Laboratory of Economics and Management
Sant'Anna School of Advanced Studies

Piazza dei Martiri della Libertà, 33 - I-56127 PISA (Italy)
Tel. +39-050-883-341 Fax +39-050-883-344
Email: lem@sssup.it Web Page: http://lem.sssup.it

LEM
Working Paper Series

Pseudo-NK:
an Enhanced Model of Complexity

Marco Valente
Università dell'Aquila

2008/26 **November 2008**

ISSN (online) 2284-0400

Pseudo-NK: an Enhanced Model of Complexity*

Marco Valente†

Abstract

This paper is based on the acknowledgment that NK models are an extremely useful tool in order to represent and study the complexity stemming from interactions among components of a system. For this reason NK models have been applied in many domains, such as Organizational Sciences and Economics, as a simple and powerful tool for the representation of complexity. However, the paper suggests that NK suffers from un-necessary limitations and difficulties due to its peculiar implementation, originally devised for biological phenomena.

We suggest that it is possible to devise alternative implementations of NK that, though maintaining the core aspects of the NK model, remove its major limitations to applications in new domains. The paper proposes one such a model, called pseudo-NK (pNK) model, which we describe and test. The proposed model appears to be able to replicate most, if not all, the properties of standard NK models, but also to offer wider possibilities. Namely, pNK uses real-valued (instead of binary) dimensions forming the landscape and allows for gradual levels of interaction among components (instead of presence-absence). These extensions provide the possibility to maintain the approach at the original of NK (and therefore, the compatibility with former results) and extend the application to further domains, where the limitations posed by NK are more striking.

Keywords: NK model, Simulation models, Complexity, Interactions.

JEL-classification: C15, D20, D83, L23, O31, O32.

*I wish to thank Tommaso Ciarli for valuable comments and suggestions. Usual caveats apply.

†Università dell'Aquila, P.le del Santuario 19, 67040 Roio Poggio (L'Aquila), Italy. valente@ec.univaq.it.

1 Introduction

Borrowing ideas developed for a specific purpose and applying them to a completely different domain is a widely used and generally successful operation, to such a point that one may sustain copying and mixing different ideas is the main driver of human creativity. Economics, just to make an example, has "stolen" the concept of equilibrium from classical physics and, more recently, the concept of evolution from biology.

The use of metaphor and concepts developed in different domains, however, runs the risk of mis-adaptation. Though the core of the original idea can be useful in the new domain, it is frequently necessary to make some adaptations, removing some parts that are no longer necessary, and adding new elements necessary for its novel application. Witness, for example, the attempts to imitate the wings' design of birds to build flying machines. As long as the imitations were too close (moving wings, for both functions of supporting the weight and propelling), the attempts continued to fail. Only when the core idea was revised, separating the function of propelling from the function of supporting, then the (revised) imitation succeeded.

Concerning the study of complexity, biologists have developed the NK model (Kauffman, 1993) representing, in very stylized and elegant way, the effects of fitness improving mutations. NK has been particularly successful because it provides a simple instrument to generate an abstract representation of a problem (the probability of survival of a species) that, contrary to other modelization techniques, could be easily tuned to make the problem harder or simpler. The representation of the problem relies on the core of complexity, that is, interdependency, using (ironically) an extremely simple representation. In other terms NK can represent complexity as a pure product of interdependency, since the rest of the model makes use of a large number of uniformly distributed random numbers, therefore avoiding any spurious property. The result is that NK fitness landscapes show statistical properties depending on the interaction levels only, which are very robust to the choice of the set of random numbers used (see, e.g. Weinberger, 1991; Durret and Limic, 2003; Skellett *et al.*, 2005; Kaul and Jacobson, 2006).

In a biological perspective NK is very useful because it is well known that genes' effects on phenotypes are strongly influenced by their interactions. NK poses the interaction explicitly at the center of the analysis, and carefully avoids to make any further assumption, being able to derive interesting results concerning the expected characteristics of a species, its pattern of evolution, and even the reasons for the spontaneous emergence of modularity (e.g., Wagner and Altenberg, 1996; Altenberg, 1995).

However, biologists are not the only ones interested in the modelization of complexity through interaction. Economists and scientists of the organizations have had a long-time interest on the study of the effects of interaction (Simon, 1969). Therefore, many researcher from these fields have adopted NK as their instrument of choice to represent and study properties of organizational structures, technological innovation, etc. Originally devised as a metaphor for how nature deals with complex problems, NK-inspired models have been used to study artificial systems, organizations, technological developments, industrial dynamics, and much more. In such applications typically NK represents a complex problem, where performance depends on the interactions among several components. Simulated agents (say a firm or a person) is engaged in solve the problem on the base of local and myopic information, and the researchers are interested in assessing the results as a function of the complexity of the task and the limited capacities of the agent. Simulation models in Economics and Management have been eager adopters of variations of NK models, with dozens of papers in top journals (see, for example, Levinthal, 1997; Frenken *et al.*, 1999; Kauffman *et al.*, 2000; Rivkin and Siggelkow, 2002; Lenox *et al.*, 2006).

This work is based on the assumption that, however successful, the diffusion of NK, and the quality of the results produced so far have been hampered by a mis-adaptation of this tool in its passage from biology to other domains. Tellingly, many models use a NK-inspired settings to study the expected performance of, e.g., different types of organizations, but no model exists, to our knowledge, integrating in a single model a NK-like complex environment *and* other activities, like production, sales, etc. For example, it would be logical to plug a NK fitness landscape representing the space of technological possibilities into an economic model representing innovating firms. However, such models are very difficult to implement. The reason, as would be argued in the following, lies in the heavy constraints posed by the original NK implementation when applied to a different domain. For example, NK models rely on random mutations as one may expect to find in natural environment, but human organizations are likely to include at least a bit of intentional, purpose-driven behavior, however limited by informational constraints. Modellers willing to include intentional strategies of search with NK would need to know where the optimum is located, and possibly also have the opportunity to determine a specific location. Given the structure of NK models this is impossible, unless one limits himself to extremely simple landscapes or to heavily modified versions lacking the core of NK properties.

The diffusion of NK is making more evident its limitations. For example, a recent paper proposes an extension of NK relaxing some of its strongest assumptions (Li *et al.*, 2006). The goal of the present paper is to propose an even more radical alternative implementation of the core features of NK in such a way to make the model more flexible and adapt for the novel applications it is increasingly put at work. The next section discusses the core elements and properties of NK, concluding that there are severe shortcomings for its application in fields different from biology. The third section proposes a possible alternative, describing what we call a *pseudo*-NK model (pNK), that is, a model that replicates all the relevant properties of NK but removes, or at least relaxes, the major shortcoming identified. Note that the implementation proposed for pNK can also be considered as an instance of a class containing many other variations of the proposed model, possibly with better characteristics than those proposed here. The fourth section put the proposed pNK model to test. We present an extended series of experiments in order to evaluate the capacity of pNK of adequately provide the original properties of NK without the limitations discussed in section two. The concluding section summarizes the paper and indicates possible directions to explore more throughly the properties of pNK.

2 NK models: advantages and limitations

A NK model[1] can be considered as composed by two, distinct, components: a problem specification and a search algorithm scanning the space of the potential solutions. The problem is composed by a set of potential solutions represented as binary strings, each associated to a *fitness* value, that is, the pay-off of that solution. The search algorithm consists in a routine meant to scan the space of solutions starting from a (normally randomly chosen) initial string, or point in the N-dimensional binary space. The search routine is defined in terms of rules on how to move from any one point to the next. For example, the typical routine, originally proposed, consists in choosing randomly a string

[1] For reasons of space we skip a detailed description of NK models, providing only a short verbal summary of its basic mechanisms. For readers unfamiliar to NK models, who found insufficient the information in the text, formal definitions can be found one of the cited works.

by mutating one of the bits in the current string; in case the mutated string has higher fitness than the current one, the new routine is accepted, conversely, it is rejected. Applying the algorithm repeatedly it is generated a pattern in the space of potential solutions. The pattern terminates when the rule reaches a string from which all possible strings within reach are rejected. The search routines are evaluated according to the expected solution it is able to identify at the end of the pattern.

Two aspects make NK particularly attractive. Firstly, it is possible to determine how complex should be the space of solutions, or *fitness landscape*. Building a landscape with no or few interactions (represented by the value of K) it is possible to generate the equivalent of simple problems, while increasing K generates "harder" problems.

