



HAL
open science

Boomerang-shaped neural embeddings for NK landscapes

Roberto Santana, Arnaud Liefoghe, Bilel Derbel

► **To cite this version:**

Roberto Santana, Arnaud Liefoghe, Bilel Derbel. Boomerang-shaped neural embeddings for NK landscapes. GECCO 2022 - Genetic and Evolutionary Computation Conference, Jul 2022, Boston, MA, United States. pp.858-866, 10.1145/3512290.3528856 . hal-03693668

HAL Id: hal-03693668

<https://hal.science/hal-03693668>

Submitted on 2 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Boomerang-shaped Neural Embeddings for NK Landscapes

Roberto Santana
Univ. of the Basque Country
(UPV/EHU)
San Sebastian, Spain
roberto.santana@ehu.eus

Arnaud Liefoghe
Univ. Lille, CNRS, Inria, Centrale
Lille, UMR 9189 CRIStAL
F-59000 Lille, France
arnaud.liefoghe@univ-lille.fr

Bilel Derbel
Univ. Lille, CNRS, Inria, Centrale
Lille, UMR 9189 CRIStAL
F-59000 Lille, France
bilel.derbel@univ-lille.fr

ABSTRACT

Understanding the landscape underlying NK models is of fundamental interest. Different representations have been proposed to better understand how the ruggedness of the landscape is influenced by the model parameters, such as the problem dimension, the degree of non-linearity and the structure of variable interactions. In this paper, we propose to use neural embedding, that is a continuous vectorial representation obtained as a result of applying a neural network to a prediction task, in order to investigate the characteristics of NK landscapes. The main assumption is that neural embeddings are able to capture important features that reflect the difficulty of the landscape. We propose a method for constructing NK embeddings, together with metrics for evaluating to what extent this embedding space encodes valuable information from the original NK landscape. Furthermore, we study how the embedding dimensionality and the parameters of the NK model influence the characteristics of the NK embedding space. Finally, we evaluate the performance of optimizers that solve the continuous representations of NK models by searching for solutions in the embedding space.

CCS CONCEPTS

• **Theory of computation** → *Optimization with randomized search heuristics*; • **Computing methodologies** → *Discrete space search; Neural networks*.

KEYWORDS

NK landscapes, fitness landscape, local optima, adaptive walk, neural embedding.

ACM Reference Format:

Roberto Santana, Arnaud Liefoghe, and Bilel Derbel. 2022. Boomerang-shaped Neural Embeddings for NK Landscapes. In *Genetic and Evolutionary Computation Conference (GECCO '22)*, July 9–13, 2022, Boston, MA, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3512290.3528856>

1 INTRODUCTION

Neural embeddings [28, 29] have been mostly used as a vectorial representation of text data, amenable for the application of machine learning techniques. In this context, a word embedding is

an m -dimensional real-valued vector that represents a word from a vocabulary. An advantage of word embedding is that it makes possible to encode semantic relationships between words [4, 9, 12]. That is, the proximity between vectors in the embedding space can be interpreted as a semantic similarity of the corresponding words these vectors codify. Word embeddings can also be used to solve supervised and unsupervised machine learning problems, for instance, as inputs to a model that predicts the class of a given word, or to estimate a continuous value associated with the word. Recently, neural embeddings have also been applied to other types of objects such as graphs [15] and biological sequences [8, 26].

The NK model for fitness landscape has been extensively applied in evolutionary biology and evolutionary optimization [1, 23, 33, 40, 41]. It provides a relevant benchmark to investigate how the ruggedness of the fitness landscape, and consequently the difficulty of the optimization process, are governed by the model parameters.

The main goal of this paper is to introduce alternative landscape representations of NK models, namely, by learning neural embedding representations of NK landscapes, and to study whether and how the features of these embedding spaces provide valuable information about the characteristics of the original landscape. We thereby consider neural networks to associate an m -dimensional continuous representation with each binary solution from an NK landscape. We evaluate how the properties of this space are related to the characteristics of the original NK landscape. More particularly, we evaluate how neighbors in the discrete space are located in the embedding space, and the way in which a path that goes from a solution to the local optimum of its basin of attraction is projected into the embedding space. We also explore the way in which the embedding dimensionality (m) and the degree of non-linearity of NK landscapes (K) influence the properties of the so-obtained NK embedding spaces. Finally, we investigate the application of numerical optimization methods in the embedding space, as a way to assess the difficulty of the original problem. Extensive experiments are conducted using the CMA-ES algorithm [16].

The remainder of the paper is organized as follows. Section 2 presents some background related to NK landscapes and neural embeddings. Section 3 gives an overview of related work. Section 4 explains the steps we follow to learn the NK model embedding representations, and the metrics we consider to evaluate them. Section 5 lists the experimental set-up, describes the numerical results and discusses the results of the experiments. Finally, Section 6 concludes the paper and discusses future work.

2 BACKGROUND

In this section we briefly present the main concepts related to NK landscapes and neural embeddings.

2.1 NK Landscapes

Let $\mathbf{X} = (X_1, \dots, X_N)$ denote a vector of N binary variables. We will use $\mathbf{x} = (x_1, \dots, x_N)$ to denote an assignment to the variables. S will denote a set of indices in $\{1, \dots, N\}$, and X_S (respectively x_S) a subset of the variables of \mathbf{X} (respectively \mathbf{x}) determined by the indices in S .

The NK model was originally introduced to study the effect of different epistatic patterns in the landscape of the function. An NK landscape [19] is defined by the following components:

- The number of variables, N , that defines the problem dimension.
- The number of interactions per variable, K , that defines the problem non-linearity.
- A set of K interactions, $\Pi(X_j) \subseteq \mathbf{X}$, for every $X_j, j \in \{1, \dots, N\}$.
- A sub-function f_j defining a real value for each combination of values of X_j and $\Pi(X_j), j \in \{1, \dots, N\}$.

