Parametrizing Convection Selection: Conclusions from the Analysis of Performance in the NKq Model

Maciej Komosinski
Poznan University of Technology
Poznan, Poland
maciej.komosinski@cs.put.poznan.pl

Konrad Miazga
Poznan University of Technology
Poznan, Poland
konrad.miazga@cs.put.poznan.pl

ABSTRACT

Convection selection in evolutionary algorithms is a method of splitting the population into subpopulations based on the fitness values of solutions. Convection selection was previously found to be superior to standard selection techniques in difficult tasks of evolutionary design. However, reaching its full potential requires tuning of parameters that affect the performance of the evolutionary search process. Performing experiments on benchmark fitness functions does not provide general knowledge required for such tuning. Therefore, in order to gain an insight into the link between the characteristics of the fitness landscape, the parameters of the selection technique, and the quality of the best found solutions, we perform an analysis based on the NKq model of rugged fitness landscapes with neutrality. As a result, we identify several rules that will help researchers and practitioners of evolutionary algorithms adjust the values of convection selection parameters based on the knowledge of the properties of a given optimization problem.

CCS CONCEPTS

• **Computing methodologies → Search methodologies**;

KEYWORDS

evolutionary algorithms, convection selection, NKq model, fitness landscape

ACM Reference Format:
Maciej Komosinski and Konrad Miazga. 2019. Parametrizing Convection Selection: Conclusions from the Analysis of Performance in the NKq Model. In *Genetic and Evolutionary Computation Conference (GECCO '19), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3321707.3321864

1 INTRODUCTION

In recent papers, convection selection (a meta-selection scheme first introduced in [14]) was shown to exhibit superior performance when compared to both a classical, single population evolutionary algorithm [12], and a multipopulational island model evolutionary algorithm [11]. These comparative experiments were performed on difficult tasks of evolutionary design and on selected CEC2005 benchmark functions [19]. The comparisons were parameter-agnostic, i.e., conclusions were based not on arbitrarily chosen parametrizations of these algorithms, but rather on the best parametrizations found across a wide range of possibilities.

As expected, it was simultaneously shown that performance of convection selection can depend on the selected values of parameters of this technique, and no single combination of parameters can serve as a silver bullet which would fit all problems. As convection selection is related to the island model, its parameters include the number of subpopulations and the frequency of communication between them. Similar selection techniques include FUSS [16] which is virtually parameter-free, and much more sophisticated HFC [21], both of which are discussed later in more detail.

It holds under the No Free Lunch Theorem [22] that incorporating the knowledge about the structure of the problem into the algorithm, which can be achieved by tuning its parameters, can improve the performance of this algorithm. It is therefore crucial to obtain knowledge about the correlations between the parameters of the algorithm, the characteristics of the fitness landscape, and the performance of the algorithm in a given optimization problem. In order to learn about such correlations, it is necessary to evaluate the performance of the algorithm on a set of problems that vary in their characteristics in a controllable way. To this end, we performed experiments on a set of problems generated with the NKq model of fitness landscapes [4], which extends Kauffman's NK model [9] with neutrality – a property common to many difficult optimization tasks.

2 CONVECTION SELECTION

Convection selection – as presented in Algorithm 1 – is a technique in evolutionary algorithms where the population is split into subpopulations according to fitness values of solutions [11, 12, 14]. Evolution proceeds independently in each subpopulation for $M \cdot S \cdot R$ evaluations in total; in a single-threaded implementation of convection selection, the currently processed subpopulation changes after every evaluation, round-robin style. Every $M \cdot S \cdot R$ evaluations, all M subpopulations, each of size S, are merged together to create a full population consisting of $M \cdot S$ solutions, which is then split again into M subpopulations according to one of two policies, i.e., *EqualWidth* or *EqualNumber*:

- *EqualWidth*. Each subpopulation is assigned a range of fitness values of *equal width*, i.e., the i-th population is assigned to range $\left[f_{min} + (i-1) \cdot \frac{f_{max}-f_{min}}{M}; f_{min} + i \cdot \frac{f_{max}-f_{min}}{M}\right]$, where $i = [1..M]$, and f_{min} and f_{max} are respectively the

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6111-8/19/07…$15.00
https://doi.org/10.1145/3321707.3321864

lowest and the highest fitness value in the current population (assuming a maximization problem). Whenever the full population is split, each solution ends up in a subpopulation assigned to its corresponding fitness value. Should any subpopulation be empty, it is filled with cloned solutions from the worse subpopulation.

- *EqualNumber.* Each subpopulation is assigned a range of fitness ranks within the population of *equal width*, i.e., the i-th population is assigned to range $((M-i)\cdot S; (M-i+1)\cdot S]$. When the full population is split, each solution ends up in a subpopulation assigned to its corresponding rank in the full population.

Whenever any subpopulation exceeds its size limit, the minimal amount of solutions required to meet the size requirements is removed from that subpopulation at random.

