature of a problem. We introduce this type of ruggedness and deceptiveness *a posteriori* as a permutation r of the values from 0 to n which is applied to the objective values.

In an objective function with low total variation, the objective values of the neighboring candidate solutions are also neighboring. In the *W-Model* without epistasis ($\nu \leq 2$), for instance, two solutions differing in one bit will also differ by one in their objective values. We can write down the list of objective values the candidate solutions will take on if they would bit-wise be improved from the worst to the best possible configuration as $(n, n-1, \dots, 2, 1, 0)$. Exchanging two of the values in this list will create some artificial ruggedness. A measure for the ruggedness of such a permutation r is $\Delta(r) = \sum_{i=0}^{n-1} |r_i - r_{i+1}|$.

The original sequence of objective values has the minimum value n and the maximum possible value is $\hat{\Delta} = \frac{n(n+1)}{2}$. We can define permutations $r_{\gamma'}$ which are applied after the objective values are computed and which have the following features:

(1) They are bijective (since they are permutations).
(2) They must preserve the optimal value, i.e., $r_{\gamma'}[0] = 0$.
(3) $\Delta(r_{\gamma'}) = n + \gamma'$.

With $\gamma' \in 0 \dots (\hat{\Delta} - n)$, we can fine-tune the ruggedness. For $\gamma' = 0$, no ruggedness is introduced. For a given n, the permutations $r_{\gamma'}$ can be produced with the function `permutate` in Algorithm 1.

Algorithm 1 consists of two parts. `permutate` constructs permutations with increasing ruggedness measure γ'. As shown in [47] and Figure 3, using this transformation alone may lead to very deceptive problems at moderate levels of γ'. Hence, the permutations are re-arranged first using a second function `translate`, which ensures that the problem hardness smoothly increases from easy to rugged to deceptive and create permutations r_γ.

To illustrate this, all ruggedness permutations r_γ for an objective function defined over bit strings of length five (i.e., which can range from 0 to $n = 5$) are shown in Figure 3. As can be seen, the permutations scramble the objective function more and more with rising γ and reduce its gradient information, before producing gradients which actually point away from the optimum. The ruggedness transformation is sketched in layer **6** of Figure 1.

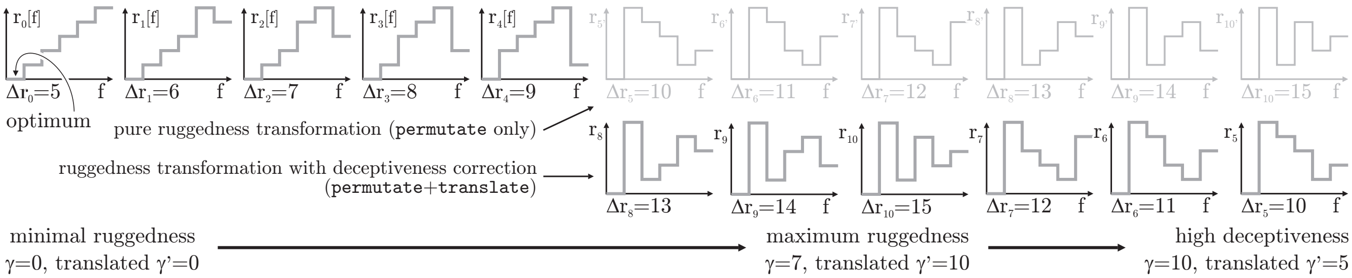


Figure 3: An example of the r_γ permutations produced by Algorithm 1 for $\gamma = 0..10$ and $q = 5$.

3 RELATED WORK

We now discuss problems based on fixed-length bit string representations which were defined in order to investigate problematic features such those discussed in the previous section.