The fitness function F , to be maximized, is then defined as follows:

$$F(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^N f_j(x_j, \Pi(x_j)). \quad (1)$$

By increasing the number of interactions K per variable, NK landscapes can be gradually tuned from smooth to rugged.

2.2 Neural Embedding

Neural embedding appeared as a straightforward way to apply machine learning algorithms to problems with a sparse representation [29]. In natural language processing, the problem arises from the issue of how to represent words and sentences so that they could be processed by machine learning models such as neural networks. A natural way to represent a sentence is using a binary vector where each word i from a vocabulary is either present ($x_i = 1$) or not ($x_i = 0$). Since only a very small fraction of the words in a vocabulary are typically present in a sentence, this bag-of-words representation is extremely sparse. By contrast, word embeddings are continuous vector representations of words which in comparison with the bag-of-words representation has a lower dimensionality, and are denser. In a word embedding vector space, each word in a vocabulary has an associated real-valued vector.

To learn word embeddings, neural networks are used. First, a vocabulary of words is defined, and a corpus of sentences that contains words from this vocabulary is selected. Then, a machine learning task receives the bag-of-words representation of the sentence as input. Once the network is trained with the corpus, the set or word embeddings is extracted from the weights of the neural network.

For instance, in the word2vec approach [28], a feed-forward neural network language model [7] with a number of added changes is used to learn two types of embedding models, the Skip-gram, and the continuous bags-of-words (CBOW) models. Both models use a bag-of-words representation for each word; i.e., a binary vector of dimension equal to the vocabulary size and where only one element is set to 1. The difference between these two models lies in the prediction task they address: Skip-gram learns to predict the surrounding words of a given word $w(t)$ in a sentence, while CBOW learns to predict the word that is most likely to be in the center,

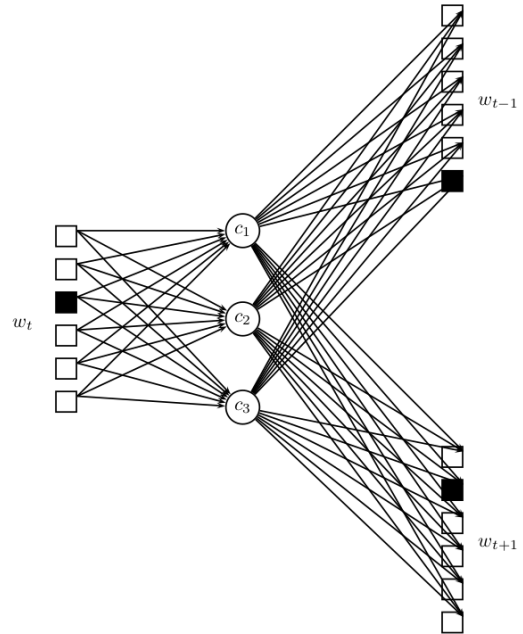


Figure 1: Skipgram model used to learn the embeddings.

given the surrounding words. The surrounding words of $w(t)$ are those from $t - k$ to $t + k$, where k is an algorithm parameter that determines the window size ($2k$).

Figure 1 shows a Skip-gram model for a vocabulary of size 6, an embedding dimension $m = 3$, and a window size of 2. Each unit i of the input layer, corresponding to the i -th word in the vocabulary, is connected to all units of the (single) hidden layer. The word embedding corresponding to word i , extracted after the neural network training is completed, has in position j the j -th weight associated to the connection from i to the j -th hidden unit. In the example shown in Figure 1, the input word is the third word of the vocabulary, and its context in the sentence is formed by the sixth and second words of the vocabulary, highlighted in the bag-of-words representations of words $w(t - 1)$ and $w(t + 1)$.

Beyond constituting a more compact representation, word embeddings have been shown to capture a number of regularities from the corpus under which they were trained [14, 21, 43]. For instance, words whose vectors are close in the embedding space are usually related semantically. Furthermore, some algebraic operations between vectors can be used to infer similarity and semantic relationships between words. For instance, since the distance between embeddings for words *man* and *woman* is similar to the distance between embeddings for words *king* and *queen*, it is possible to define algebraic equations such as $\vec{queen} = \vec{woman} - \vec{man} + \vec{king}$, where \vec{w} indicates the embedding representation for word w .

3 RELATED WORK

In this section, we review related work in three areas: representations proposed to study NK landscapes, applications of evolutionary algorithms on embedding representations, and proposals of

methods and algorithms on embedding representations for other domains.

3.1 Representation for NK Landscapes

Buzas and Dinitz [10] proposed a characterization of NK landscapes using parametric linear models comprised of main effects and interaction effects. Based on this “analytical” representation, they computed the statistical distribution of the model effect coefficients, and showed that the model coefficients have meaningful interpretations.

In their seminal work, Ochoa et al. [31] introduced a network characterization of NK landscapes. In this network representation, each vertex corresponds to a local optimum, and edges represent basin adjacency between two local optima. The representation is compact since non-optimal solutions are discarded. The analysis of local optima networks (LONs) shows the existence of *small-world* characteristics. The network representation additionally allows for the extraction of general network metrics that serve to compare how different characteristics of the fitness landscapes change according to the parameters of the NK model. A number of extensions and variations of LONs have been proposed in the literature [32, 39, 41]. Most relevantly, these models have been instrumental in the identification of features inferring on problem difficulty and on algorithm performance prediction [11, 17].

Such a network representation approach has been later extended to study multi-objective NK landscapes. The Pareto local optimal solutions network (PLOS-net) is introduced in [22]. Similarly to LONs, sub-optimal solutions are not represented in the network, vertices correspond to PLOs and edges are constructed between mutually non-dominated neighbors. Network metrics are extracted from PLOS-nets and used to explain the performance of different types of multi-objective algorithms.

The work presented in this paper have a number of similarities and differences from network-based approaches. Similarly to LONs, and as it will be discussed in Section 4, local optima play an important role in the construction of the embedding representations. Exhaustive search or sampling are required to construct a set of paths that, in each case, lead to a local optimum. In contrast with LONs, the representation of solutions lies in a real-valued space.