The second aspect of NK is the representation of the search algorithm. NK assumes a search to be local and myopic. Local because the search implies the impossibility to observe the space beyond the current neighbourhood. Myopic because it prevents collecting past information or predict future events, focusing on the immediate goal of just improving the current condition.

The two aspects (complexity through interaction, and local and myopic search) provide together a simplified, and therefore manageable, representation of many real world situations. NK allows the modeller to play the role of God in a mini-universe, setting the basic properties of a world inhabited by (again controlled) agents. In the following we analyse, firstly, the major advantages of NK and then its more prominent shortcomings for a large number of applications.

2.1 NK features

The most attractive aspect of NK models is the algorithm used to associate the fitness values to the strings representing the solutions. This algorithm mandates the generation of 2^{K+1} random values for each bit of a string (call them the *fitness contributions* for each element of the string), for a total of $N \times 2^{K+1}$ random draws. The fitness of a string is then obtained by averaging the fitness contributions from each bit of the string. The set of strings and their fitness define a *landscape*, a generic representation of problem with a pre-determined complexity level (the parameter K).

The earliest literature on NK models used a single search algorithm, representing a random search in the space of solutions by means of the so-called *one-bit* mutation. The search algorithm is based on the initial choice of a random solution, and a routine meant to find better and better ones (i.e. strings associated with higher fitness). At each step of the routine the fitness of the current solution is compared with the fitness of a string identical over all the elements but one, randomly chosen. If the mutated string shows higher fitness, it is adopted as a new string, otherwise, the search remains on its current location.

Applying the one-bit mutation search strategy to the set of strings whose fitness is determined as above for a given K, one can obtain several general properties. For example, it is possible to determine whether the search strategy can eventually reach the highest fitness string, or if it is bound to get stuck in a dominated string, a local maximum whose fitness is not the highest one, but that it is surrounded by strings with lower fitness. Other properties concern the average fitness of local peaks, their number and distribution.

With the application of NK models to domains different from biology, the blind, one-bit mutation strategy looked increasingly inadequate to represent the behaviour of human problem solvers. The most recent literature has proposed different search algorithms, more sensible for specific applications. For example, it has been proposed to consider search algorithms mutating more than one bit at any one time. Obviously, changing the search routines modifies the very definition of local peak, since the mutation strategy determines the very definition of the neighbourhoods, and therefore the topology of the search space, i.e. which strings can be accessed from any given string. Changing the topology therefore alters the structure of local peaks. In theory, very "smart" routines are able to include in their neighbourhood the whole space of possibilities, though this, has been shown, eventually generates a trade-off between the level of expected final results and the time requested to reach a final state.

Although the diffusion of NK to different research applications has suggested modification of the research algorithm, the basic structure of the landscape (the set of strings and their associated fitness) has never been questioned, because it offers, up to date, the best way to represent a system composed by (partially) interdependent components. Given the growing interest in the study of complex, interdependent systems, NK has become the *de facto* standard for such studies. Let's see in some detail the most attractive features NK, and, following, its major drawbacks.

2.2 NK advantages

The literature on complexity can be interpreted as concerned in two large classes. One class comprises studies on the properties of complex set or complex problems, such as chaotic functions or the traveller salesman problem. The second class studies features of problem solving tools, like genetic algorithms or neural networks.

The NK model is an attractive tool because provides the opportunity for the modeller to represents and control both aspects of problem solving: the degree of complexity of the problem and the degree of the skills available for its searching a solution. Modellers can therefore use NK to generate and assess the space formed by the two dimensions of problems' complexity and skills in solving strategies in order to represent both aspects of a real world system on a scaled down version. The NK model's attraction stems from the possibility to determine a sort of ratio between the relative skills of the problem solver and the relative difficulty of the task. In this case, it is no more relevant that the problem modelled is far simpler than real ones, since also the solution strategies modelled are far less sophisticated. Controlling both aspects one can expects that the properties of the set comprising the solutions generated in the model are similar to the set of the actual solutions generated in real systems with equivalent ratios of task difficulty to solving skills.

We can summarize the most prominent features of NK as follows.

- **Measuring complexity**. With the increasing popularity of the term, what exactly means "complex" is hard to define, and frequently this adjective is used as a byword for "don't understand". NK provides formal definitions distinguishing elements of the complex studies pertaining the environment and those related to the problem solvers, which is the necessary pre-condition to assess the effects of the interaction among the two domains. A problem, *per se*, cannot be neither hard nor simple, unless a solution method is, at least implicitly, specified. After all, a hard problem for someone can be a simple one for someone else. NK offers a simple and easy method to tune a desired level of complexity in respect of a given research strategy.

- **Complexity as interdependency**. NK explicitly identifies the source for complexity: interdependency. The higher the interaction among elements of a structure, the more difficult is to use "local" information (i.e. gathered within a short range) to gain general knowledge of the whole structure. While this concept is not new to the literature (e.g., Simon's work), NK offers a simple and intuitive way to implement interdependencies.

- **Intuitive fitness landscapes**. The most interesting feature of NK is that the modeller can tune a landscape from a smooth, single peaked, highly correlated landscape up to an extremely rugged, multi-peaked, uncorrelated one. This provides a very intuitive representation of how a fitness climbing strategy is likely to score, offering a powerful visual metaphor for the effects of different degrees of complexity.

- **Problem solving as local search**. The problem solving strategy to be applied by simulated agents on a NK fitness landscape was originally proposed as a one-bit random mutation, representing mutations of genes of incumbent species. However, different search strategies can easily be devised to represent the behaviour of different types of agents, giving the possibility to "tune" the smartness of agents, for example when NK is used as a metaphor of intentional agents. In any case, the fitness climbing strategy, based on local information and generating a fitness increasing path on the landscape, is a highly attractive metaphor for many types of problem solving strategies.

2.3 NK limits

Given the power of NK models in representing a complex problem, it is natural that many researchers were tempted to plug a NK system into larger and more comprehensive representations. For example, a model studying the economics of technological innovation containing, say, equations representing production, investment, demand etc. would be greatly enhanced by using NK to represent the technological space and the R&D efforts by firms.

However, this and many similar attempts in organisational theory, biology, etc., have been frustrated by a few, severe limitations of NK models. In the following we review and comment the most prominent difficulties preventing a wide-spread use of NK model not as stand-alone package, but when it is plugged into a wider context.

- **Memory limitation**. The computational representation of a NK model is rather straightforward, such that even relatively inexperience programmers can easily implement programs for NK models. However, the construction of even relatively small NK landscapes require a huge memory space, if N and, especially, K are large (say above 20 or 30). We mention this problem because of its popularity, but, in effect, it is the least serious. In fact, although realistic problems involving hundreds of interdependent components would require, in theory, an amount of memory several dimensions larger than those available on any computer, there are many programming tricks that can simulate such landscapes on a normal computer. For example, a complete NK representation of a system composed by $N = 100$ elements, each interacting with $K = 50$ other elements would require about 1.13×10^{17} memory locations, or more than 6 million gigabytes, well beyond the limits of every computer. The technical solution is to implement a partial representation of the landscape, using only the memory necessary to calculate the tiny portion of the landscape actually explored in a search path. Still, the combinatorial requirements of NK landscapes prevents the full exploration of the landscape to determine, for example, the identification of the maximum fitness string or the full representation of the distribution of the local peaks.

- **Binary interdependence**. More serious is the impossibility to determine a *degree* of interdependence. In a NK model two dimensions are either interdependent or not, without the possibility to tune gradually interdependency levels. Worse, the actual effects of interdependence can be determined only statistically, and only *ex-post*, once the landscape has been built, given its dependence on random draws (at least in theory, since in practice it is a highly challenging task). In fact, a given link can pose a more or less serious obstacle to a fitness climbing strategy, depending on the set of random values. There is therefore no way to pre-determine from the outset the importance of an interdependence link between two elements, or, at least, determine the relative importance of the different links.

- **Binary dimensions**. The components representing the module of a system in the NK metaphor are binary variables, whose values are restricted to the $\{0, 1\}$ set. Though formally any real valued variable can be represented by binary strings, the complication requested in practice prevents the adoption of multi-valued (real or discrete) variables. In fact, what matters in a NK representation is the Hamming distance between two strings. For example, the strings 01111 is on the opposite location than 10000. Therefore, a binary representation if a multi-valued variable (say, assumed to represent natural numbers) would not work: in the example, we would have that number 17 is very *far* from 18, while the latter is *close* to 19. This restriction forces modellers to apply a NK model only to components made of two states, instead to implementing a complex system made of multi-valued variables.