In the late 1980s, Kauffman [20] defined the maybe most prominent member of this problem class, the *NK landscape* [20–22], a family of objective functions with tunable epistasis. We exemplarily describe it with slightly more details to give an impression of the general concept according to which such problems can be constructed. Each of the N bits x_i in a candidate solution of the *NK landscape* contributes one real value $f_{NK,i} : \{0, 1\}^{K+1} \mapsto [0, 1]$ to the objective function f_{NK}. $f_{NK,i}$ is determined the value of x_i and the values of K other bits $x_{i_1}, x_{i_2}, \ldots, x_{i_K}$ called the *neighbors* of x_i, i.e., we get

$$f_{NK}(x) = \frac{1}{N} \sum_{i=1}^{N} f_{NK,i}\left(x_i, x_{i_1}, x_{i_2}, \ldots, x_{i_K}\right) \tag{3}$$

Whenever the value of a bit changes, all the contributions of the bits to whose neighbor set it belongs will change too – to values uncorrelated to their previous state. While N describes the basic problem complexity, the intensity of this epistatic effect can be controlled with the parameter K: If $K = 0$, there is no epistasis at all. For $K = N - 1$ the epistasis is maximized and the fitness contribution of each gene depends on all other genes.

Weise [44] discusses a variety on research work analyzing the *NK landscape*, which did not allow modeling features such as neutrality or multi-objectivity – capabilities provided by the *W-Model*. Meanwhile, multi-objectivity is introduced in the MNK landscapes [3, 42]. It should be noted that the *W-Model* would allow using an *(M)NK landscape* on top of its neutrality transformation or in conjunction with the multi-objectivity mapping as a replacement of its epistasis and ruggedness transformations. The problems devised by Barnett [7], Geard et al. [15], Newman and Engelhardt [32] and Lobo et al. [24] can similarly be integrated into the *W-Model*. They extend the *NK landscapes* with neutrality features, which then could be studied together in the context of multi-objective optimization. The same holds for the *p-spin* model developed by Amitrano et al. [5], which can be considered as an alternative to the *NK fitness* landscape for tunable ruggedness [43].

The *Royal Road* functions developed by Mitchell et al. [29] are a set of special fitness landscapes for GAs. Platel et al. [36] combined them with Kauffman's *NK landscapes* and introduced the *Epistatic Road*. This landscape is significantly harder to construct and to tune than *W-Model* and – like the other related works – also has fewer capabilities.

The *ND family* of fitness landscapes has been developed by Beaudoin et al. [8] in order to provide a model problem with tunable neutrality. It also features deceptiveness via the internal use of trap functions. Yet, it cannot model multi-objectivity, ruggedness, or epistasis.

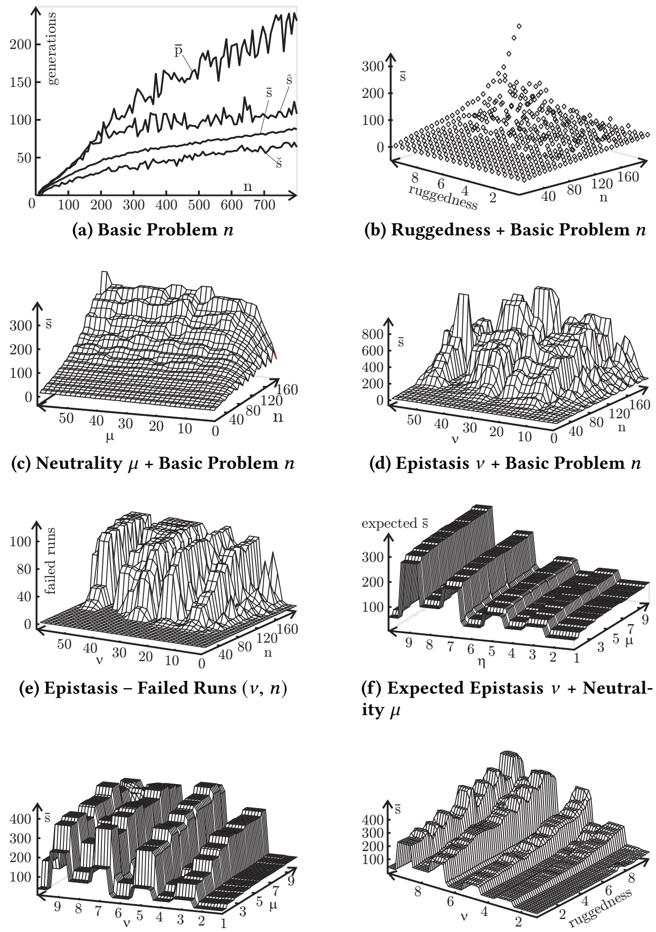
In [25], Lochtefeld and Ciarallo present an extension of the original version of the *W-Model*. Their TOP model aims to provide a more fine-grained objective convolution mechanism and it also applies two levels of ruggedness transformations. This extension has successfully been used to explore the relationship of problematic landscape features are related to the performance of multiobjectivization via helper objectives.