3.2 Evolutionary Algorithms and Embedding Representations

A number of works have investigated the combination of evolutionary computation and embedding representations. In [25, 37, 38], word embedding representations learned by means of word2vec are used as inputs to evolve genetic programs which are able to solve different natural language processing tasks. The dimension of the embeddings and the choice of the genetic programming fitness function were shown to play an important role in algorithm performance. Roman et al. [35] applied genetic programming to sentence embeddings to evolve Gaussian kernels for sentiment classification. Similarly to the case when word embeddings are used, genetic programming shows its ability to efficiently evolve programs that process the sentence embedding representation to solve the prediction tasks.

Also related to our research are proposals that apply embeddings for a better visualization and understanding of evolutionary search spaces. Michalak [27] proposes to use a different class of embeddings, known as the t-Distributed Stochastic Neighbor Embedding (t-SNE). At last, Fyvie et al. [13] apply principal components analysis for trajectory mining of estimation of distribution algorithms. They used this approach to identify new methods of population diversity. To our knowledge, neural embeddings have not been previously used as an analytical or visualization tool of search spaces in evolutionary computation.

3.3 Embedding Representations for Other Domains

While neural embeddings have been mainly proposed for text representation, an increasing number of works report representations learned for other domains, including biological sequences [5, 8, 42], program source codes [3] and graphs [34]. Particularly relevant for our work is research on learning graph embeddings. Some of these approaches construct the graph node descriptions by first conducting a random walk on the graph [15, 34]. In this paper, we take inspiration from these previous approaches to create sequences of search trajectories from which we construct the corpora required to learn the embeddings.

4 INTRODUCING NEURAL EMBEDDING REPRESENTATIONS OF NK LANDSCAPES

As discussed in Section 2.2, word embeddings are learned from a corpus of sentences comprising words from a vocabulary. The embedding captures the semantic relationships between the words as expressed in the sequences. In our approach, we want to represent the relationship between solutions from an NK landscape (either local optima or not), in terms of their fitness values. Therefore, our vocabulary will be integrated by NK solutions; i.e., each word from our vocabulary is a binary solution from the search space.

We define a sentence as a sequence of NK model solutions that are related according to some *predefined semantics*. In this paper, the chosen semantics establishes that each solution in the sequence, except the first one, is the best improving neighbor for the previous solution in the sequence. The neighborhood structure \mathcal{N} can be defined in different ways, we here consider the usual 1-bit flip neighborhood structure; i.e., two solutions are neighbors if their Hamming distance is 1. Sequences of solutions can be constructed by means of adaptive walks [19, 20, 30]. For small search spaces, an alternative is to conduct an exhaustive enumeration and to run an adaptive walk from each possible solution of the search space, as done for constructing LONs in [31]. For the purpose of our analysis, we follow this exhaustive approach in the paper.

For each solution from the binary search space $x_0 \in X$, we generate the sequence $(x^0, x^1, \dots, x^\ell)$ such that $x^t \in \mathcal{N}(x^{t-1})$ and $F(x^t) > F(x^{t-1})$ for all $t \in \{1, \dots, \ell\}$. We follow a best improvement adaptive walk such that $x^t = \arg \max_{x \in \mathcal{N}(x^{t-1})} F(x)$, $t \in \{1, \dots, \ell\}$, until no further improvement is possible. The adaptive walk determines the local optimum x^ℓ , and therefore the basins of attraction of x^0 , required for constructing the sequences for inferring the embedding space. Notice that, when there is no equivalent solution in the neighborhood, there is one single local optima

t	x^t	$I(x^t)$	$F(x^t)$	t	x^t	$I(x^t)$	$F(x^t)$
0	010011	19	0.25	0	100100	36	0.73
1	110011	51	0.43	1	110100	52	0.76
2	110001	49	0.54	2	110100	52	0.76
3	110101	53	0.70	3	110100	52	0.76
4	110100	52	0.76	4	110100	52	0.76
5	110100	52	0.76	5	110100	52	0.76

Figure 2: Construction process for sentences (19, 51, 49, 53, 52, 52) and (36, 52, 52, 52, 52, 52) computed for a toy NK landscape with 6 variables.

per starting solution for such an adaptive walk. Figure 2 shows an example of the way two sentences are constructed for a toy NK landscape with 6 variables. In Figure 2, t is the order of the words in the sentence, x^t the binary representation of the solution, $I(x^t)$ the index of the binary solution in the vocabulary, and $F(x^t)$ the NK model fitness value.

The embedding representation is learned from the set of sequences by means of gradient optimization of the neural network loss function. Since the neural network is shallow, e.g., the network in Figure 1 has only one hidden layer, the learning process is extremely fast and is scalable to datasets comprising of millions of sentences. In this paper, we rely on the word2vec algorithm [28], that represents sentences as a sequence of indices of the words in the vocabulary. This means that the model does not actually use the particular encoding associated with each index, being a word for word embeddings or a binary vector for NK embeddings. What is relevant for the algorithm is the way in which these components are organized in the sequence.

Once the neural network is trained, the learned embeddings are extracted as explained in Section 2.2 and will serve as a basis to scrutinize the characteristics of the considered NK landscape, as captured by the embeddings. More details on the methods for learning word embeddings using the word2vec method can be found in [28, 29].

4.1 Parameters for Learning NK Embeddings

An important parameter to be specified before using the word2vec algorithm is the size of the embedding representation. In this paper we consider to study the impact of using different embedding dimensions. Furthermore, the word2vec algorithm assumes that for a word to be represented as an embedding, it has to appear a minimum number of times in the corpus (5 times is the default parameter). This represents an issue in our case since some NK model solutions might only appear in a single sequence. To sort out this obstacle, we simply include five copies of the complete set of sequences in the input dataset. This guarantees that each solution is represented by one embedding, without biasing the embedding representation. The incurred computational cost can be neglected since the embedding space is only learned once.