Algorithm 1: The pseudocode of the Convection Selection algorithm.

```
function convectionSelection (M, S, R);
Input: subpopulation number M, subpopulation size S,
       migration period multiplier R
count = 0;
pop = initialize(size = M · S);
subpops = split(pop, number = M);
while not stopping-condition do
    count += 1;
    currentpop = subpops[count  mod  M];
    // Fill the empty subpopulations with solutions from the
       first worse and not empty subpopulation
    i=1;
    while size(currentpop) == 0 do
        currentpop = copy(subpops[(count−i)  mod  M]);
        i += 1;
    end
    child = generate_solution(currentpop);
    currentpop.add(child);
    subpops[count  mod  M] = currentpop;
    // Every M*S*R evaluations, update the split into
       subpopulations
    if count  mod  (M · S · R) == 0 then
        pop = merge(subpops);
        subpops = split(pop, number = M) ; // EqualWidth or
          EqualNumber used here
    end
    // Trim the size of each subpopulation to S
    for subpop in subpops do
        if size(subpop) > S then
            subpop = trim(subpop, size = S);
        end
    end
end
```

The idea behind convection selection comes from a simple observation: in difficult problems, it often happens that in order for a solution to improve, it must first pass a valley in the fitness landscape that corresponds to deterioration in its quality. Classical selection schemes offer monotonic selective pressure, which discourages the algorithm from investing computation time in testing solutions which are significantly worse than the rest of the population. This in turn leads to a premature convergence to a sub-par local optimum. In order to avoid that problem, the algorithms most often employ some mechanism in order to increase the diversity of solutions, which should facilitate escaping from local optima. Some of the more popular examples of such mechanism are fitness sharing [5, 7] and crowding [2, 6, 17], none of which however directly facilitates crossing valleys in a fitness landscape.

There exist already some modifications of evolutionary algorithms whose purpose is to spread the selective pressure between different levels of fitness, most notable being Fitness Uniform Optimization (FUSS) [16] and Hierarchical Fair Competition (HFC) [21]. Both of these algorithms share similarities with convection selection, however there are also some meaningful differences between them. While FUSS uses a single population of solutions, convection selection supports a number of subpopulations coexisting simultaneously. Additionally, whereas FUSS has no direct way to introduce and control selection pressure in the population, convection selection can be coupled with any traditional selection scheme (a modified version of FUSS, FUDS [8], does however allow such control).

Comparing convection selection with HFC also reveals a number of differences. Although both HFC and convection selection maintain a number of fitness-based subpopulations, they differ in a way solutions can migrate between these subpopulations. While in convection selection each subpopulation evolves fully independently between the merge/spilt events, in HFC solutions that exceed some threshold fitness level are moved from a population into a buffer, where they wait for the next migration event. Additionally, solutions in HFC can only migrate "upwards" in the chain of subpopulations, while the algorithm fills the weakest subpopulation on a regular basis with randomly generated new solutions. These differences suggest that while convection selection focuses on facilitating the traversal of valleys in the fitness landscape, HFC focuses on continuously incorporating new genetic material into the population.

3 NKQ MODEL OF FITNESS LANDSCAPES

The NK model of fitness landscapes was first introduced by Stuart Kauffman in the context of theoretical biology – specifically, the maturation of the immune response [9]. Kauffman's model describes a family of rugged fitness landscapes defined by two parameters N and K. The parameter N defines the size of the problem (and therefore the dimensionality of the landscape), while the parameter K defines the degree of epistasis present in the problem (and therefore the ruggedness of the landscape). The fitness value f for any point in the landscape (i.e., for any solution of the problem) can be computed as a sum of N fitness contributions (value of each depending on a locus-specific multiplexer-like function f_i), a gene at the given locus, and genes at $K-1$ other associated loci.

Although it was first introduced in the context of biology, due to its universal structure the NK model was quickly adopted by researchers in the fields of evolutionary algorithms. Because of its simplicity, clear mathematical structure, and the fact that – in general – decision versions of optimization problems based on the NK model are NP-complete [23], the NK model is often used as a benchmark. It is also used as an object of theoretical analysis in hopes of gaining knowledge about general characteristics of fitness landscapes [10, 20] and the ways the optimization algorithms search through these landscapes [18].

Despite its popularity, it has been noticed that the NK model does not capture an important feature of many fitness landscapes – both biological, and those based on optimization problems – namely, neutrality. It has been shown that the presence of neutrality in a genotype-phenotype mapping increases evolvability of the population [3]. Such partially neutral fitness landscapes contain structures such as plateaus or neutral networks, which allow populations to diversify their genetic material through random genetic drift while simultaneously maintaining high fitness. This facilitates the discovery of new, beneficial mutations, helps population avoid becoming stuck in a local optimum, and improves its adaptation to changes in the environment.

In an effort to incorporate the phenomenon of neutrality to Kauffman's NK model, extensions of the model were presented, most notably NKq [4] and NKp [1] models. The NKq model [4] introduces neutrality by quantizing contributions of each gene to q distinct levels. In effect, the fitness landscape becomes more terraced. The NKp model [1] sets the contribution of random genes to zero with a probability p. This also results in a more neutral landscape, however its characteristics are more extreme than in the case of NKq – in particular, a significant fraction of mutations becomes deleterious, especially for high values of p, and when approaching best fitness values [4]. Because of that, the neutrality present in the NKp model might have more ambiguous effect on the evolvability of the population. This is why we decided to perform experiments using the NKq model of fitness landscapes. Still, results obtained with the NKp model would be very interesting, especially since the NKp model may have similar characteristics to the previously considered difficult tasks of evolutionary design [13].