4 EXPERIMENTAL RESULTS

In order to verify whether this model suitably represents the features discussed, we have performed a comprehensive set of experiments [33] from which we will list the most significant results. These experiments were done in the framework of a Bachelor's thesis and are partially unpublished. They are based on an older implementation of the model and the variable-length representation (which we do not recommend for usage in BB-DOB), but can serve here to illustrate the features of our model problem.

In these experiments, we applied a standard multi-objective genetic algorithm with population size 1000, single-point crossover, single-bit mutation, and a variable-length bit string genome with a maximum string length of 8000 bits. In each test, we applied a non-functional objective minimizing the length of the strings. We suggest using these settings as default setup for all experiments involving our model in order to keep the results comparable. Furthermore, we have used tournament selection with tournament size 5 and Pareto ranking for fitness assignment. For each setting, at least 50 runs have been performed.

In the experiments, we distinguished between *success* and *perfection*. Success means finding individuals x of optimal *functional* fitness, i.e., $f(x) = 0$. Multiple such *successful* strings may exist, since superfluous bits at the end of genotypes do not influence their functional objective. The perfect string $x^\star$ has no such useless bits, it is the shortest possible solution with $f = 0$ and, hence, also optimal in the non-functional length criterion. We refer to the number of generations needed to find a successful individual as *success generations* s and to those needed to find the perfect solution as *perfection generations* p.



(a) Basic Problem n

(b) Ruggedness + Basic Problem n

(c) Neutrality μ + Basic Problem n

(d) Epistasis ν + Basic Problem n

(e) Epistasis – Failed Runs (ν, n)

(f) Expected Epistasis ν + Neutrality μ

(g) Real Epistasis ν + Neutrality μ

(h) Real Epistasis ν + Ruggedness

Figure 4: Experimental Results

4.1 The Basic Problem

In Figure 4a, we illustrate the basic problem complexity. The minimum, average, and maximum success generations $\check{s}$, $\bar{s}$, and $\hat{s}$ measured rise almost linearly after the basic problem parameter n has exceeded 300 bits. The average perfection generations $\overline{p}$ are much higher and rise faster, indicating that trimming down a solution to the minimum length is a complicated process.

4.2 Ruggedness

As outlined in Section 2.4, the number of ruggedness permutations r depends on the maximum objective values. Hence, it changes with the basic problem complexity. Furthermore, with the r permutation algorithms, also deceptive fitness landscapes will be created [47]. For visualization purposes, a scale from 0 to 10 for ruggedness and for deceptiveness were used in [33], separating and ordering the two characteristics.

In Figure 4b, the average generations needed for finding a successful individual have been plotted against the basic problem complexity n and the ruggedness according to this scale. Apart from a few peaks in the diagram occurring for $n > 70$, the problem hardness, as expected, increases very fast with the ruggedness.

4.3 Neutrality

The redundancy-based neutrality in our model exhibits a rather interesting behavior illustrated in 4c. Until a degree of $\mu \approx 10$, the problems rapidly gets harder. From there on, a further increase of μ only leads to a very slow increase in hardness. The reason for this behavior is rooted in the crossover operations. If crossover is present, it seemingly plays no role whether 10, 20, or even more bits of the genotype determine the single phenotypic bits. To prove this, the experiments were repeated with lower crossover rates. Then, $\bar{s}$ increases much faster and also becomes unsolvable (in 1000 generations) very early. For the BB-DOB benchmark suite, we suggest using values $\mu \in 1 \dots 4$.