In addition to the number of times each sentence is included in the corpus, the way we generate sequences of NK solutions determine that they may have a different length. For instance, a sequence that starts from a local optimum has a single element. Since this sort of short sequences does not allow the algorithm to

learn any pattern from the sequence, we simply complete short sequences to a minimum length of 5 elements. Completion is done by repeating the last element from the original sequence (i.e., the local optimum) until reaching the required minimum length.

Another relevant parameter for the word2vec algorithm is the window size. We set this parameter to $k = 1$, and we use the skip-gram variant of word2vec; i.e., the task to be solved by the neural network is to predict the two words that surround every single word in a sequence.

4.2 Evaluating the Information Content of NK Embeddings

The usability of an embedding space lies in its capacity to encode the particular characteristics of the NK landscape. We propose to compute a number of metrics in order to evaluate the information from the NK model that the embedding encodes. In this section, we introduce the metrics and the rationale behind using them.

All the metrics are based on the information about the distance between solutions in the embedding space, given that this distance provides an idea of the “landscape” of the NK model in this space. Commonly, in the analysis of embedding spaces, the cosine distance between embeddings is used. We computed both the Euclidean and the cosine distances in our analysis. However, due to space restriction, we focus on the cosine distance below, as emphasized by the use of *cos* in the metrics described below.

The considered metrics are as follows:

- (1) `nsteps_vs_cos_last`: Correlation between the number of steps (from x^0 to x^ℓ) performed by adaptive walks to reach the local optimum (i.e.; the value of ℓ) and the (cosine) distance between x^0 and x^ℓ in the embedding space.
- (2) `fitness_diff_vs_cos_last`: Correlation between the fitness difference among each starting and ending points of the sequence (i.e.; $F(x^\ell) - F(x^0)$) and the cosine distance between them in the embedding space.
- (3) `max_fitness_diff_vs_cos_last`: Correlation between the fitness difference ($F(x^*) - F(x^0)$) of each starting point x^0 with the global optimum x^* and the cosine distance between them in the embedding space.
- (4) `basin_size_vs_cos_last`: Correlation between the size of the basin of the attraction of x^0 and the cosine distance between them in the embedding space.

For all considered metrics, we use the Spearman correlation coefficient.

The `nsteps_vs_cos_last` metric serves to determine whether local optima that require more steps to be reached from the starting point are similarly more distant from the starting solution in the embedding space. The same rationale is behind the `fitness_diff_vs_cos_last` metric, but in this case we consider the actual difference between the fitness values. The `max_fitness_diff_vs_cos_last` metric is more related to the original fitness-distance correlation measure of problem difficulty [18]. A high correlation would indicate that, as we move closer to the global optimum in the embedding space, the fitness value also increases. At last, the `basin_size_vs_cos_last` metric serves to determine whether there is any relationship between the size of a local optimum’s basin of the attraction and the distance of each point to its local optimum. It addresses the

following question: Is a point closer to its corresponding local optimum if the latter has a larger basin of attraction?

5 EXPERIMENTAL ANALYSIS

In this section, we report experimental results addressing the following research questions:

- How is the landscape of solutions in the embedding space? Are solutions with similar fitness clustered together, or is the distance among solutions in the embedding space determined by their corresponding Hamming distance?
- What is the distribution of local optima in the embedding space?
- Is the landscape in the embedding space smoother than in the original binary representation?
- Which is the influence of K in the characteristics of the embedding space?
- Can optimization be conducted in the space of the embedding representation by mapping the continuous representation to the fitness values of encoded solutions?

We start by introducing the considered benchmark. Thus, a visual exploration of the learned embedding spaces is presented. Subsequently, we provide an analysis of the metrics introduced in Section 4.2. Finally, we report results on the application of a numerical optimizer in the (continuous) embedding space.

5.1 Benchmark

In terms of benchmark problems, we generate NK landscapes with $N = 14$, $K \in \{1, \dots, 12\}$, and random variables interactions. Fitness tables are generated following a uniform distribution in $[0, 1]$. For each parameter setting, 50 instances are independently generated. For each of the 2^{14} solutions in a given NK landscape, we construct the path from each solution to its corresponding local optimum using a best-improvement adaptive walk. To evaluate the effect of the embedding dimension, embedding spaces are generated for each set of sequences for embeddings of dimension $m \in \{2, 3, 5, 10, 25, 50, 100\}$. Therefore, the total number of embedding spaces considered in our experiments is $12 \times 50 \times 7 = 420$.

5.2 Visual Inspections of the Embedding Spaces

Figure 3 gives a representation of the obtained embedding space for different K values and an embedding dimension of $m = 2$ for selected NK landscapes. Local optima are represented as circles, and other solutions as squares. The color of each solution renders its fitness values. We observe that solutions are distributed in a boomerang-shaped area, where local optima are concentrated to the extremes of the boomerang arms. As the number of local optima increases with K , their location gets closer to the base of the boomerang. The shapes also display different orientations for different NK models. However, for a specific instance, the shapes are similar under multiple repetitions of the embedding learning algorithm (data not shown due to space constraints). Another important characteristic of the embedding space is that solutions are not uniformly distributed, and there are large areas where no solution is represented.

Using the embedded representations of the complete space of 2^N solutions, we can map the NK fitness function to the continuous space. However, there are large areas of the continuous space that are empty. In order to “fill the gaps”, we propose to simply assign any m -dimensional vector for which a pre-image in the (binary) domain does not exist with the fitness of the closest point in the embedding space for which a pre-image exists. This amounts to a reconstruction or estimation of the NK model in the continuous space. We use an efficient implementation of the nearest neighbor searching procedure based on KDtrees [24] to determine the fitness value of any point within the boundaries of the embedding space. Figure 4 shows such reconstructions for the same embedding spaces displayed in Figure 3. For the reconstructions, a grid of 120×120 points in the continuous space is used. Displayed in the figure as a black dot is the location of the global optimum. Remarkably, for the three different K values, the global optimum is located in the yellow area that covers most local optima.