4 EXPERIMENTAL SETUP

The experiments were implemented and carried out in the Framsticks environment [15]. We have performed 15 independent evolutionary runs with 1 000 000 evaluations each, for every combination of the parameter values enumerated below. This resulted in 21 870 independent evolutionary runs in total.
Evolutionary algorithm parameters were:

- tournament size $t \in \{2, 5\}$,
- subpopulation number $M \in \{5, 10, 25\}$,
- subpopulation size $S \in \{10, 20, 50\}$,
- migration period multiplier $R \in \{2, 10, 50\}$.

NKq model parameters were:

- genotype length $N \in \{20, 50, 100\}$,
- the degree of epistasis $K \in \{2, 5, 8\}$,
- the number of available levels of contribution $q \in \{2, 10, \infty\}$, where ∞ means continuous, non-quantized values of fitness contributions.

We have used the implementation of convection selection as described in Sect. 2, with the addition of elitism. When performing random deletion within a subpopulation, we always ensured that the single best solution from that subpopulation is never deleted. Such elitism was used because initial experiments demonstrated that it can improve the performance of the search process, with an added benefit of guaranteeing a non-decreasing fitness of the best solution in the population.

Half of the new offspring solutions were created by the mutation operation, and half of them resulted from the crossover operator. When applying a mutation operator to a genotype, we guaranteed that the value of at least one bit was flipped. After flipping the first (guaranteed) bit, every other bit in the genotype was flipped with the probability of $\frac{1}{N}$, where N was the length of the genotype.

In our experiments, the adjacent version of the NKq model was used in which fitness contribution linked with a locus depended on the value of the gene at that locus and K *following* loci. Although in evolutionary algorithms we always want to exploit the locality of gene interactions to facilitate the search process (and therefore two-point crossover is preferred over bit-wise crossover), in many optimization problems it is difficult to assure such locality. As the goal of our experiments was not to solve the problems generated by the NKq models *per se*, but rather to find useful dependencies between the beneficial values of algorithm parameters and the characteristics of the optimization problem, we decided to use bit-wise crossover to ensure that the locality of gene interactions cannot be abused by the search process.

Since finding the optimal value in NK models is in general considered to be an NP-hard problem, in order to facilitate normalization of the fitness values returned by the algorithm, we restricted our experiments to relatively low values of the degree of epistasis K. This allowed us to compute the exact global minima and maxima for each considered fitness landscape [23]. After normalization, fitness values are in the range $[0.0, 1.0]$, where 0.0 corresponds to the lowest, and 1.0 – to the highest value obtainable for a given problem.

5 ANALYSIS

Due to high dimensionality of the parameter space used in our experiments (seven parameters in total), paired up with parameters divided into two separate groups, it would be difficult to find regularities and rules through visual examination of the entire set of results. For this reason we decided to use linear regression to search for simple regularities.

The analysis involved two stages of multiple linear regression. In the first stage, multiple linear regression was performed separately for each combination of the NKq model parameters. The parameters of the algorithm were used as independent variables, and the highest fitness value found by the algorithm, averaged over 15 runs, was used as a dependent variable. The results of the first stage of regression provide coefficients for each of the parameters of the algorithm. These coefficients can be then interpreted as a general gradient in which parameters of the algorithm should be

		N	20	50	100
K	q				
2	2		0.299	0.758	0.897
2	10		0.435	0.833	0.920
2	∞		0.344	0.793	0.908
5	2		0.255	0.837	0.841
5	10		0.237	0.802	0.877
5	∞		0.273	0.711	0.914
8	2		0.129	0.778	0.854
8	10		0.030	0.753	0.876
8	∞		0.069	0.747	0.866

Table 1: R^2 values for the regressions of fitness (first stage) after 100 000 evaluations. R^2 values for other lengths of evolution exhibit similar patterns, although generally R^2 decreases the longer the evolution.

changed in order to improve the performance of the algorithm for a given parametrization of the NKq model. Therefore, one could say that these coefficients correspond to a certain "goodness" of the parameter. For example, should the coefficient for the number of subpopulations M be equal to 0.001, that would mean that by changing M from 10 to 25 one should expect the fitness of the solution found by the algorithm to increase by 0.015.

In the second stage of the analysis, we have used the algorithm parameter coefficients from the first stage (their "goodness") as a dependent variables, and performed – once more – a multiple regression for each algorithm parameter separately. This time, the parameters of the NKq model were used as independent variables. Coefficients for these parameters (we shall call them "goodness correlation coefficients" for the remaining part of the paper) should reveal how the goodness of each parameter of the algorithm changes as the characteristics of the fitness landscape is changed. For example, if the coefficient for the degree of epistasis K were equal to -0.0003 for the number of subpopulations M, that would mean that increasing K by 10 would decrease the goodness of M by -0.003. This in turn would mean that if, for a lower degree of epistasis, increasing M was beneficial (as in the previous example where it led to the increase in fitness by 0.015), it could be in fact detrimental for higher epistasis; using data from the previous example, it would lead to the decrease in fitness by 0.03. In general, it can be assumed that the positive goodness correlation coefficient indicates that the optimal value of the parameter of the algorithm increases as the value of the parameter of the fitness landscape increases. In the case described above, the negative goodness correlation coefficient suggests that the optimal value of M decreases as K grows. Alternatively, a positive value of the goodness correlation coefficient might imply that the magnitude of the effect the parameter of the algorithm has on fitness grows across the board as the value of the parameter of the fitness landscape increases, without the optimal value of the parameter of the algorithm actually changing. In this paper however, we focus on the former possibility.