4.4 Epistasis

The behavior of the epistasis model component is as interesting as that of the neutrality layer. Figure 4d shows that the problem complexity steeply increases with rising values of ν. This becomes even more obvious when comparing with Figure 4e, where the number of experimental runs (out of 100) are plotted that were not able to find a successful individual up to the 1000-generation limit. Both graphs, however, have also deep incisions at locations where ν takes on values of the form $2 + 4v : v \in \mathbb{N}$. Such epistasis settings lead to significantly easier problems, which can be explained by the nature of the epistatic mapping e_ν – it decreases the Hamming distance of elements x_1, x_2 which have originally $h(x_1, x_2) = \nu/2$ for such values [33]. For inclusion in the BB-DOB benchmark suite, we therefore suggest to only use ν values that are not such multiples.

4.5 Epistasis and Neutrality

The usefulness of our model problem stands and falls with the ability to combine the different features introduced in Section 2. Therefore, we ran multiple test experiments with a fixed problem size $n = 80$. One of them was to check how the epistasis and neutrality interact in the model. Therefore, we have simply added up the previous two experiments (sketched in Figure 4f) and compared these "expected $\bar{s}$" with results from real experiments with the same parameter settings. The results, depicted in Figure 4g, meet the expectations almost exactly in terms of the problem structure, while exhibiting an almost constant quantitative offset of about 100 generations.

4.6 Epistasis and Ruggedness

In Figure 4h, we have plotted an experiment series which combines ruggedness and epistasis. The outcomes of these experiments are very similar to the expected results when adding up Figure 4b and Figure 4d for $n = 80$. A rising ruggedness component leads, however, to over-proportional increases in $\bar{s}$. This may be due to epistasis making optimization complicated because it leads to ruggedness in the fitness landscape. By introducing additional ruggedness in the objective functions (which is what we are doing in this series), resonance like fish tailing seems to result.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have discussed requirements that a good benchmark problem for the BB-DOB suite should exhibit. On the functional side, the benchmark problem should allow investigating different fitness landscape features separately and in combination. Then, researchers can explore the mutual advantages and disadvantages of their algorithms. Non-functional requirements such as low complexity, ease of understanding, and easy replication of experiments should increase the usability of the problem.

We showed that the *W-Model* meets all of these requirements. We provide a Java implementation of this model problem at http://github.com/thomasWeise/BBDOB_W_Model along with unit test for verifying other implementations, an automated parallel experimentation environment, and example experiments.

We then presented some results from experiments with the *W-Model* problem [33, 47]. We have shown that our model is not only simple and easily tangible from a theoretical point of view, but also exhibits a behavior which meets our expectations in experiments.

We suggest to apply a set of specific single-objective, fixed-length representation settings of the model problem in the framework of the BB-DOB benchmark suite. While we are still researching good settings for the model parameters, we, for now, propose using

(1) a selection of values of n ranging from 10 to 64,
(2) all values of $\mu \in 1\ldots3$,
(3) values of ν which are not of the form $2 + 4v$ and are close to $2 + ((n-2) \cdot i/10)$ for $i \in 0\ldots10$ together with powers of 2 and 10, and
(4) values of γ which are in $n(n-1) \cdot i/20$ for all $i \in 0\ldots10$ together with powers of 2 and 10.

These settings should lead to a set of well-reproducible problems that cover a wide range of difficulties, from very easy (i.e., OneMax) to highly epistatic and rugged landscapes with neutral plateaus. We believe that establishing the *W-Model* as component of the BB-DOB benchmark can help researchers to evaluate optimization algorithms in different situations in an unbiased manner.

We thank the reviewers for pointing out that classical hardness measures on which *W-Model* is conceptually built are known to not be perfect and potentially misleading [19, 30]. Hence, more research is necessary and one part of our future work is to learn more about the impact of the model settings on the optimization process. By conducting further experiments, we will attempt to collect more empiric data on how the features of the fitness landscape influence the success probability of optimization.