- **Randomness**. NK models allow to determine the statistical properties of the landscape, but not its overall shape. That is, we can control the average frequency and levels of local peaks, but their exact location depends on random draws, changing at each generation of a new landscape. Though this is not a relevant issue for the abstract study of complexity which NK is originally directed to, it becomes a problem when the modeller wants to study a specific complex environment. That is we would like to know *where* the local peaks are located, in order to properly assess the behaviour of the simulated agents engaged in a local and myopic search.

- **Problem specification**. In many cases researcher have some knowledge of the real world problem they want to model, and they would like to include these aspect in their studies. For example, they may have some knowledge of the relative strength of interaction. Or, they have some information concerning the location, distribution of values and overall density of local peaks. NK makes impossible to use such information to generate a landscape including the available information, forcing the problem space to be totally random (but for the interdependency structure), since, otherwise, the properties of the system are lost.

- **Difficulty to use in agent-based models**. Given the highly attractive features of NK one may have expected to find many applications in models where agents representing, for example, firms engaged in technological competition use NK to implement the technological environment, along, say, a demand and a capital sector. However, every attempt of this sort has been frustrated, and most of the models, though using NK as a metaphor for a given complex environment, keep clear from implementing it together with the other aspects of the model. The reason is that NK models provide their powerful results in *statistical* terms, as averages of many individual searches or over many starting points, but a very large volatility is observed concerning the single experiments. For example, one may assess the expected fitness of agents moving on a given landscape, but the actual results produced by a single search run (i.e., the actual fitness pattern obtained by any given agent on a given landscape), will vary considerably depending on the initial step and the randomness of the landscape. Worse, even for relatively large landscapes, any search by agents is made by very few fitness-improving steps in between of long series of failed mutations. Timing the rate of landscape exploration with the rate of other events in an integrated model becomes therefore extremely difficult.

  Considering the combined effect of the volatility and type of fitness increasing path generated make NK practically unusable to represent entities involved a search supposed to produce continuous, gradual improvements co-ordinated with other activities.

- **Representation of fitness landscapes**. We noticed above that NK models provide an intuitive representation of the difficulty of a problem solving task, but this comes at the price of easy misinterpretations of the actual content of the model. In fact, when speaking of "valleys" and "hills" of a landscape, the image one has in mind is of a geographical space (a plane with co-ordinates (X, Y)) with the third dimension representing fitness. Therefore, one may feel entitled to image a fitness climbing strategy as moving around climbing the hill closest to the current position. This metaphor is reliable up to a crucial point: NK dimensions are only binary, and therefore there are only two values available for any one dimension, while the power of NK properties stems from the use of many dimensions, something that our experience and skills are not trained to deal with. The difference between an environment with a small number of multi-valued dimensions and one with a large number of binary dimensions is critical. For example, the shape of the landscape depends in NK on the algorithm used to represent the search strategy. What may be a rough shape or a smooth one depends on the "length" of the steps allowed, measured by the number of dimensions one may mutate at once. While this is not a problem when fixing the mutation strategy (as in the original proposal), this problem is relevant in the (many) application where modellers tweak the search strategy for specific purposes.

In conclusion, in this section we sustained that NK deserves its popularity because of several factors that make it uniquely suitable to study the result of an arbitrary "intelligent" search strategy applied on an arbitrary "complex" problem. However, NK is far from an ideal tool. Mainly because of the original context it was supposed to be used for (evolution of a metaphorical biological species), it carries on a number of limitations and problems when transferred in different contexts, as it is increasingly the case.

The following section is devoted to propose an alternative implementation close to NK that, in our intention, is a step toward a model retaining all the positive features of NK while relaxing the constraints its implementation poses.

3 Design of a Pseudo-NK model

In this section we propose a model, called *pseudo-NK* (pNK) model, that replicates the same useful properties of the original NK model but lacks the limitations listed above. In particular, pNK offers the following features:

- **Functional representation of fitness**: The fitness function is defined as a deterministic function of a multidimensional vector. Therefore, it can be implemented as a routine, without need to store any (large or small) data set, simplifying the coding of very fast implementation for even large landscapes.

- **Multidimensional real-valued landscape**: the landscape of pNK is composed by the set of $\vec{x} = \{x_1, x_2, ..., x_N\} \in \Re^N$, with fitness represented by a real-valued function $f(\vec{x})$. Both domain and co-domain can be freely determined by the modeller.
- **User determined maximum and landscape's overall shape**: $f(\vec{x})$ has a maximum[2] in a point $\vec{x}^*$ (such that $f(\vec{x}^*) \geq f(\vec{x})$) determined by the user. The shape of the landscape is a well-behaving function, so that the modeller can easily evaluate every single point of the landscape, determining, for example, areas of local peaks, probability of a random search strategy to end up in the global optimum, etc.
- **User determined interdependency**: for any given couple of dimensions i and j the user can set a varying degree of interdependence $a_{i,j}$, ranging from full independence to maximum interdependence. But it is also possible to define intermediate levels of interdependency, so as, for example, to define landscapes where a dimension depends strongly on some dimensions and weakly on others.

The definition of interdependency used in pNK remains the same as the one used in the NK model: dimension i is dependent on dimension j if, for at least some value of the other N-2 variables, the derivative of $f(\vec{x})$ in respect to x_i changes signs for different values of x_j. It is worth to notice that in NK (and pNK, too), interdependence does not simply imply that modifications x_j affects how x_i impact on the fitness function $f(\vec{x})$. In fact, it may be possible that such influence is strictly monotonic, that is, for different x_j we observe that x_i has different impacts on the overall fitness value, but they always have the same sign. Though in this case we do have interdependency, in a sense, this is not affecting the possibility to identify the optimal value for x_i independently from x_j. Only when $\frac{\partial f(\vec{x})}{\partial x_i}$ changes sign for different values of x_j, than the fitness-optimizing value of x_i is a function of x_j. Only in this case any hill-climbing strategy based on varying x_i only is liable to be stuck in a local peak determined by the specific value of x_j.[3]

In the following we describe a fitness function providing the desired properties, which will be used in the rest of the paper for a series of tests, replicating the results of standard NK models and extending them to include potentially useful new features.

3.1 Implementation of pNK

There is a whole class of mathematical functions providing the properties discussed above, and each of them may suit particular needs. Here we propose one of these functions, as an example of how pNK may be implemented.

pNK, as NK, consists of a fitness function defined on a set of N variables and a search algorithm. The fitness function proposed here for pNK borrows heavily from the NK implementation, with three notable differences. Firstly, it considers real-valued variables (instead of binary); secondly, it is a deterministic function, rather than stochastic; thirdly, it allows for different levels of interdependence, instead of presence/absence.

The overall fitness value of a point of the landscape domain (i.e. a point of $\Re^N$) is, as in NK, the average of N fitness contributions for each of the variables:

[2]Actually, it is possible to extend the model to admit multiple global maxima, though we will ignore this aspect.

[3]We wish to thank Luigi Marengo for having raised this relevant point.

$$f(\vec{x}) = \frac{\sum_{i=1}^{N} \phi_i(\vec{x})}{N} \quad (1)$$

where $\phi_i(...)$ is the fitness contribution function for dimension i. While in NK ϕ_i is a random value, in NK this is a deterministic function defined as:

$$\phi_i(\vec{x}) = \frac{Max}{(1 + |x_i - \mu_i(\vec{x})|)} \quad (2)$$

where Max is a user-determined parameter indicating the maximum of the function. ϕ_i is a decreasing function of the distance between the variable's value and another function $\mu_i(\vec{x})$, defined as:

$$\mu_i(\vec{x}) = c_i + \sum_{j=1}^{N} a_{i,j} x_j \quad (3)$$

The values μ_i's define a sort of "target" that, when hit by x_i, determines the maximum level of contribution of the variable to the overall fitness function. However, the value of x_i maximising ϕ_i may not be the most desirable, concerning the overall fitness value. In fact, x_i influences also all contributions ϕ_j for the variables whose $a_{j,i} \neq 0$. Therefore, it is well possible that moving x_i to maximise ϕ_i actually decreases the overall fitness value because of the deterioration generated in other fitness contributions ϕ_j's where $a_{j,i} \neq 0$.

The fitness function here proposed allows for ample flexibility. In particular, it is possible to determine features of the landscape that are not available in NK models:

- **Set maximum fitness**. Simply setting Max determines the maximum value of the fitness function.
- **Set the global optimum**. For any dimension i it is possible to compute c_i such that the maximum fitness is obtained at a desired point $\vec{x}^* : c_i = x_i^* - \sum_{j \neq i} a_{i,j} x_j^*$.
- **Set interdependencies**. Varying the values of $a_{i,j}$ it is possible to make more or less relevant the interdependency between two dimensions.