5.3 Metric-based Analysis

In the next set of experiments, we evaluate the metrics defined in Section 4.2 for increasing K values and embeddings of different dimensions. Figure 5 reports the correlation between the cosine distance, in the embedding space, from an initial point to its local optimum and: (left) the number of steps performed by the adaptive walk to reach the local optimum, (center) the fitness difference between the starting point and its corresponding local optimum, (right) the fitness difference between the starting point and the global optimum.

We remark that the strongest correlation arises between the number of steps and the distance. The more the intermediate solutions visited by the adaptive walk to reach the local optimum in the original space, the higher the distance from the initial solution to the local optimum in the embedding space. This correlation increases with K but it does differently depending on the dimensionality of the embeddings. The fitness difference between a solution and its local optimum is, for any K and m values, always positively correlated with the distance between the two solutions in the embedding space. However, the strength of the correlation depends on K , and there is an important gap in the strength of these correlations for embeddings of dimension $m = 2$ and those of higher dimensions. These results indicate that the distribution of solutions according to the fitness, as reported in Figure 3, significantly changes in higher dimensions.

The fitness-distance correlation was one of the first metrics proposed for fitness landscape analysis [18]. We report the fitness-distance correlation using the distance in the embedding space in Figure 5 (right). It can be seen in the figure that the correlation is always negative but the strength of the correlation depends both on K and m . For a high dimensionality the embedding space is highly deceptive, being closer to the global optimum does not imply an improvement in the fitness. It is important to notice that the correlations are computed considering all solutions. Another approach could consider solutions that are within a certain distance to the global optimum

The correlation between the distance and the size of local optima’s basins of attraction is reported in Figure 6. It is weaker than

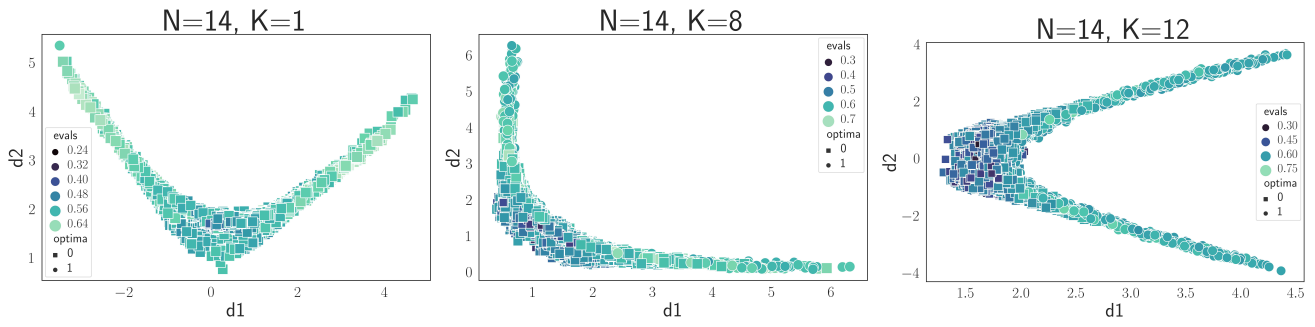


Figure 3: Embedding spaces for exemplary NK landscapes with different K values. The size of the local optima is proportional to their basin size.

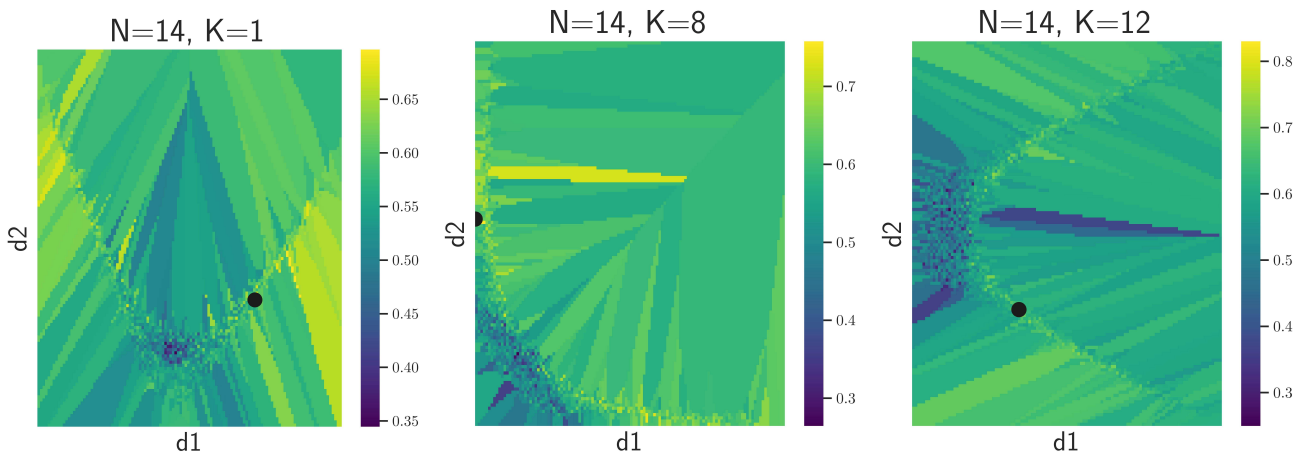


Figure 4: Estimation of the fitness landscape in the continuous domain from the NK embeddings. The location of the optimal solution is indicated with a black circle.