In the experimental data, we expect not only a lot of nonlinearity (e.g., a linear increase in the tournament size leads to a sublinear increase in selective pressure), but also non-monotonicity – each parameter is expected to have some optimal value, changes to which

	const	N	K	q
const	1.06	−1.4e−3	−1.3e−2	−1.4e−5
t	1.3e−3	1.2e−4	−4.2e−4	2.2e−6
S	−6.3e−5	−1.0e−5	4.5e−5	1.3e−8
M	−3.4e−4	−3.3e−5	1.4e−4	−1.5e−6
R	6.5e−5	2.0e−6	−9.0e−6	−1.9e−7

Table 2: Regression coefficients reflecting the influence of landscape parameters (columns) on goodness of algorithm parameters (rows) after 100 000 evaluations. "const" is the constant term of the regression model.

will lead to a decrease in performance of the algorithm. The value of ∞ for parameter q cannot be properly described with real numbers – in this case we assumed $q = 100$, which is a decent approximation due to the nonlinear nature of this parameter. Because of these issues, results of this analysis should be taken with caution, especially if R^2 values for the regression model are low, as is the case for lower values of N and longer evolutionary runs (as shown in Table 1). Still, this analysis may find some general tendencies and correlations present in the data; even though the exact values of the regression coefficients may not be of much use, their signs and magnitude may indicate a general negative or positive trend, or lack thereof.

6 RESULTS AND DISCUSSION

Fig. 1 shows average fitness values obtained for each combination of parameters (excluding the tournament size t) after 100 000 evaluations. It is clearly visible that most of the variance in fitness values comes here from the parameters of the fitness landscape, not from the parameters of the algorithm. This observation is not surprising, as increasing the size of the problem N and the degree of the epistasis K makes the problems more difficult, and the solutions found for more difficult problems are to be expected to be farther from the global optimum. It is however interesting to note that the degree of neutrality in a problem does not seem to influence the difficulty of the problem.

Fig. 2 presents the same data, but this time fitness is normalized for each parametrization of the NKq model separately. This allows one to see finer details and the structure of influence the parameters of the algorithm have on fitness values. The most obvious observation is that the gradient of fitness values does not vary a lot depending on the parameters of the NKq model: generally, lower sizes (S) and numbers (M) of subpopulations, and longer periods (R) between mixing events yield higher fitness. This reveals that – at least for problems generated by the NKq model – it should be possible to select parameters that provide high performance most of the time. However, Fig. 3 shows that after 1 000 000 evaluations this is no longer the case – when evolution runs for a long time, the effect the parameters of the fitness landscape have on the optimal values of the parameters of the algorithm is no longer obvious.

Other differences in fitness gradients may be too subtle to be noticed by eye and may require statistical analysis, as described in Sect. 5. Fig. 4 shows how the goodness correlation coefficients for different parameters of the convection selection and fitness landscapes change in time. The figure omits the scale on the vertical

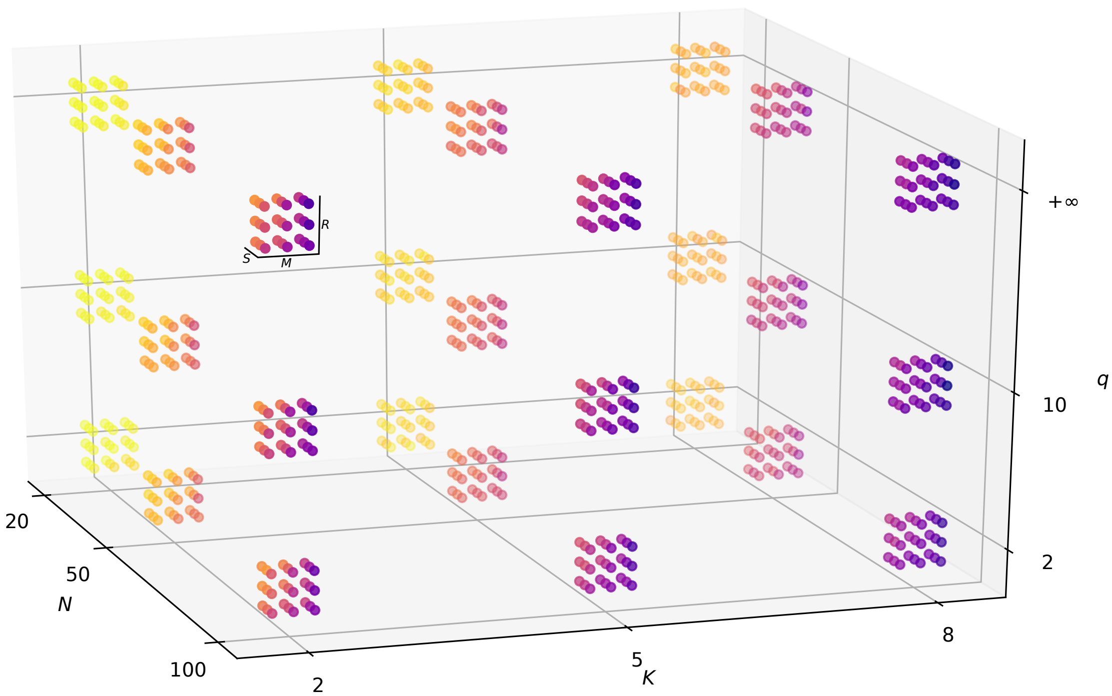


Figure 1: A 6D plot demonstrating 3D mini-plots with evolutionary algorithm parameters (subpopulation size S, subpopulation number M, mixing time R) embedded in a 3D plot with the NKq model parameters. Colors reflect fitness values (dark blue is the lowest, yellow is the highest) normalized using the optimal values. In this plot, tournament size is 2, and the evolutionary process is visualized after 100 000 evaluations.