The first two properties are useful to exploit pNK in a context where the modeller is interested in determining a specific maximum fitness value at a specific point of the landscape[4]. We will, however, concentrate on the last property. Besides replicating the NK results with pNK, we will also test the effects of changing the degrees of interdependency, an option not available in NK.

Before reporting on the tests performed we describe below the definition of a search strategy in a pNK context.

3.2 Search strategy on *p*NK

Having modified the landscape's definition, we need also to provide the equivalent definition of one-bit mutation for a real-valued landscape. As we will see, this definition will be much closer to one's intuition of a search strategy.

On a real-valued fitness landscape the one-bit mutation strategy consists of the following steps:

[4] Also, it may be possible to shift the maximum fitness and/or the optimal point defining appropriate functions.

1. Choose randomly one dimension.
2. Make a step Δ in one direction on the chosen dimension.
3. If the fitness increases, move to the new point.
4. If the fitness decreases, stay in the same point[5].

The routine is repeated until no change of step Δ is able to produce a fitness increment, i.e. the routine reaches a (local or global) peak. Note that pNK offers also further flexibility on Δ, which may be set to different values to emulate a finer or rougher research strategy, as well as to be implemented as a function (e.g. a random value). For example, pNK allows to distinguish two different types of "long jumps" (Levinthal, 1997): one made of modifications of many dimensions, and another made of a large change on a single-dimension.

For our purposes we will consider Δ as a constant parameter represented by a small value, in effect determining only the discretization of the real-valued space supporting the fitness function. Tests performed changing the value of this parameter show that it does not affect the results presented.

In the next section we test the pNK model (under a few conditions) for replicating the same results of NK models and extending them as suggested above.

4 Testing *p*NK

We test the properties of pNK by running a set of simulations on a sub-set of all the possible configurations of the proposed model. We will limit to consider the $\phi(...)$ function as described above, so that the landscape presents a single global peak producing fitness of $MAX = 1$. For all the simulations presented below we set the optimal point at $\vec{x}^* = (100, 100, ..., 100)$. Moreover, for reasons of readability of the results, we also constrain the interdependency coefficients to be symmetrical: $a_{i,j} = -a_{j,i}$ and $0 \leq |a_{i,j}| \leq 1$. Obviously, these constraints may be removed to implement specific versions of pNK, but for the present purposes they allow an easier testing of the properties of pNK.

In the rest of this section we test, firstly, pNK implemented over two dimensions only ($N = 2$). This will provide a better understanding of the main difference with NK, since we have the possibility of a clear visual representation for the fitness produced by pNK, the results of search strategies, and their pattern through the landscape. Secondly, we will test pNK on several dimensions, testing pNK for the replication of the most relevant features of NK.

4.1 *p*NK fitness landscape

As a first exercise figure 1 shows a fitness landscape built on two dimensions, setting the global optimum at $x^* = (100, 100)$, maximum fitness to $MAX = 1$, and a high, but not maximum, level of symmetric interdependency $|a_{i,j}| = 0.7$.

The landscape is composed by a single peaked surface with monotonically decreasing fitness for points farther from the optimum. However, points with the same distance from the optimum have different fitness. The surface is composed by four "ridges", separated by "valleys", leading to the global optimum. The angles of the ridges between themselves

[5] The case of neutrality is not relevant for our purposes, though it may be relevant for certain applications.

Figure 1: Fitness landscape for $N = 2$ with symmetric and relatively strong interdependency. Global optimum set at (100,100) with maximum fitness set to 1, and $|a_{i,j}| = 0.7$. A hill-climbing strategy based on one-dimensional mutation corresponds to moves parallel to one of the axes. If, as in the figure, the ridges of the landscape are diagonal to the axes, they represent areas of local optima for one-dimensional search strategies. In fact, any vertical or diagonal step would force to "step down" from the ridge resulting in a fall of fitness. Only if the ridges are parallel to the axes ($a_{i,j}| = 0$) a one-dimensional search strategy can walk on the ridges and will always reach the global maximum.

and in respect of the axes are controlled by the interdependency parameters $a_{i,j}$. The example shows a symmetric landscape, composed by orthogonal edges, built imposing the same interdependency level of X_1 on X_2 and of X_2 on X_1. This is obtained by setting $a_{1,2} = -a_{2,1}$; in the rest of the paper we will keep on assuming symmetric landscape in order to simplify the study of the model properties. The angles in respect of the axes are determined by the absolute values of the parameters $|a_{i,j}|$, in this case set to a large, but not maximum, value.

The next paragraph shows different landscapes for different values of interdependency.

4.2 Varying complexity of fitness landscapes

This paragraph shows how the coefficients $a_{i,j}$ affect the fitness landscape. Limiting the number of dimensions to two, we can represent the fitness as the dependent variable on three dimensional graphs, visualizing the shape of the landscape.

The graphs reported in figure 2 describe five different landscapes generated by pNK for $|a_{i,j}| = 0, 0.25, 0.5, 0.75, 1$. For $|a_{i,j}| = 0$ the two ridges leading to the global maximum are parallel to the axes. For $|a_{i,j}| = 1$ the ridges run on the diagonals, at 45 degrees in respect of the axes. For intermediate values they have increasing angles[6].

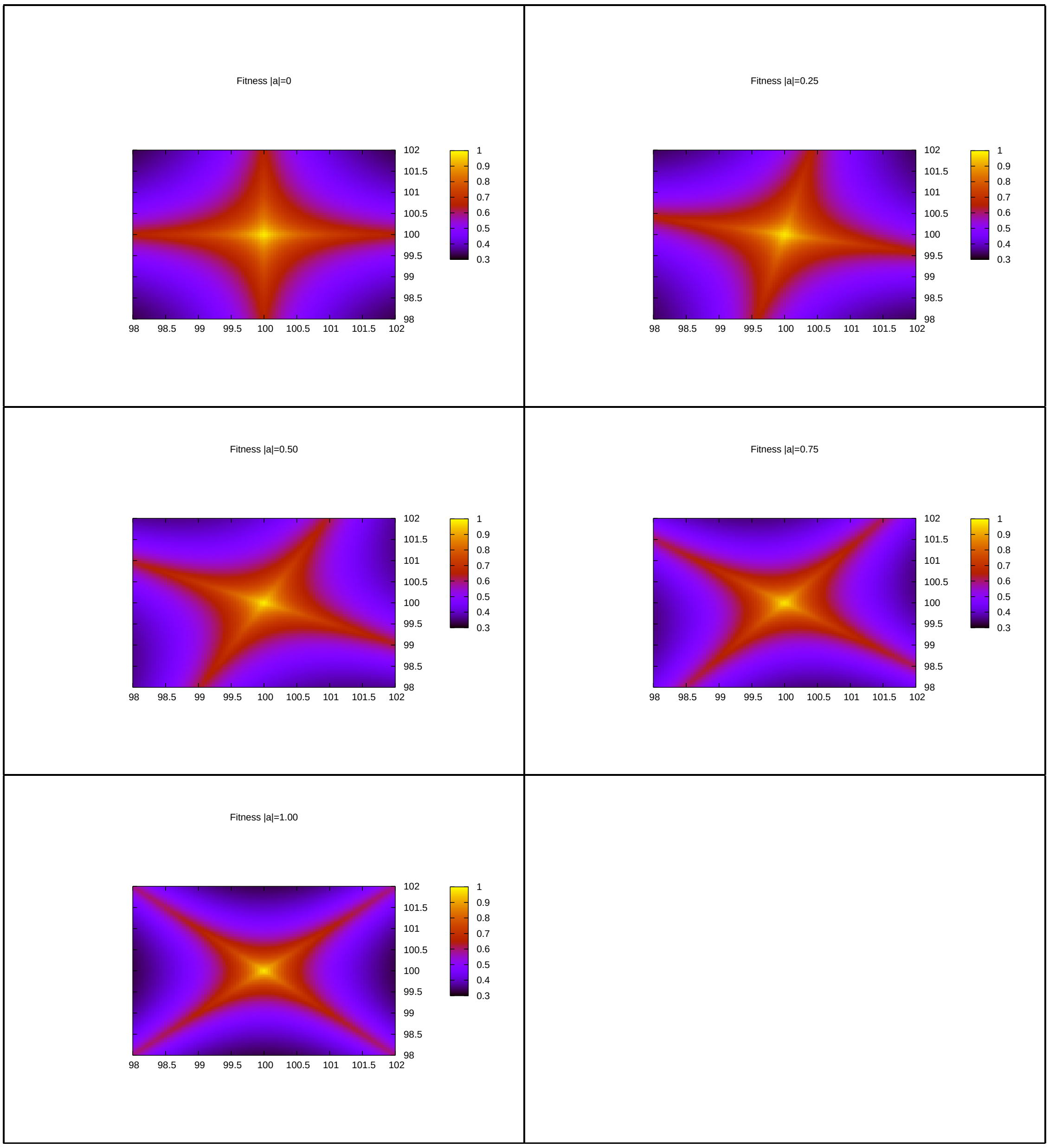

Figure 2: Fitness landscapes for $N = 2$ and different values of $|a_{i,j}|$.