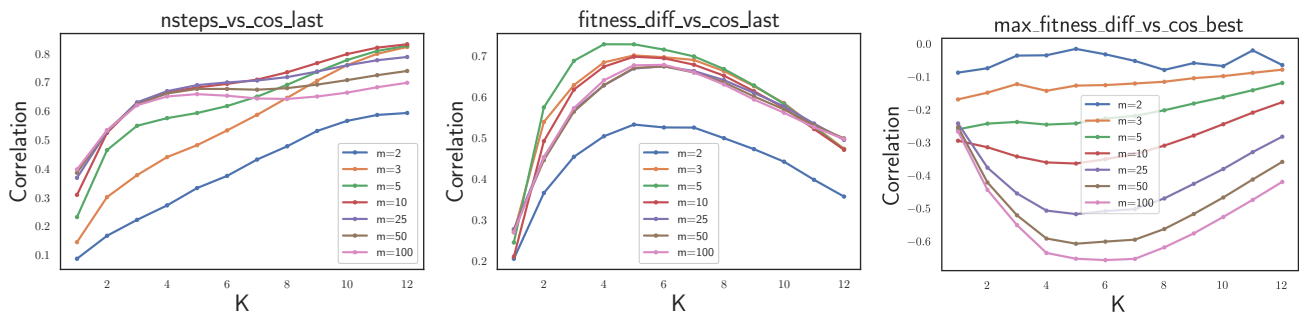


Figure 5: Correlation between the distance, in the embedding space, from an initial point to its local optimum and: (left) the number of steps performed by the adaptive walk to reach the local optimum, (center) the fitness difference between the starting point and the local optimum, (right) the fitness difference between the starting point and the global optimum.

the previously-discussed correlations. Similarly, it increases with

K , although it does not seem to depend on the embedding dimensionality. Regardless of the dimension of the embeddings, the distances of the points to their corresponding local optima is ranked in a similar way with respect to the size of the basin of attraction.

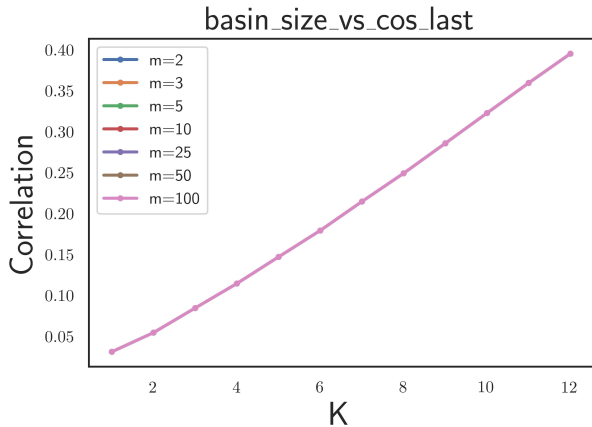


Figure 6: Correlation between the distance, in the embedding space, from an initial point to the size of the local optimum's basin of attraction.

We analyze below the relationship between the average distance of each solution to its (14) neighbors $cos_neighbors$, and the average distance of each solution to its local optimum cos_best . Figure 7 shows this information for different K values and $m \in \{2, 10, 100\}$. Analyzing in detail the distance to the neighbors is important since neighboring solutions are the set of candidates for the local search algorithm. While in the original domain the distance to neighbors is fixed to 1, it is not clear how this distance is in the embedding space. It can be seen in Figure 7 that, for small K values, a solution gets closer to its neighbors in the embedding space as it is further away from the local optima. As K increases, and the number of local optima also increases, the average distance of a solution to its neighbors increases relatively to the distance to its local optimum. This effect is visible for all the embedding dimensions. However, the transition is smoother for $m = 2$ and much more abrupt for $m = 100$.

5.4 Optimizing in the Embedding Space

An open question is how the difficulty of optimizing the NK model in the original binary space translates to a search in the embedding space, and whether this difficulty is also associated to the ruggedness of the original landscape, as defined by K . To look into this issue, we consider the surrogate function $\tilde{F}(x)$ that assigns to each possible continuous solution x , the fitness value of the closest point in the embedding space with a pre-image in the corresponding NK landscape. To optimize this continuous function, we use a variant of the covariance matrix adaptation evolution strategy (CMA-ES) [16], with an increasing population size restart [6] (with an increasing factor of $ipop = 2$), and the covariance matrix constrained to be diagonal [36]. This configuration was selected for efficiency and after preliminary experiments. The stopping condition for CMA-ES was to reach a maximum number of 500 function evaluations.

We evaluate the relative fitness deviation of the solution returned by CMA-ES to the global optimum; i.e., $\frac{\tilde{F}(x^*) - F(\hat{x})}{F(x^*)}$ where \hat{x} and

x^* are the optimum, and the best solution found by CMA-ES, respectively. As a second measure of performance, we compute the average rank of the solution found by CMA-ES (in terms of its fitness value), where the optimum has a rank of 0 and the solution with the lowest fitness has rank of $2^{14} - 1$. CMA-ES was executed 30 times for each of the 50 instances corresponding to models with $K \in \{1, \dots, 12\}$, and for embedding dimension $m \in \{2, 3, 5, 10\}$. In total, $30 \times 50 \times 12 \times 4 = 72\,000$ runs of CMA-ES were performed. For the experiments, we used the Optuna library [2].

Figure 8 reports that, as K increases, the relative deviation to the optimum of the solution found by CMA-ES also increases. This effect is more noticeable for $m = 2$ and less clear for higher dimensions $m \geq 5$. This result suggests that embedding spaces indeed capture the increase of difficulty induced by higher K values, although this effect also depends on the embedding dimensionality. Another relevant finding from Figure 8 is that, as the embedding dimensionality increases, so does the approximation quality of the solution found by the optimizer. We speculate this could be due to the fact that, in higher dimensions, it is more likely that the basin of attractions of two different solutions are communicating. In two dimensions, the optimal solution could be isolated in a region completely surrounded by lower-quality solutions.

Figure 9 reports the average rank of the solution found by CMA-ES as K increases. This figure provides a different perspective of the algorithm behavior. Instead of increasing, the average rank of the solution decreases with K until it reaches a plateau, at different values for different m . This indicates that, even if CMA-ES is not able to reach the optimal solution, it often identifies high-quality local optima.

5.5 Discussion

We summarize the findings from our experimental analysis by revisiting the questions addressed at the beginning of this section.

Firstly, in embedding landscapes, the neighborhood relationship determined by 1-bit flip does not hold. The solutions that are closest to a solution x in the embedded representation do not necessarily include the neighbors in the original binary domain. The location of solutions in the embedding space is more related to whether solutions are local optima, or not. This organization is consistent with the intuition behind the construction of word embeddings, where words with a similar “semantics” are grouped closer in the embedding space. In our model, the primary semantics relates to local optimality.