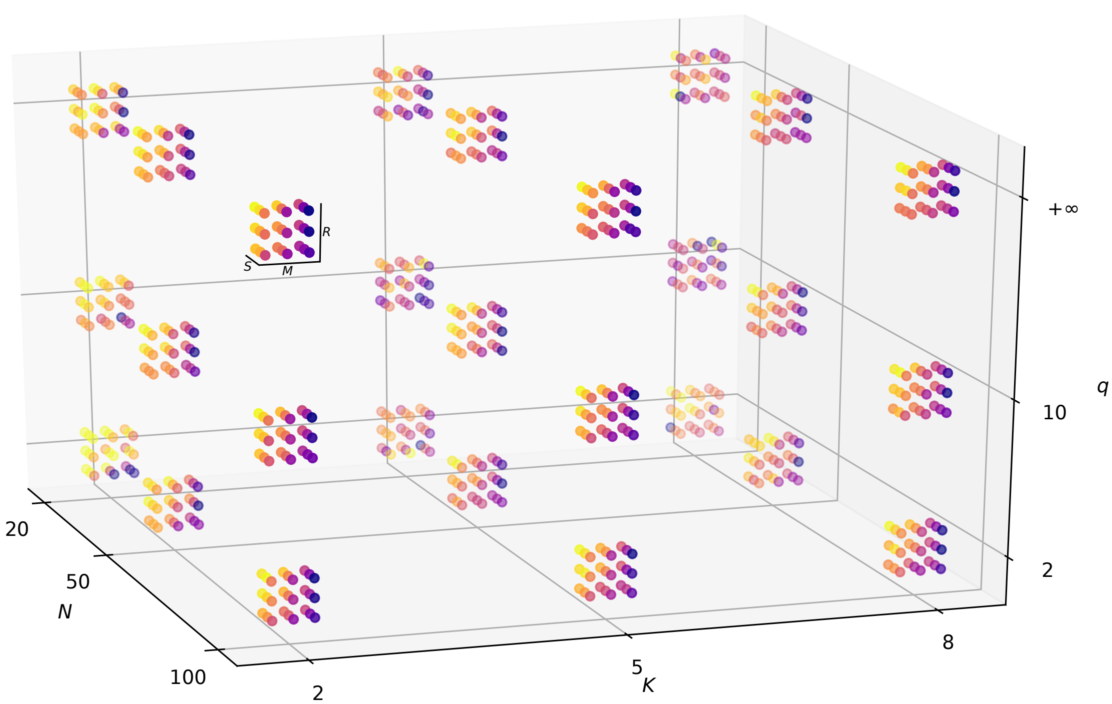


Figure 2: A 6D plot similar to the one in Fig. 1, but here fitness values were normalized independently for every mini-plot (a given NKq model). The evolutionary process is visualized after 100 000 evaluations.

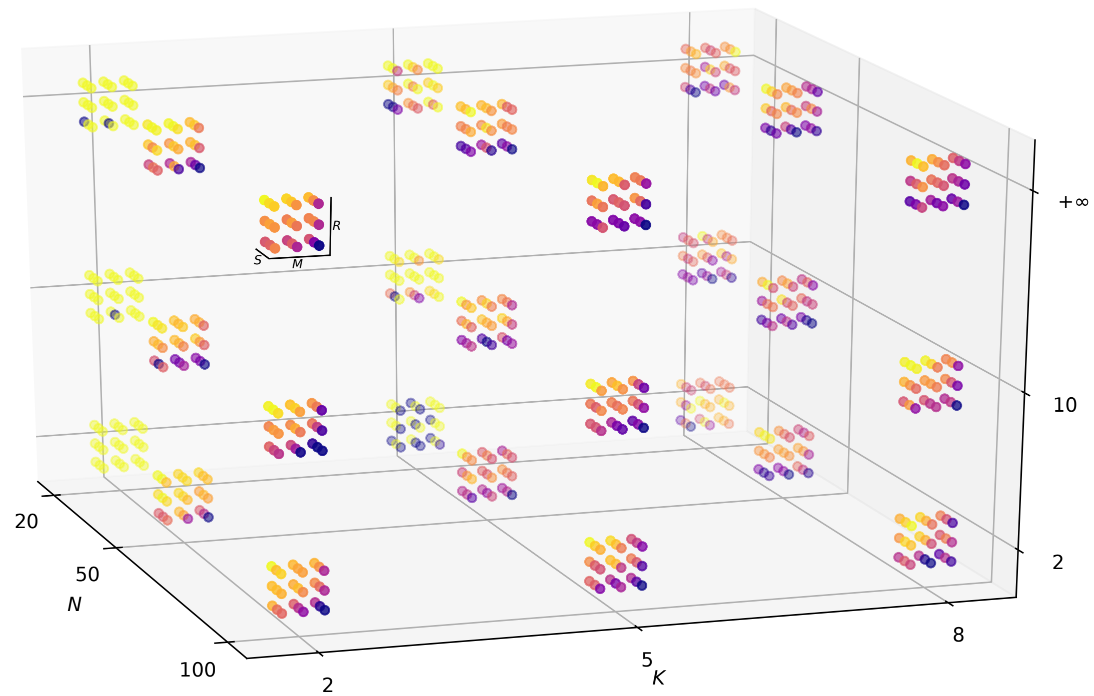


Figure 3: A 6D plot similar to the one in Fig. 2, but here the evolutionary process is visualized after 1 000 000 evaluations.