The angle of the ridges in respect of the axes is crucial to determine the properties of pNK. A one-dimensional search strategy means to explore vertically or horizontally the points surrounding the currently held one. If the ridges run parallel to the axes, than such strategy will surely bring to the global optimum. Conversely, for landscapes with large angles, search patterns leading to points on the ridges risk to get stuck in local optima.

In fact, if the ridges are diagonal, moving parallel to the axes in the direction of the optimal point generates two opposite variation of fitness. Firstly, since the step will bring

[6]The simulation programs producing the results presented in the paper (code, parameterization and graphs) are implemented in Laboratory for Simulation Development-Lsd. The Lsd platform is available for download at `www.business.aau.dk\ ~mv\Lsd`. The code for the models, together for the instructions on their use, can be requested to the author.

away from the ridge, the fitness of the new point will tend to be lower than the starting point on the ridge. Secondly, since one direction of the step brings necessarily closer to the optimal point (because of the angle of the ridges), the fitness will tend to increase. The net effect is uncertain in general, and depends on which segment of the ridges the pattern has reached, besides their angle in respect to the axes. Given the functional form chosen, the slope of the ridges is gentler the farther away from the global optimum, and therefore, in these areas, the positive effect on fitness by getting closer to the maximum will be weaker. Conversely, segments of the ridges near the global optimum have steeper fitness and shallower valleys, and therefore it is more likely that the fitness loss caused by stepping off the ridge is smaller than the gain in getting closer to the maximum. In the next paragraph we explore how one-dimensional search strategies score on different pNK fitness landscapes.

4.3 Local peaks for one-dimensional search strategies on pNK landscapes

A one-dimensional search strategy consists in starting from a randomly chosen point and then selecting randomly among the four possible steps along the two dimensions. The step is accepted in case the new point has higher fitness, otherwise, the strategy remains on the current point. A local maximum is a point whose four surrounding points have all a lower fitness, and therefore the fitness increasing search strategy is trapped there.

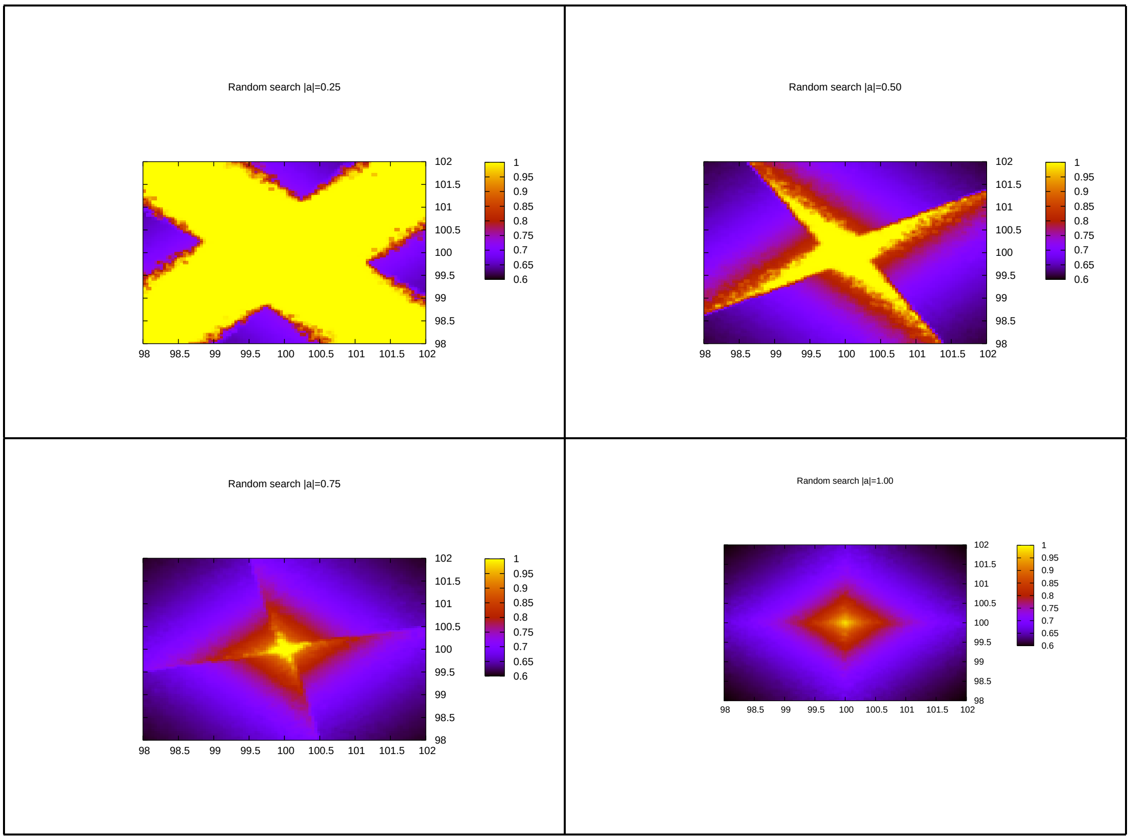

Figure 3: Final fitness produced by random one-dimensional search strategy for landscapes with $N = 2$ and different values of $|a_{i,j}|$. The graphs report the average final fitness in correspondence with each starting point. The average is computed from an expected number of about 5 searches from each point.

We ran 30,000 independent searches starting from randomly chosen points. Each search consists in steps performed by adding or subtracting Δ to a randomly chosen direction, and moving to the new point in case the change generates higher fitness[7]. For each search it is recorded the starting point and final fitness of the point eventually reached when the search terminates (i.e. local or global maximum). Graphs reported in figure 3 show the final fitness values for each search in correspondence of the initial point. The graph for $|a_{i,j}| = 0$ is not reported, since it consists of a uniform plane of value 1, meaning that all searches starting from every point of the landscape manage to reach the global maximum. The remaining cases show that, while $|a_{i,j}|$'s increase, the area of starting points bringing to the global optimum shrinks. Under the most challenging case ($|a_{i,j}| = 1$) we obtain a *Fuji*-like figure: only the searches starting close to the global optimum are able to reach the top spot. In this setting, any initial point far from the global peak generates a pattern leading to final fitness values orderly distributed according to the distance from the optimum. The reason is that on this maximum-complexity landscape movements parallel to the axes lead to a point on the ridges with a probability increasing with the distance from the global optimum. The fitness values of the ridges also have decreasing fitness farther from the global peak, and this is why the final fitness of a search started far from the optimum is generally lower.

For intermediate values of complexity we observe “propeller”-like figures. The farther from the global peak the search starts, the lower the probability of reaching the global optimum, but the probability is not uniformly distributed according to the distance only. In fact, there are “special” areas that, in respect of other ares with the same distance from the global peak, lead with (almost) certainty to the maximum fitness point. The shape of these areas depends, obviously, on the shape of the fitness landscape, that is, the position of the landscape ridges in respect of the axes. Starting from the “lucky” areas a fitness climbing path ends at or near the global maximum. Conversely, the “unlucky” points are pushed through a pattern that hits earlier on an ridge, a local peak far from the optimum.

4.4 Length of search strategies

In order to better understand the shape of the local peaks distribution, as well as their basins of attraction, figure 4 shows the number of steps employed by the search strategies to reach the final peak (global or local maximum). For each initial point the graphs report the (average) number of steps employed by the search strategy before stopping. For $|a_{i,j}| = 0$ these values generate a sort of “funnel”: the further the starting point, the longer the path, as obviously can be expected by searches all ending at the global optimum.