ACKNOWLEDGEMENTS

Dr. Weise acknowledges support by the National Natural Science Foundation of China under Grants 61673359, 61150110488, and 71520107002.

REFERENCES

[1] 2015. *Proc. of the 18th Intl. Conf. on Theory and Applications of Satisfiability Testing (SAT 2015)* (Austin, TX, USA). Lecture Notes in Computer Science book series (LNCS), Vol. 9340. Springer, Cham. https://doi.org/10.1007/978-3-319-24318-4
[2] David H. Ackley. 1987. *A Connectionist Machine for Genetic Hillclimbing.* Ph.D. Dissertation. Carnegy Mellon University (CMU), Pittsburgh, PA, USA.
[3] Hernán E. Aguirre and Kiyoshi Tanaka. 2007. Working Principles, Behavior, and Performance of MOEAs on MNK-Landscapes. *European Journal of Operational Research* 181 (2007), 1670–1690. Issue 3. https://doi.org/10.1016/j.ejor.2006.08.004
[4] Lee Altenberg. 1997. NK Fitness Landscapes. In *Handbook of Evolutionary Computation*, Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz (Eds.). Oxford University Press, Oxford, England, UK, Chapter B2.7.2.
[5] Ciro Amitrano, Luca Peliti, and M. Saber. 1989. Population Dynamics in a Spin-Glass Model of Chemical Evolution. *Journal of Molecular Evolution* 29, 6 (1989), 513–525.
[6] David Lee Applegate, Robert E. Bixby, Vašek Chvátal, and William John Cook. 2007. *The Traveling Salesman Problem: A Computational Study.* Princeton University Press, Princeton, NJ, USA.
[7] Lionel Barnett. 1998. Ruggedness and Neutrality – The NKp Family of Fitness Landscapes. In *Proc. of the Sixth Intl. Conf. on Artificial Life (Artificial Life VI)* (Los Angeles, CA, USA) *(Complex Adaptive Systems)*, Vol. 6. MIT Press, Cambridge, MA, USA, 18–27.
[8] William Beaudoin, Sébastien Vérel, Philippe Collard, and Cathy Escazut. 2006. Deceptiveness and Neutrality the ND Family of Fitness Landscapes. In *8th Conf. on Genetic and Evolutionary Computation.* ACM Press, Seattle, WA, USA, 507–514.
[9] Tobias Blickle and Lothar Thiele. 1995. *A Comparison of Selection Schemes Used in Genetic Algorithms* (2 ed.). TIK-Report 11. ETH Zürich, Department of Electrical Engineering, Computer Engineering and Networks Laboratory (TIK), Zürich, Switzerland. ftp://ftp.tik.ee.ethz.ch/pub/publications/TIK-Report11.ps
[10] Carlos Artemio Ceollo Coello. 1999. An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends. In *Congress on Evolutionary Computation (CEC)* (Washington, DC, USA). IEEE Press, Piscataway, NJ, USA, 3–13. https://doi.org/10.1109/CEC.1999.781901
[11] Lianping Chen, Muhammad Ali Babar, and Bashar Nuseibeh. 2013. Characterizing Architecturally Significant Requirements. *IEEE Software* 30 (March–April 2013), 38–45. Issue 2. https://doi.org/10.1109/MS.2012.174
[12] Yuval Davidor. 1990. Epistasis Variance: A Viewpoint on GA-Hardness. In *Proc. of the First Workshop on Foundations of Genetic Algorithms (FOGA'90)* (Bloomington, IN, USA). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 23–35.
[13] Kalyanmoy Deb. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons Ltd., New York, NY, USA.