	const	N	K	q
const	1.06	−1.0e−3	−1.0e−2	−2.6e−6
t	−1.0e−3	6.3e−5	4.0e−5	−4.0e−6
S	5.3e−5	−4.0e−6	4.0e−6	2.0e−7
M	2.8e−4	−1.5e−5	1.7e−5	−1.5e−7
R	−3.8e−5	4.0e−6	−5.0e−6	3.5e−7

Table 3: Regression coefficients reflecting the influence of landscape parameters (columns) on goodness of algorithm parameters (rows) after 500 000 evaluations. "const" is the constant term of the regression model.

	const	N	K	q
const	1.06	−9.2e−4	−9.2e−3	−1.6e−5
t	−1.3e−3	4.3e−5	1.3e−4	−2.4e−6
S	4.1e−5	−2.0e−6	−5.0e−6	2.8e−7
M	2.5e−4	−9.0e−6	2.0e−6	−2.5e−8
R	−1.0e−4	4.0e−6	4.0e−6	5.8e−7

Table 4: Regression coefficients reflecting the influence of landscape parameters (columns) on goodness of algorithm parameters (rows) after 1 000 000 evaluations. "const" is the constant term of the regression model.

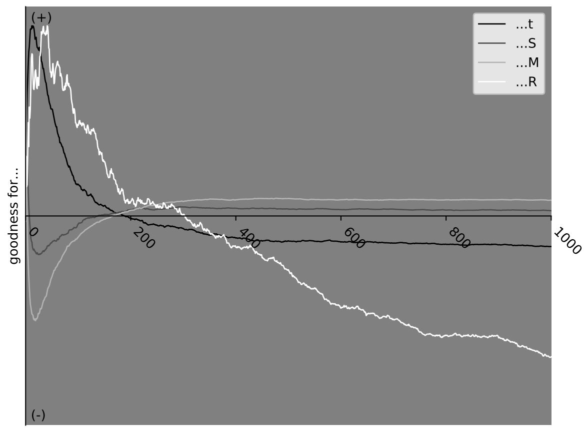
axis, as the exact values of the parameter goodness correlation coefficient are mostly irrelevant. This allows for each series to be scaled differently. For the examples of the exact values of the goodness correlation coefficients we refer the reader to Tables 2, 3 and 4.

It is worth observing that the goodness correlation coefficients change dynamically in early evolution, and tend to stabilize as the evolution continues. As an example, let us consider Fig. 4a, which shows how the goodness of the parameters changes in time (independently of the problem). At the very start the algorithm prefers many big subpopulations, as the initial best fitness in the population is strongly correlated with the size of the full initial population. This preference however reverses very quickly, as the big population inevitably slows down the otherwise very quick initial increase in the fitness of the population. After the initial surge, as the optimization becomes more difficult, the process stabilizes with a slight preference towards the bigger and more numerous populations. Similarly, although the size of the tournament (and therefore the selective pressure) has at first no effect on the fitness values (as the fitness of the initial population is completely independent of the selective pressure), the high selective pressure quickly becomes an effective way to increase fitness in the population, only to become detrimental on longer evolutionary scales, where lower selective pressure can help avoid becoming stuck in local optima.

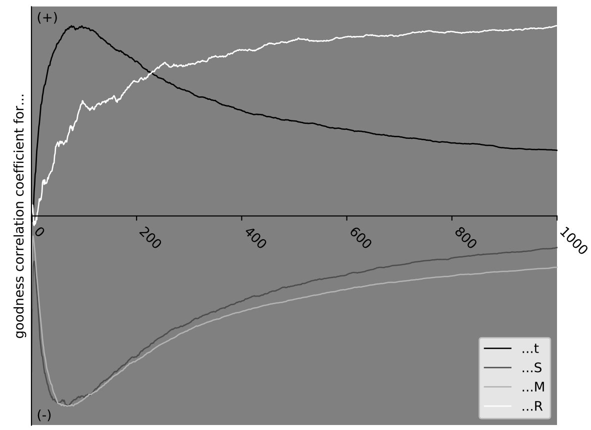
Similar analyses can be performed for combinations of the parameters of the convection selection and the parameters of the fitness landscape. Below, we will focus on the more interesting ones.

Fig. 4b shows the effect the increase in the size of a problem N has on the goodness of parameters of the algorithm. It can be seen that bigger problems will in general require a higher selective pressure, a lower number of smaller subpopulations and longer intervals between the mixing of subpopulations.