Secondly, for two dimensions, we corroborated that solutions are distributed in a boomerang-shaped form where local optima are more likely to be located in the arms. This distribution changes as the proportion of local optima increases with K . Although it is difficult to determine whether the embedding landscape is smoother than in the original domain, we have seen that the fitness-distance correlation is strongly negative, and that it achieves the highest values for embeddings of high dimensionality. We have also shown that it is indeed possible to conduct the optimization process in the embedding space and observed that, for medium and large K values, the average rank of the best-found solutions is not affected by the ruggedness of the original landscape.

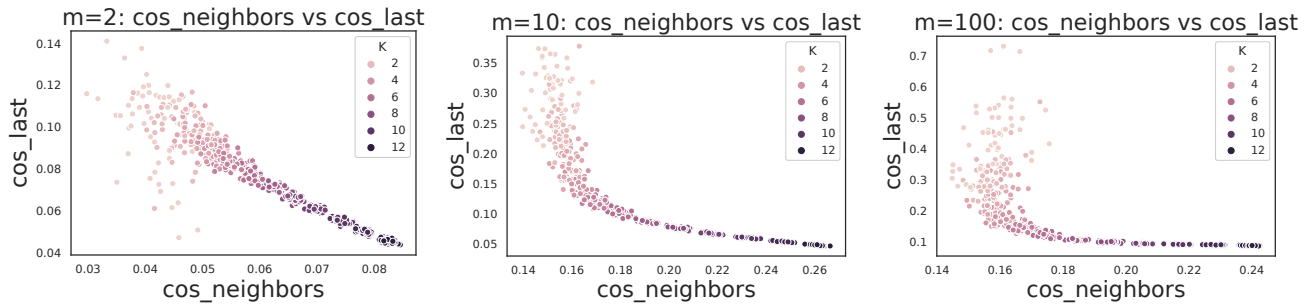


Figure 7: Relationship between the average distance of a point to its neighbors and the distance of a point to its local optimum for different values of K and $m \in \{2, 10, 100\}$.

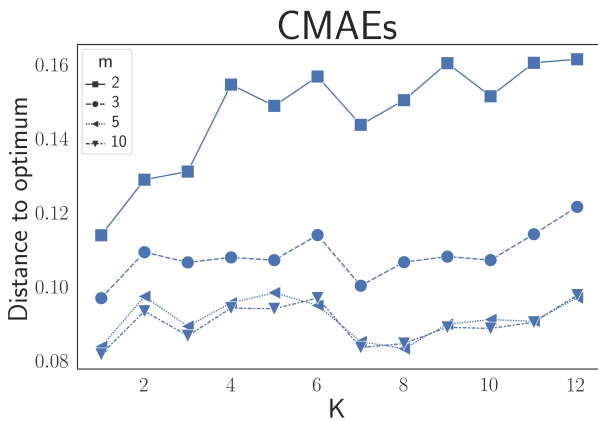


Figure 8: Relative distance to the optimal fitness value for the different embedding sizes as K increases.

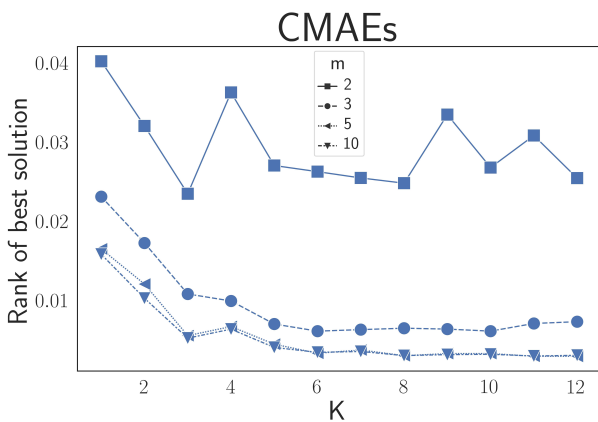


Figure 9: Rank, in terms of the fitness value, of the best found solution for the different embedding sizes as K increases. Rank 0 corresponds to the optimum. Ranks are normalized by the search space size (2^{14}).

6 CONCLUSIONS

In this paper, we proposed to approach NK landscapes from the perspective of neural embeddings. We addressed the question of how to create embeddings for NK models, and showed that these representations have as a characteristic feature that non-local solutions and local optima are located in different areas of the embedding space. We also corroborated that the distance between solutions in the embedding space is well correlated with the fitness difference between any two solutions and with the number of steps to reach local optima.

The transformation of the original search space to the continuous domain opens new perspectives that are worth being explored. We showed that it is possible to transform the search for the optimal solution from a binary to a continuous search space. This continuous space could be used to learn a model on the continuous representation that serves as a surrogate, or to infer properties of some NK solutions based on the proximity they have in the embedding space. One important direction for future research is to evaluate embedding spaces constructed using only a subset of solutions. In principle, it would be possible to apply similar methods for creating a set of paths from which learning the embedding spaces. However, a requirement would be to have a good covering of solutions from different areas of the search space. Another direction that deserves consideration is how to learn an inverse mapping from the continuous representation to the binary domain (e.g., using a neural network or a simple regression approach). Such inverse mapping would allow us to evaluate if perturbations to high-fitness solutions in the embedding representation can be mapped back to other high-fitness solutions in the binary representation. Finally, the strategies for the construction of the embedding spaces could be applied to other discrete optimization problems.

ACKNOWLEDGMENT

R. Santana acknowledges support by the Basque Government, Spain (project KK-2020/00049 through the ELKARTEK program and IT 124419), and by the Spanish Ministry of Economy and Competitiveness MINECO, Spain (projects TIN2016-78365-R and PID2019-104966GB-I00).