[14] Carlos M. Fonseca and Peter J. Fleming. 1998. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part I: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* 28, 1 (1998), 26–37.
[15] Nicholas Geard, Janet Wiles, Jennifer Hallinan, Bradley Tonkes, and Benjamin Skellett. 2002. A Comparison of Neutral Landscapes – NK, NKp and NKq. In *Proc. of the IEEE Congress on Evolutionary Computation (CEC'02), IEEE World Congress on Computation Intelligence (WCCI'02).* IEEE, Honolulu, HI, USA, 205–210.
[16] Holger H. Hoos and Thomas Stützle. 2000. SATLIB: An Online Resource for Research on SAT. In *SAT2000 – Highlights of Satisfiability Research in the Year 2000 (Frontiers in Artificial Intelligence and Applications)*, Vol. 63. IOS Press, Amsterdam, The Netherlands, 283–292. http://www.cs.ubc.ca/~hoos/Publ/sat2000-satlib.pdf
[17] Holger H. Hoos and Thomas Stützle. 2005. *Stochastic Local Search: Foundations and Applications.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
[18] Matthew Hutson. 2018. Missing Data Hinder Replication of Artificial Intelligence Studies. *Science* (Feb. 15, 2018). https://doi.org/10.1126/science.aat3298
[19] Thomas Jansen. 2001. On Classifications of Fitness Functions. In *Theoretical Aspects of Evolutionary Computing*, Leila Kallel, Bart Naudts, and Alex Rogers (Eds.). Springer, Berlin, Heidelberg, 371–385. https://doi.org/10.1007/978-3-662-04448-3_18
[20] Stuart Alan Kauffman. 1988. Adaptation on rugged fitness landscapes. In *Lectures in the Sciences of Complexity: The Proc. of the 1988 Complex Systems Summer School* (Santa Fé, NM, USA) *(Santa Fé Institue Studies in the Sciences of Complexity)*, Vol. Lecture I. Addison-Wesley Longman Publishing, Boston, MA, USA, 527–618.
[21] Stuart Alan Kauffman and Simon Levin. 1987. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology* 128, 1 (1987), 11–45.
[22] Stuart Alan Kauffman and Edward D. Weinberger. 1989. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology* 141 (1989), 211–245. Issue 2.
[23] Joshua D. Knowles and Richard A. Watson. 2002. On the Utility of Redundant Encodings in Mutation-based Evolutionary Search. In *Proc. of the 7th Intl. Conf. on Parallel Problem Solving from Nature (PPSN VII)* (Granada, Spain) *(Lecture Notes in Computer Science (LNCS))*, Vol. 2439. Springer-Verlag GmbH, Berlin, Germany, 88–98. https://doi.org/10.1007/3-540-45712-7_9
[24] José Lobo, John H. Miller, and Walter Fontana. 2004. *Neutrality in Technological Landscapes.* Working Papers. Santa Fe Institute, Santa Fé, NM, USA.
[25] Darrell F. Lochtefeld and Frank W. Ciarallo. 2012. Multiobjectivization via Helper-Objectives with the Tunable Objectives Problem. *IEEE Transactions on Evolutionary Computation* 16 (2012). Issue 3. https://doi.org/10.1109/TEVC.2011.2136345
[26] Jay L. Lush. 1935. Progeny Test and Individual Performance as Indicators of an Animal's Breeding Value. *Journal of Dairy Science (JDS)* 18, 1 (1935), 1–19.
[27] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Markus Wagner, Jakob Bossek, and Frank Neumann. 2013. A Novel Feature-based Approach to Characterize Algorithm Performance for the Traveling Salesperson Problem. *Annals*