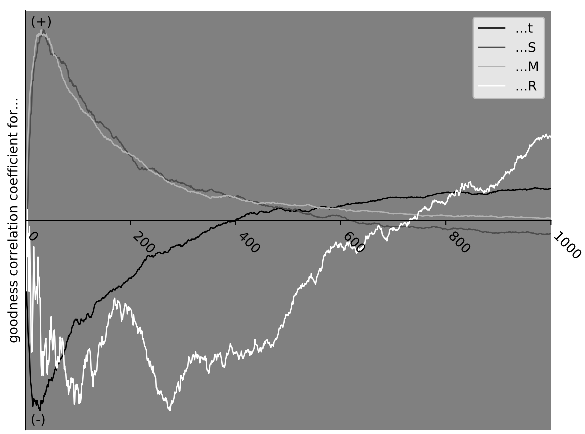
An interesting behavior is visible when increasing the level of epistasis, K, as presented in Fig. 4c. Initially, higher K leads to



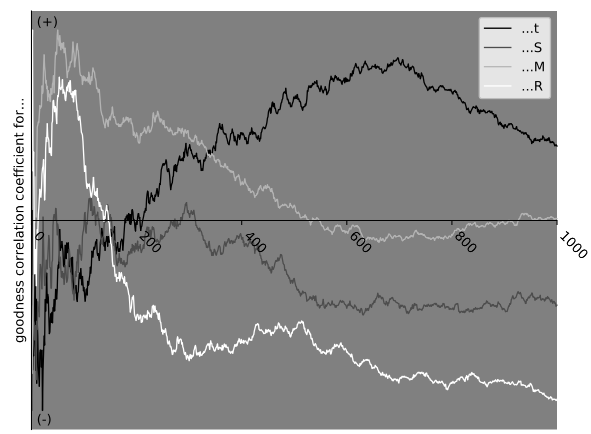
(a) Goodness independent of the parameters of the problem.



(b) Goodness correlation coefficients dependent on N (the size of the problem).



(c) Goodness correlation coefficients dependent on K (the degree of the epistasis).



(d) Goodness correlation coefficients dependent on $1/q$ (the degree of the neutrality).

Figure 4: Changes of the parameter goodness correlation coefficients (vertical axis) dependent on the parameters of the fitness landscape in time (horizontal axis, counted as thousands of evaluations of individuals). Each series corresponds to one of the parameters of the algorithm, and each plot corresponds to one of the parameters of the fitness landscape (or a constant term, independent of the problem). A positive goodness correlation coefficient of a parameter of the algorithm for a given parameter of the fitness landscape indicates that the optimal value of the parameter of the algorithm is expected to increase as the value of the parameter of the fitness landscape increases. The opposite is true for the negative goodness correlation coefficient values. Zero goodness correlation coefficient suggests that the changes in the value of a given parameter of the fitness landscape have no effect on the optimal value of the parameter of the algorithm. As the exact values of the parameter goodness correlation coefficient are mostly irrelevant, the vertical axis has no designated scale and each series is rescaled independently. Examples of the exact values of the parameter goodness correlation coefficients are shown in Tables 2, 3 and 4.

increased preference of a low selective pressure, short intervals between mixing events, and a higher number of bigger subpopulations, all of which cause higher exploration of the search space. For longer evolutionary experiments, this trend reverses – and less exploratory, more exploitative behaviors are preferred as K increases. This can be interpreted as a suggestion that while the use of a higher exploration in convection selection can give one a head start in difficult problems, on longer time scales higher exploration does not have to be as useful.

In order to facilitate the interpretation of the results, Fig. 4d shows the goodness correlation coefficients dependent on $1/q$ instead of q, as $1/q$ can be interpreted as the degree of the neutrality in a problem. This figure demonstrates perhaps the most intriguing of our results. For most parameters of the fitness landscape, the size and the number of subpopulations exhibit a very similar behavior (which suggests that it is not important how exactly the entire population is divided into subpopulations, but rather just how big it is). Here, however, the number of the subpopulations has a positive (and – eventually – mostly neutral) effect on the fitness values returned by the search process, while the size of the subpopulations has generally a negative impact on fitness. This suggests that, at least initially, a higher number of subpopulations facilitates simultaneous exploration of many different neutral networks. Although initially a high selective pressure has a negative impact on the search process when the problem exhibits high neutrality, it eventually becomes preferred over a low selective pressure. This preference grows as the evolution becomes longer, however after 700 000 evaluations – although still present – it starts to diminish. This shows that the degree of neutrality may influence the goodness of the parameters of the algorithm in a complex way – even on longer evolutionary time scales.

7 SUMMARY

In this work we studied the relationships between values of the convection selection parameters and the characteristics of the fitness landscape. The analysis concerned the performance of an evolutionary algorithm in the NKq model. The results of the regression of fitness (and subsequently the results of the regression of the coefficients of that regression of fitness) suggest that the increase in the dimensionality of the problem, N, can lead to higher exploitation of the fitness landscape being favored. On the other hand, the increase in the level of epistasis K leads to higher exploration being favored in early stages of evolution, however, as the evolutionary time passes, this preference is reversed. We have also shown that the presence of neutral networks in a fitness landscape can have an interesting, non-obvious effect on the optimal values of the parameters of convection selection, and perhaps also on other parameters of evolutionary algorithms.