The second graphs, for $|a_{i,j}| = 0.25$, provides a more detailed intuition of the paths implied by the landscape. In fact, we see that not all the points on the ridges constitute local peaks, but only those far from the global peak. For a clearer intuition of the reason for this result, consider that an ridge is defined by two relevant aspects. Firstly, its angle in respect of the axes, and, secondly, its slope in respect of the maximum fitness point. Consider a step Δ made from a point right on the ridge; the fitness of the new point, in respect of the fitness of the starting point on the ridge, tends to decrease because the angle causes the new point to be necessarily off the ridge. However, if the step gets closer to the

[7]Tests with different values of Δ showed the irrelevance of the value chosen for this parameter, but for the level of detail of the graphs. The value used for the simulations is 0.05, so that the portion of the graphs shown (from 98 to 102 on both dimensions) is turned in a square lattice made of 80 units on each side and composed of 6,400 points. Therefore, the 30,000 runs generate, on average, almost 5 random searches started from each point of the landscape. The graph report the average values across all the strategies for each starting point.

optimum, the fitness tends to increase.

The combined effect depends on the slope of the area around the starting point. The segments of the ridges far away from the global peak have a gentle slope, so that the fitness increments provided by getting closer to the global optimum are dwarfed by the fitness decrements forced by getting away from the ridges. Therefore, these segments tend to trap a search strategy more easily, representing local peaks. Conversely, the ridges close to the global peak have a steep slope, allowing a criss-crossing strategy to evade the ridge getting closer and closer to the global optimum on a fitness increasing path.

This property affects larger segments of the ridges the lower is the angle with the axes (because this decreases the costs of "jumping" down an ridge) and the closer is the point to the global peak (which makes the ridges steeper). Therefore, the basin of attraction of the global peak includes also the segments of the ridges closer to the global optimum, and the landscapes with larger $|a_{i,j}|$ have smaller basins of attractions to the global peak, as suggested by larger parts with shorter searches, ending more frequently on local peaks.

These results show that pNK offers the possibility to represent the equivalent of local peaks by means of hyperplanes (sub-spaces of dimension $N-1$), that, in case of $N=2$ means lines. Moreover, we can determine the dimension and position of the basins of attraction of the local peaks in a deterministic way, depending, besides the functional form chosen, on the coefficients used for the level of interdependency. Finally, the patterns followed by search strategies have a clear and intuitive interpretation.

These exercises show that pNK offers modellers the possibility to implement intuitive results even using two dimensions only, and therefore without requiring large structure to generate the basic properties of complex systems.

4.5 Greedy vs. Random strategies

As a further test for pNK we show the results in using the proposed fitness landscape when matched by alternative research strategies. The goal of this exercise is to show that pNK is not only able to generate intuitively sensible results, but, given it simplicity, it allows also to explore the mechanisms generating the results, allowing modellers to have a detailed account of the phenomena simulated.

We use again the simplest pNK (two dimensions only) to make the classical comparison of a random vs. a greedy research strategy. Random strategies are those that choose the dimension (and direction) of a change randomly. The greedy strategy is based on the assumption that, since there are only two dimensions, an agent may able to test all the four available steps. While in the random search strategy the simulated agent chooses randomly among the fitness increasing steps, the greedy strategy mandates to choose systematically the direction providing the largest fitness increment. Given the simplicity of the computations involved, we could perform a relatively intense series of tests, for landscape having $|a_{i,j}| = 0.0, 0.1, ..., 1.0$. For each of landscapes with the 11 different interdependency values we performed both the random search and the greedy one, testing, as before, 30,000 random points for each landscape, and starting from each a random and a greedy search. We then computed the average fitness produced by the two types of strategies on each landscape. Figure 5 reports the average fitness obtained at the end of the two research strategies in respect of the coefficients determining the level of interdependency.

As expected for low values of $|a_{i,j}|$'s both strategies manage to systematically reach the global maximum, as indicated by the first values equal to 1 for both strategies. Also expected, when $|a_{i,j}|$ approaches 1 the greedy strategy manages to obtain a higher average fitness. In these landscapes the best that can be obtained is a relatively high local peak, given that there are no chances to reach the global peak by means of fitness increasing movements parallel to the axes. Interestingly, the greedy strategy is surpassed by the random one for low-intermediate values of $|a_{i,j}|$, from about 0.2 to 0.4.

To explain this result we need to consider the pattern generated by the greedy strategy. Figure 6 reports the final fitness reached by the greedy research strategy in correspondence of the coordinates of the starting point. These graphs are the equivalent, for the greedy strategy, to those reported in figure 3 for the random strategy.

The comparison of the results between the greedy and the random strategy reveals the difference between the two. When $|a_{i,j}|$ is close to 1, the greedy strategy manages to develop a path at the "bottom" of the valleys formed by the diagonal ridges of the fitness function (see last graph in figure 2). This is practically impossible for the random search, since, wandering randomly, it will be easily captured by the basin of attraction of a local peak located on one of the ridges. However, the very first graph of figure 6, referring to the case with $|a_{i,j}| = 0.25$, shows that the greedy strategy misses conspicuously to gain the global peak from the extreme corners of the space, while the random search can. The reason is that in those regions, with those values of the parameters, the highest gains are obtained moving toward the closer ridge, that, once reached, is a trap that cannot be escaped. Instead, a random search will very likely (practically always), alternate steps toward the ridge to step in the direction of the global peak, since both provide positive fitness gains. In this way it will easily enter the basin of attraction of the global peak, avoiding the trap capturing the greedy search strategy.

This exercise is an example of the risks of early optimization. Choosing the highest gains when far from the optimum risks getting to a local peak that cannot be escaped. Instead, random movements produce a path that is more likely to bring into the basin of attraction of the global optimum, providing, eventually, a better final performance. This well-known result does not deserve more comments here, but for supporting the claim that *p*NK works on spaces with topologies closer to everyday intuition. Also, the example shows the flexibility and simplicity of use of *p*NK, stemming from its low computational requirements. We will further use this advantage in the following tests exploring properties of multidimensional *p*NK landscapes.

4.6 Testing *p*NK on Multiple Dimensions

Let's move now to consider the behaviour of *p*NK in respect of multiple dimensions. We test the model for $N = 24$ dimensions. This choice for N allows to remove a potential source of disturbance. The original Kauffman's NK model prescribes that each bit is linked to K different bits. However, the choice of the K bits, or dimensions, is made randomly, implying that some dimensions may be chosen frequently (influencing many dimensions) and other less frequently (with a lower impact). In other words, dimensions may differ radically on their effect on fitness, since the number of dimensions depending on a change of a single bit cannot be known in advance. To avoid this problem (and therefore increase the reliability of our tests), we adopt the convention that landscape's dimensions are divided in equally large groups. All the dimensions in a group influence each other, but have no link to dimensions outside their group (the same convention is adopted in Frenken *et al.*, 1999). Setting $N = 24$ we can therefore build landscapes with groups composed by $K = \{1, 2, 3, 4, 6, 8, 12, 24\}$ dimensions[8].

[8]Note that in our case the equivalent of K represent the number of dimensions in a group, and therefore corresponds to $K + 1$ in Kauffmann's sense, which refers to the links among dimensions.

We use this setting to, firstly, test whether pNK is able to replicate the basic results provided by NK, setting $|a_{i,j} = 1|$ when a link exists (i.e. i and j are in the same group), and $a_{i,j} = 0$ when there is no influence between the two dimensions[9]. Secondly, we will see how the results change introducing intermediate values for $a_{i,j}$'s.

4.6.1 Local Peak's Fitness and K

The basic property of NK is that a fitness climbing search strategy is more likely to end up in a low fitness local peak when K is large, that is, with on a landscape with strong interdependency among dimensions. To test whether pNK can replicate this property we generated 300 random points and applied independently the search algorithm until a local peak is found. We replicate this exercise for all landscapes with different K values, and register the average fitness provided by the 300 searches for each landscape. Figure 7 reports these values, confirming the capacity of pNK to replicate this basic property of NK. The reason behind this results are the same as for the NK model: higher interdependence makes more likely that a strategy based on single-dimension movements ends up in a local peak.

Having shown that pNK generates, on average, lower fitness the higher the K value of a landscape, we may explore the average levels of local peaks, their distribution and so on. However, these results have no particular relevance here, for two reasons. Firstly, the statistical properties of the set of local peaks depend on the functional forms used. If necessary, it may be possible to search specific functional forms of the pNK fitness function such to satisfy determined conditions. Secondly, the distributions' properties of NK depend themselves on the peculiar implementation of NK, which does not reflect any particular property of a natural or artificial real-world system. Therefore, we skip this particular analysis and move to consider properties specific to pNK.