of Mathematics and Artificial Intelligence 69, 2 (Oct. 2013), 151–182. https://doi.org/10.1007/s10472-013-9341-2

[28] Brad L. Miller and David Edward Goldberg. 1996. Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise. *Evolutionary Computation* 4, 2 (1996), 113–131. https://doi.org/10.1162/evco.1996.4.2.113

[29] Melanie Mitchell, Stephanie Forrest, and John Henry Holland. 1991. The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. In *Toward a Practice of Autonomous Systems: Proc. of the First European Conf. on Artificial Life (ECAL'91)* (Paris, France). MIT Press, Cambridge, MA, USA, 245–254.

[30] Bart Naudts and Leila Kallel. 2000. A Comparison of Predictive Measures of Problem Difficulty in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* 4 (April 2000), 1–15. Issue 1. https://doi.org/10.1109/4235.843491

[31] Bart Naudts and Alain Verschoren. 1996. Epistasis On Finite And Infinite Spaces. In *Proc. of the 8th Intl. Conf. on Systems Research, Informatics and Cybernetics (InterSymp'96)* (Baden-Baden, Germany). Intl. Institute for Advanced Studies in Systems Research and Cybernetic (IIAS), Tecumseh, ON, Canada, 19–23.

[32] Mark E. J. Newman and Robin Engelhardt. 1998. Effect of Neutral Selection on the Evolution of Molecular Species. *Proc. of the Royal Society B: Biological Sciences* 256, 1403 (1998), 1333–1338.

[33] Stefan Niemczyk. 2008. *Ein Benchmark Problem für Globale Optimierungsverfahren.* Bachelor's thesis. Distributed Systems Group, University of Kassel. Supervisor: Thomas Weise.

[34] Charles Campbell Palmer. 1994. *An approach to a problem in network design using genetic algorithms.* Ph.D. Dissertation. Polytechnic University, New York, NY.

[35] Charles Campbell Palmer and Aaron Kershenbaum. 1994. Representing Trees in Genetic Algorithms. In *Proc. of the 1st IEEE Conf. on Evolutionary Comput. (CEC'94)* (Orlando, FL, USA), Vol. 1. IEEE Comp. Soc., Piscataway, NJ, USA, 379–384.

[36] Michaël Defoin Platel, Sébastien Vérel, Manuel Clergue, and Philippe Collard. 2003. From Royal Road to Epistatic Road for Variable Length Evolution Algorithm. In *Proc. of the 6th Intl. Conf. on Artificial Evolution, Evolution Artificielle (EA'03)* (Marseilles, France) *(Lecture Notes in Computer Science (LNCS))*, Vol. 2936. Springer-Verlag GmbH, Berlin, Germany, 3–14. https://doi.org/10.1007/b96080

[37] Christian M. Reidys and Peter F. Stadler. 2001. Neutrality in Fitness Landscapes. *Journal of Applied Mathematics and Computation* 117, 2–3 (2001), 321–350. https://doi.org/10.1016/S0096-3003(99)00166-6

[38] Franz Rothlauf. 2006. *Representations for Genetic and Evolutionary Algorithms* (second ed.). Studies in Fuzziness and Soft Computing, Vol. 104. Springer-Verlag, Berlin/Heidelberg. https://doi.org/10.1007/3-540-32444-5

[39] Rob Shipman. 1999. Genetic Redundancy: Desirable or Problematic for Evolutionary Adaptation?. In *Proc. of the 4th Intl. Conf. on Artificial Neural Nets and Genetic Algorithms (ICANNGA'99)* (Protorož, Slovenia). Springer-Verlag GmbH, Berlin, Germany, 1–11.

[40] Dirk Thierens and David Edward Goldberg. 1994. Convergence Models of Genetic Algorithm Selection Schemes. In *Proc. of the Third Conf. on Parallel Problem Solving from Nature (PPSN III)* (Jerusalem, Israel) *(Lecture Notes in Computer Science (LNCS))*, Vol. 866/1994. Springer-Verlag GmbH, Berlin, Germany, 119–129. https://doi.org/10.1007/3-540-58484-6_256

[41] Marc Toussaint and Christian Igel. 2002. Neutrality: A Necessity for Self-Adaptation. In *Proc. of the IEEE Congress on Evolutionary Computation (CEC'02)* (Honolulu, HI, USA). IEEE Computer Society Press, Los Alamitos, CA, USA, 1354–1359. https://doi.org/10.1109/CEC.2002.1004440

[42] Sébastien Verel, Arnaud Liefooghe, Laetitia Jourdan, and Clarisse Dhaenens. 2013. On the Structure of Multiobjective Combinatorial Search Space: MNK-Landscapes with Correlated Objectives. *European Journal of Operational Research* 227 (2013), 331–342. Issue 2. https://doi.org/10.1016/j.ejor.2012.12.019

[43] Edward D. Weinberger and Peter F. Stadler. 1993. Why Some Fitness Landscapes are Fractal. *Journal of Theoretical Biology* 163, 2 (1993), 255–275.