REFERENCES

- [1] H. Aguirre and K. Tanaka. Genetic algorithms on NK-landscapes: Effects of selection, drift, mutation, and recombination. *Applications of Evolutionary Computing*, pages 131–142, 2003.
- [2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.
- [3] U. Alon, M. Zilberstein, O. Levy, and E. Yahav. code2vec: Learning distributed representations of code. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.
- [4] M. Antoniak and D. Mimno. Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119, 2018.
- [5] E. Asgari and M. R. Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS one*, 10(11):e0141287, 2015.
- [6] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of The IEEE Congress on Evolutionary Computation (CEC-2005)*, volume 2, pages 1769–1776. IEEE, 2005.
- [7] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.
- [8] T. Bepler and B. Berger. Learning protein sequence embeddings using information from structure. *CoRR*, abs/1902.08661, 2019.
- [9] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.
- [10] J. Buzas and J. Dinitz. An analysis of NK landscapes: Interaction structure, statistical properties and expected number of local optima. *Evolutionary Computation, IEEE Transactions on*, 18(6):807–818, 2014.
- [11] F. Chicano, F. Daolio, G. Ochoa, S. Vérel, M. Tomassini, and E. Alba. Local optima networks, landscape autocorrelation and heuristic search performance. In *International Conference on Parallel Problem Solving from Nature*, pages 337–347. Springer, 2012.
- [12] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer. Problems with evaluation of word embeddings using word similarity tasks. *CoRR*, abs/1605.02276, 2016.
- [13] M. Fyvie, J. A. McCall, and L. A. Christie. Towards explainable metaheuristics: PCA for trajectory mining in evolutionary algorithms. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 89–102. Springer, 2021.
- [14] Y. Goldberg and O. Levy. word2vec explained: Deriving Mikolov et al.’s negative-sampling word-embedding method. *CoRR*, abs/1402.3722, 2014.
- [15] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.
- [16] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.
- [17] S. Herrmann, G. Ochoa, and F. Rothlauf. Communities of local optima as funnels in fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 325–331, 2016.
- [18] T. C. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L.J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192, 1995.
- [19] S. Kauffman. *Origins of Order*. Oxford University Press, 1993.
- [20] K. Klemm, J. Qin, and P. F. Stadler. Geometry and coarse-grained representations of landscapes. In *Recent Advances in the Theory and Application of Fitness Landscapes*, pages 153–176. Springer, 2014.
- [21] J. J. Lastra-Díaz, J. Goikoetxea, M. A. H. Taieb, A. García-Serrano, M. B. Aouicha, and E. Aguirre. A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. *Engineering Applications of Artificial Intelligence*, 85:645–665, 2019.
- [22] A. Liefvooghe, B. Derbel, S. Verel, M. López-Ibáñez, H. Aguirre, and K. Tanaka. On Pareto local optimal solutions networks. In *International Conference on Parallel Problem Solving from Nature*, pages 232–244. Springer, 2018.
- [23] B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150. San Mateo, CA: Morgan Kaufman, 1991.
- [24] S. Maneewongvatana and D. M. Mount. It’s okay to be skinny, if your friends are fat. In *Center for Geometric Computing 4th Annual Workshop on Computational Geometry*, volume 2, pages 1–8, 1999.
- [25] L. Manzoni, D. Jakobovic, L. Mariot, S. Picke, and M. Castelli. Towards an evolutionary-based approach for natural language processing. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 985–993, 2020.
- [26] D. P. Melidis, B. Malone, and W. Nejdl. dom2vec: Unsupervised protein domain embeddings capture domains structure and function providing data-driven insights into collocations in domain architectures. *bioRxiv*, 2020.
- [27] K. Michalak. Low-dimensional Euclidean embedding for visualization of search spaces in combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 23(2):232–246, 2019.
- [28] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
- [29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
- [30] S. Nowak and J. Krug. Analysis of adaptive walks on NK fitness landscapes with different interaction schemes. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(6):P06014, jun 2015.
- [31] G. Ochoa, M. Tomassini, S. Vérel, and C. Darabos. A study of NK landscapes’ basins and local optima networks. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO-2008)*, pages 555–562. ACM, 2008.
- [32] G. Ochoa, S. Verel, and M. Tomassini. First-improvement vs. best-improvement local optima networks of NK landscapes. In *International Conference on Parallel Problem Solving from Nature*, pages 104–113. Springer, 2010.
- [33] M. Pelikan. Analysis of estimation of distribution algorithms and genetic algorithms on NK landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2009*, pages 1033–1040. ACM, 2008.
- [34] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, 2014.
- [35] I. Roman, R. Santana, A. Mendiburu, and J. A. Lozano. Sentiment analysis with genetically evolved Gaussian kernels. In *Proceedings of the 2019 on Genetic and Evolutionary Computation Conference*, pages 1328–1336. ACM, 2019.
- [36] R. Ros and N. Hansen. A simple modification in CMA-ES achieving linear time and space complexity. In *International Conference on Parallel Problem Solving from Nature*, pages 296–305. Springer, 2008.
- [37] R. Santana. Reproducing and learning new algebraic operations on word embeddings using genetic programming. *CoRR*, abs/1702.05624, 2017.
- [38] R. Santana. Semantic composition of word-embeddings with genetic programming. In *Heuristics for Optimization and Learning*, pages 409–423. Springer, Cham., 2021.
- [39] M. Tomassini, S. Vérel, and G. Ochoa. Complex-network analysis of combinatorial spaces: The NK landscape case. *Physical Review E*, 78(6):066114, 2008.
- [40] S. Verel, P. Collard, and M. Clergue. Where are bottlenecks in NK fitness landscapes? In R. Sarker, R. Reynolds, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, editors, *Proceedings of the 2003 Congress on Evolutionary Computation CEC-2003*, volume 1, pages 273–280. IEEE, 2003.
- [41] S. Verel, G. Ochoa, and M. Tomassini. Local optima networks of NK landscapes with neutrality. *Evolutionary Computation, IEEE Transactions on*, 15(6):783–797, 2011.
- [42] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.
- [43] M. Zhai, J. Tan, and J. D. Choi. Intrinsic and extrinsic evaluations of word embeddings. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 4282–4283, 2016.