4.6.2 Graded interdependency

pNK can implement different levels of complexity not only by varying the number of connections, but also by setting a different strength for the interdependency links. That is, we can change, besides K, also the values for the $|a_{i,j}|$'s. When such values are positive but lower than 1 we expect the average fitness reachable by a search to be higher, other things being equal (i.e. same K), because of the weaker interdependency. In fact, the closer the coefficients to zero, the weaker the effects of interdependency, increasing the probability to find a path to areas closer to the global maximum. Figure 8 reports the average fitness for the same K's and for a set of intermediate values of $|a_{i,j}|$, ranging from 0.0 to 1.0. The hypothesis is fully confirmed, showing how the degradation of fitness generated by higher K values is lower the smaller are the interdependency strength coefficients $|a_{i,j}|$.

The tests made so far showed that pNK can, both, replicate the standard results produced by NK models and extend sensibly these results considering the additional aspects of pNK: real-valued dimensions and variable interdependency strength. In addition, pNK is also able to provide a measure of distance between two points which is not limited to the number of dimensions differing between the two points (Hamming distance), as in NK, but is defined in the more intuitive Euclidean distance defined on real-valued variables.

As a last exercise we abandon the assumption of "one-bit" search strategy and test pNK in replicating results obtained from a variation of NK with multi-bit search strategies.

[9] As before we assume that $a_{i,j} = -a_{j,i}|$ expressing negative interactions.

4.7 Search Strategies and "K"

The most prominent feature of NK (and, we have seen, pNK) is that, as the interdependence grows, limited mutation strategies fail to reach the global peak, and that the "costs" of lower fitness increases with increasign interdependency. However, it has been shown that suitable mutation strategies can exploit particular structures of interdependence to avoid the trap of local peaks (Frenken *et al.*, 1999).

We implemented a series of pNK fitness landscapes where dimensions either interact fully ($|a_{i,j}| = 1$) or don't interact at all ($|a_{i,j}| = 0$). As in the last exercise, dimensions are assigned to different groups, having links with all dimensions within those in the same group and no link with dimensions in other groups. However, instead of representing agents with search routines based on the one-bit (i.e. one-dimension) mutation strategy, we allow for agents using different search strategies.

We generate a population of agents exploring these landscape each endowed with specific mutation strategy. Agents member of a given *class* C divide the whole research space (i.e. all the set of dimensions) in blocks containing C dimensions each. To perform an attempted mutation, these agents choose randomly one the blocks, and then mutate one or more dimensions contained in the block. This exercise in a NK setting shows that agents in classes containing the same number of dimensions as the interdependency structure (i.e. when $C = K$) manage to avoid local peak and systematically reach the global optimum. Classes managing a smaller number of dimensions ($C < K$) are instead doomed to be trapped in local peaks. Classes containing more dimensions than those actually interacting ($C > K$), though able to reach the global optimum, are far slower in their path to the global optimum. Moreover, it was also shown that, though eventually limited in their path to the global peak, smaller classes are faster to reach relatively high fitness areas of the search space.

We tested a similar settings in pNK. We consider a pNK model with $N = 24$ dimensions. We implemented 8 different interdependency structures for K ={1,2,3,4,6,8,12,24}, meaning that each dimension i has $K - 1$ dimensions j for which $|a_{i,j}| = 1$, while the rest of the coefficients is 0. On each of these landscapes we generated 7 populations of agents applying a research strategy of class $C = \{1, 2, 3, 4, 6, 8, 12\}$. We located then all the agents (we used 100 agents for each class) on a randomly chosen starting point of the landscape (the same point for all the agents), and observed the average fitness of the agents in each population for 30,000 time steps. At each time step each agent tries a mutation as described above, mutating one or more dimensions within a randomly chosen group.

The results are shown in the graphs in figure 9, where we report the average fitness computed over the agents adopting the same strategies across time. We may interpret each experiment as if agents were facing problems with different levels of modularization: overmodular agents ($C < K$) divide the search space in too small modules, in respect of the true modularization of the environment (K). Optimal agents adopt a modularization coherent with that of the environment. Finally, "over-integrated" agents assume in their research strategies interdependencies that do not actually exist ($C > K$). The results produced using pNK are essentially identical to those generated with a standard NK model.

Besides replicating the same results, pNK can easily admit more flexible research strategies, since it allows not only to deal with different directions, but also with the extensions of the steps in the local exploration. Moreover, pNK uses a fraction of the memory and computational capacity required for similar exercises in NK, greatly simplifying the implementation and the computational time required even for large landscapes and interdependency structure.

4.8 pNK and agent-based models

In our review of the limitation of NK we included also the difficulty in using NK as a model for complexity to plug into a wider model of, say, organizations, markets, etc. The reason we mentioned is that the NK properties are statistical, while each single run of a landscape exploration is ill adapt to represent an actual exploration process. In practice, the pattern actually generated by a simulated agent on a NK landscape is composed by very few fitness increasing steps in between a long series of failed attempts. This features prevent the use of NK in order to represent agents that, for example, are engaged in competition, since, in practice, NK imposes its own timing of events and limits the number of fitness "jumps".

Such limitations do not affect pNK. The proposed model represents a fitness increasing pattern on the landscape as composed both by twists and turns (looking for the right direction, as in NK), but also as actual steps, required to move across a Euclidean space. To highlight the difference between NK and pNK in this respect figure 10 shows a comparison between 1,000 searches (one-bit mutations) on a NK landscape ($K = 3$) starting from the same location. On both cases the graphs represent the scatter plot between the value of the local peak found by the search and the number of fitness increasing mutations leading to it. The left graph (NK) presents a random cloud of points, indicating that there is no relation between the number of steps and the fitness of the final destination. Moreover, the largest number of steps of a search is only 16. Conversely, the right graph shows that pNK reports a more credible positive correlation between the number of positive mutations and final fitness of a search path. In addition, the search strategies in pNK manage to make between 160 and 300 steps, a far more sensible representation of gradual improvements.

This feature of pNK, though not concerning the inner properties of the system, is highly relevant for applications of the system in agent-based models. Using a model of complexity to represent the pattern of improvement of an agent, the modeller does not want to have a system allowing agents to make a few steps scattered in a huge time span. Rather, it is much more sensible to have agents exploring their space by means of patterns made of frequent and small steps. Only in this way it is possible to synchronize the modelling of the search activities with other agents' activities represented in the model.

For example, Ciarli *et al.* (2007) and Ciarli *et al.* (2008) use an elaboration of pNK to express the technological race of innovative firms. In these models firms undergo several economic activities (e.g. sell final and/or intermediate products, collect revenues, set price, etc.) as well research activities (searching for better technologies on a complex technological landscape). In these works the relative timing between the results of research (i.e. better technologies) and their economic impact (i.e. higher competitiveness) is crucial. A NK model, implying a slow and erratic pattern of "discoveries", would not have worked to represent the intended functions. pNK, besides providing the same intuition of complexity, gives also the flexibility to represent highly realistic search patterns. Moreover, the functional form of gives also the opportunity to extend the representation of the complexity space. In these works a technological innovation consists in improving the quality of a product on a shifting landscape, representing the exogenous movement of the technological frontier. As the red-queen, firms must invest in research even to just maintain the current quality level. This feature is easily represented in pNK by imposing a slow dynamics on the location corresponding to the maximum fitness. In general, pNK has be shown to be a very flexible tool to model a continuous stream of innovations, more adapt than the few volatile "jumps" produced by a standard NK model.

5 Conclusions and further research

NK is a very valid model to represent complexity, and has become quite popular to model complex systems different from its original biological metaphor, such as management of organisations and economic values of technological innovations. However, a number of limitations have hindered a wider diffusion of NK among scholars interested in a model representing complex systems, such as economists, organizational and social scientists. We sustain that these limitations are due to the origins of the system (biological metaphor), that suggested features, such as binary variables for system components and a stochastic fitness function, ill adapt to other domains. We propose a new model implementing the core elements of NK but removing, or at least relaxing, its major limitations. Namely, the proposed model, pNK, generates a complex landscape by means of the (controlled) interaction among the dimensions of the landscape. However, pNK considers real-valued dimensions (instead of the binary ones of NK), allows for grades of interaction (instead of presence/absence), and it is implemented by means of a deterministic function, greatly simplifying the implementation of the model and the interpretation of the results, significantly extending the potential applications, particularly in agent-based models.