[44] Thomas Weise. 2009. *Global Optimization Algorithms – Theory and Application.* it-weise.de (self-published), Germany. http://www.it-weise.de/projects/book.pdf

[45] Thomas Weise, Raymond Chiong, and Ke Tang. 2012. Evolutionary Optimization: Pitfalls and Booby Traps. *Journal of Computer Science and Technology (JCST)* 27 (2012), 907–936. Issue 5. https://doi.org/10.1007/s11390-012-1274-4

[46] Thomas Weise, Raymond Chiong, Ke Tang, Jörg Lässig, Shigeyoshi Tsutsui, Wenxiang Chen, Zbigniew Michalewicz, and Xin Yao. 2014. Benchmarking Optimization Algorithms: An Open Source Framework for the Traveling Salesman Problem. *IEEE Computational Intelligence Magazine (CIM)* 9, 3 (Aug. 2014), 40–52. https://doi.org/10.1109/MCI.2014.2326101

[47] Thomas Weise, Stefan Niemczyk, Hendrik Skubch, Roland Reichle, and Kurt Geihs. 2008. A Tunable Model for Multi-Objective, Epistatic, Rugged, and Neutral Fitness Landscapes. In *Genetic and Evolutionary Computation Conference.* ACM Press, Atlanta, GA, USA, 795–802.

[48] Thomas Weise, Michael Zapf, Raymond Chiong, and Antonio Jesús Nebro Urbaneja. 2009. Why is optimization difficult? In *Nature-Inspired Algorithms for Optimisation*, Raymond Chiong (Ed.). Studies in Computational Intelligence, Vol. 193. Springer-Verlag, Berlin/Heidelberg, Chapter 1, 1–50. https://doi.org/10.1007/978-3-642-00267-0_1

Algorithm 1: $r_\gamma \longleftarrow$ `build_permutation`(γ, n)
This algorithm is a corrected version compared to [44, 47].

Input: n: the maximum objective value
Input: γ: the γ value for tuning the ruggedness, with $\gamma \in 0\ldots\frac{1}{2}n(n-1)$ value
Data: i, j, k: counter variables
Data: $start, max, upper$: computed values
Data: γ': the translated version of γ
Output: r_γ: the permutation r_γ for re-arranging objective values

```
begin
    return permutate(translate(γ), n)

sub-algorithm r ⟵ permutate(γ', n)
    r ⟵ allocate integer array of length n + 1
    max ⟵ ⌊(1/2) n(n − 1)⌋
    if γ' ≤ 0 then start ⟵ 0
        else start ⟵ n − 1 − ⌊1/2 + sqrt(1/4 + 2(max − γ'))⌋
    k ⟵ 0
    for j ⟵ 1 up to start − 1 do
        if j is odd then r[j] ⟵ n − k
        else
            k ⟵ k + 1
            r[j] ⟵ k
    for j ⟵ start up to n do
        k ⟵ k + 1
        if start is odd then r[j] ⟵ n − k
        else r[j] ⟵ k
    upper ⟵ (γ' − max) + (1/2)(n − start − 1)(n − start)
    j ⟵ start
    for i ⟵ 1 up to upper do
        j ⟵ j − 1
        swap r[j] and r[n]
    return r

sub-algorithm γ' ⟵ translate(γ, n)
    if γ ≤ 0 then return 0
    l ⟵ n(n−1)/2
    i ⟵ ⌊n/2⌋ · ⌊(n+1)/2⌋
    if γ ≤ i then
        j ⟵ ⌊(n+2)/2 − sqrt(n²/4 + 1 − γ)⌋
        k ⟵ γ − j(n + 2) + j² + n
        return k + 2(j(n + 2) − j² − n) − j
    else
        j ⟵ ⌊((n mod 2)+1)/2 + sqrt((1−(n mod 2))/4 + γ − 1 − i)⌋
        k ⟵ γ − (j − (n mod 2))(j − 1) − 1 − i
        return l − k − 2j² + j − (n mod 2)(−2j + 1)
```