Alongside the results described above, the additional value of the experiments comes from the introduced methodology. We have performed a linear regression analysis on two levels – first, to assess the direction of the improvement in the space of the parameters of the algorithm for the given parameters of the problem, and then again, based on the coefficients resulting from the first regression, to capture how the direction of improvement changes as the parameters of the problem change. Although the exhaustive exploration of the vast space of many possible combinations of parameters is usually a strenuous task, the presented methodology facilitates the drawing of meaningful conclusions about the interactions between the parameters of the problem and the parameters of the algorithm. Such conclusions are more comprehensive than simple knowledge on how to set reasonable values of algorithm parameters for a given problem. They may also allow to gain an insight into the way the algorithm performs. The systematic, automated and quantitative approach offered in this work allows for gaining a deeper understanding of the behavior of various algorithms, and for the extrapolation of the observed trends beyond the examined sets of parametrizations.

The work reported in this paper can be extended in a number of ways. Higher values of N and K can be tested, which could strengthen the conclusions from this analysis. The NKp model could be investigated in addition to NKq, as the former model may have a more severe effect on the preferred values of parameters of convection selection.

REFERENCES

[1] Lionel Barnett. 1998. Ruggedness and neutrality – the NKp family of fitness landscapes. In *Proceedings of the 6th international conference on Artificial life*. MIT Press, Cambridge, MA, USA, 18–27.
[2] Kenneth Alan De Jong. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. Dissertation. University of Michigan, Ann Arbor, MI, USA. AAI7609381.
[3] Marc Ebner, Patrick Langguth, Juergen Albert, Mark Shackleton, and Rob Shipman. 2001. On neutral networks and evolvability. In *Proceedings of the Congress on Evolutionary Computation (CEC2001)*, Vol. 1. IEEE, 1–8.
[4] Nicholas Geard, Janet Wiles, Jennifer Hallinan, Bradley Tonkes, and Ben Skellett. 2002. A comparison of neutral landscapes – NK, NKp and NKq. In *Proceedings of the Congress on Evolutionary Computation (CEC2002)*, Vol. 1. IEEE, 205–210.
[5] David E. Goldberg, Jon Richardson, et al. 1987. Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 41–49.
[6] Georges R. Harik. 1995. Finding Multimodal Solutions Using Restricted Tournament Selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 24–31. http://dl.acm.org/citation.cfm?id=645514.658086
[7] John Henry Holland. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, Cambridge, MA, USA.
[8] Marcus Hutter and Shane Legg. 2006. Fitness uniform optimization. *IEEE Transactions on Evolutionary Computation* 10, 5 (2006), 568–589.
[9] Stuart A. 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, 2 (1989), 211–245.
[10] Hemanshu Kaul and Sheldon H. Jacobson. 2006. Global optima results for the Kauffman NK model. *Mathematical Programming* 106, 2 (2006), 319–338.
[11] Maciej Komosinski and Konrad Miazga. 2018. Comparison of the tournament-based convection selection with the island model in evolutionary algorithms. *Journal of Computational Science* (2018). https://doi.org/10.1016/j.jocs.2018.10.001
[12] Maciej Komosinski and Konrad Miazga. 2018. Tournament-based convection selection in evolutionary algorithms. *PPAM 2017 proceedings, Lecture Notes in Computer Science* 10778 (2018), 466–475. https://doi.org/10.1007/978-3-319-78054-2_44
[13] Maciej Komosinski and Adam Rotaru-Varga. 2001. Comparison of different genotype encodings for simulated 3D agents. *Artificial Life Journal* 7, 4 (Fall 2001), 395–418. https://doi.org/10.1162/106454601317297022
[14] Maciej Komosinski and Szymon Ulatowski. 2017. Multithreaded computing in evolutionary design and in artificial life simulations. *The Journal of Supercomputing* 73, 5 (2017), 2214–2228. https://doi.org/10.1007/s11227-016-1923-4
[15] Maciej Komosinski and Szymon Ulatowski. 2019. Framsticks Web Site. (2019). http://www.framsticks.com
[16] Shane Legg, Marcus Hutter, and Akshat Kumar. 2004. Tournament versus fitness uniform selection. In *Proceedings of the Congress on Evolutionary Computation (CEC2004)*, Vol. 2. IEEE, 2144–2151.
[17] Samir W. Mahfoud. 1995. *Niching Methods for Genetic Algorithms*. Ph.D. Dissertation. University of Illinois at Urbana-Champaign, Champaign, IL, USA. UMI Order No. GAX95-43663.
[18] Martin Pelikan. 2008. Analysis of estimation of distribution algorithms and genetic algorithms on NK landscapes. In *Proceedings of the 10th annual conference on genetic and evolutionary computation*. ACM, New York, NY, USA, 1033–1040.
[19] Ponnuthurai N. Suganthan, Nikolaus Hansen, Jing J. Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. 2005. *Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization*. Technical Report. Nanyang Technological University.
[20] Sébastien Verel, Gabriela Ochoa, and Marco Tomassini. 2011. Local optima networks of NK landscapes with neutrality. *IEEE Transactions on Evolutionary Computation* 15, 6 (2011), 783–797.
[21] K. Warwick and Erik D. Goodman. 2002. The hierarchical fair competition (HFC) model for parallel evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation (CEC2002)*. IEEE, 49–54.
[22] David H. Wolpert and William G. Macready. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1, 1 (1997), 67–82.
[23] Alden H. Wright, Richard K. Thompson, and Jian Zhang. 2000. The computational complexity of NK fitness functions. *IEEE Transactions on Evolutionary Computation* 4, 4 (2000), 373–379.