The paper shows that pNK successfully replicates the core properties of NK: higher interactions generate more stringent constraints to a local search, forcing a simulated agent to get stuck in low fitness local peaks. Moreover, pNK offers many advantages in respect of the standard NK. Some of these are technical, for example, pNK is far simpler to implement and the landscape properties, stemming from a deterministic function instead of a huge set of random values, can be easily evaluated. Furthermore, and more importantly, the interaction structure depends not only on the number of interdependent dimensions, but also on the level, or strength, of these interdependencies. Finally, being composed by deterministic functions, pNK is, besides easier to implement, liable to parameterizations incorporating specific data, such as those derived from empirical evidence.

The paper presents a list of tests for pNK, showing how the core properties of complex fitness landscapes are maintained in a more familiar context of a real-valued space. The first tests explore the properties on a two dimensional space, providing a simple visual representation of the properties of pNK, showing how both the representation of the landscape and the search strategies conform more easily to the intuition in respect to the NK metaphor. Given its simplicity, it is possible to determine the properties of search strategies in complex landscapes by analysing the functional shape of the model. Thus, for example, it is possible to assess (and therefore to control) the basins of attraction of the different local peaks, or to explain the properties of different search strategies. The following tests show the equivalence of pNK to NK in representing a multidimensional landscape. Finally, it is shown that pNK offers far better possibilities of applications of complex landscapes in agent-based models, where the stochastic nature of NK make it unsuitable for applications requiring continuous improvements.

References

ALTENBERG, L. (1995), "Genome Growth and the Evolution of the Genotype-Phenotype Map", in W. BANZHAF and F. H. EECKMAN, eds., "Evolution and Biocomputation: Computational Models of Evo-lution", volume 899, pp. 205–259, Springer-Verlag Lecture Notes in Computer Sciences.

CIARLI, T., LEONCINI, R., MONTRESOR, S. and VALENTE, M. (2007), "Innovation and competition in complex environments", *Innovation: Management, Policy and Practice*, **3-4**.

CIARLI, T., LEONCINI, R., MONTRESOR, S. and VALENTE, M. (2008), "Technological change and the vertical organization of industries", *Journal of Evolutionary Economics*, **18**(3), pp. 367–387.

DURRET, R. and LIMIC, V. (2003), "Rigorous Results for the NK Model", *Annals of Probability*, **31**(4), pp. 1713–1753.

FRENKEN, K., MARENGO, L. and VALENTE, M. (1999), "Interdependencies and Adaptation", in T. BRENNER, ed., "Computational Techniques to Model Learning in Economics", pp. 145–165, Kluwer Academics.

KAUFFMAN, S. A. (1993), *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press.

KAUFFMAN, S. A., LOBO, J. and MACREADY, W. G. (2000), "Optimal search on a technology landscape", *Journal of Economic Behavior and Organization*, **43**, pp. 141–166.

KAUL, H. and JACOBSON, S. H. (2006), "Global optima results for the Kauffman NK model", *Math. Program.*, **106**(2), pp. 319–338, ISSN 0025-5610.

LENOX, M. J., ROCKART, S. F. and LEWIN, A. Y. (2006), "Interdependency, Competition, and the Distribution of Firm and Industry Profits", *Management Science*, **52**(5), pp. 757–72.

LEVINTHAL, D. A. (1997), "Adaptation on Rugged Landscapes", *Management Science*, **49**, pp. 934–950.

LI, R., EMMERICH, M. T. M., EGGERMONT, J., BOVENKAMP, E. G. P., BÄCK, T., DIJKSTRA, J. and REIBER, J. H. C. (2006), "Mixed-Integer NK Landscapes", in "Parallel Problem Solving from Nature - PPSN IX", Lecture Notes in Computer Science, pp. 42–51, Springer Berlin / Heidelberg.

RIVKIN, J. W. and SIGGELKOW, N. (2002), "Organizational sticking points on NK Landscapes", *Complexity*, **7**, pp. 31–43.

SIMON, H. A. (1969), *The Sciences of Artificial*, Cambridge MA. and London: MIT Press.

SKELLETT, B., CAIRNS, B., GEARD, N., TONKES, B. and WILES, J. (2005), "Maximally rugged NK landscapes contain the highest peaks", in "GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation", pp. 579–584, ACM Press, New York, NY, USA, ISBN 1-59593-010-8.

WAGNER, G. P. and ALTENBERG, L. (1996), "Complex Adaptations and the Evolution of Evolvability", *Evolution*, **50**(3), pp. 967–976.

WEINBERGER, E. (1991), "Kauffman, S. A. 1993. The Origins of Order: Self-Organization and Local properties of Kauffman's N-k model: A tunably rugged energy landscape.", *Physical Review A*, **44**(10), pp. 6399–413.

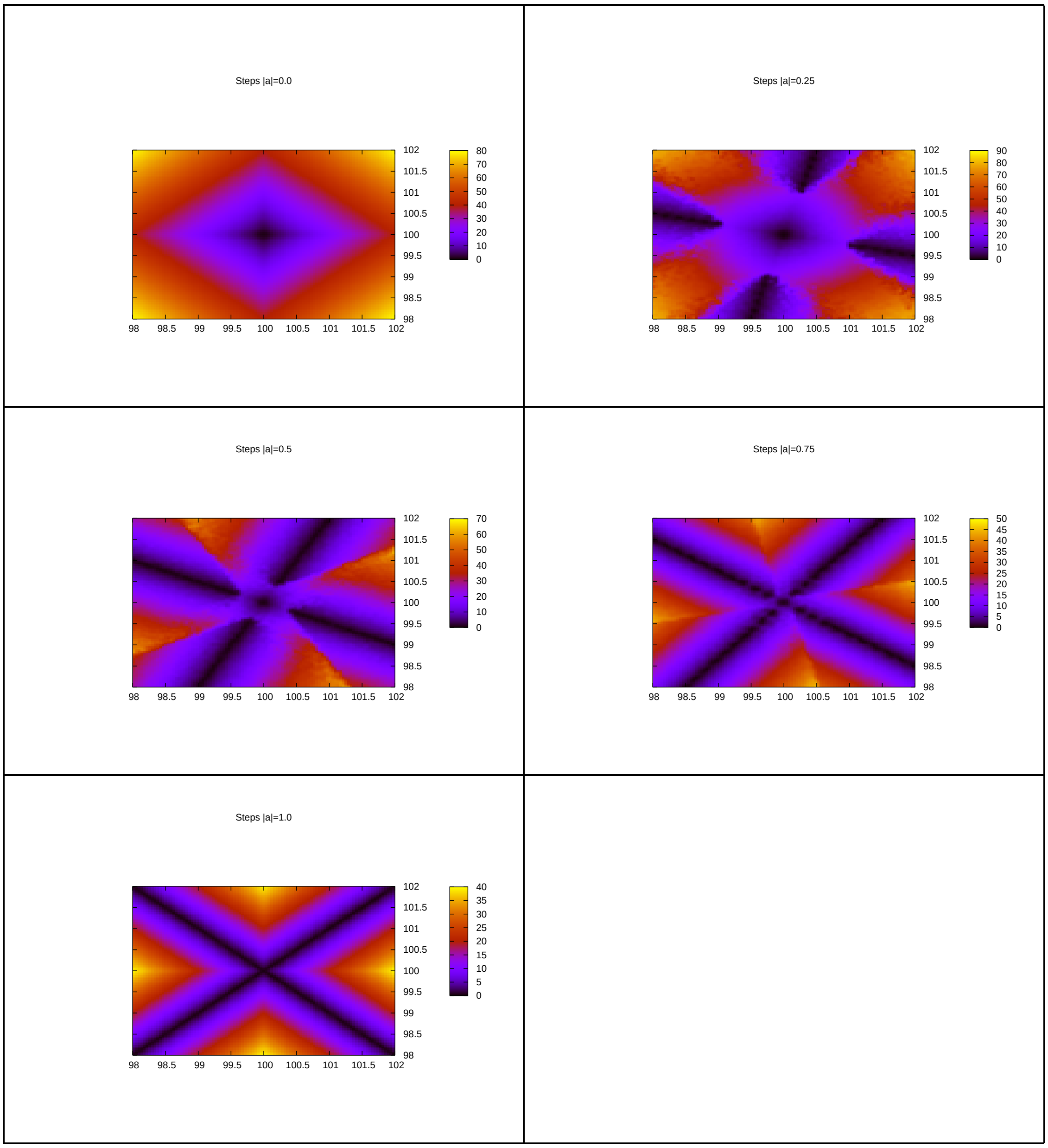

Figure 4: Number of steps before hitting a local or global peak in respect of the starting point. Average values from a sample of 30,000 searches (almost 5 searches per point on average) on fitness landscapes for $N = 2$ and different values of $|a_{i,j}|$.

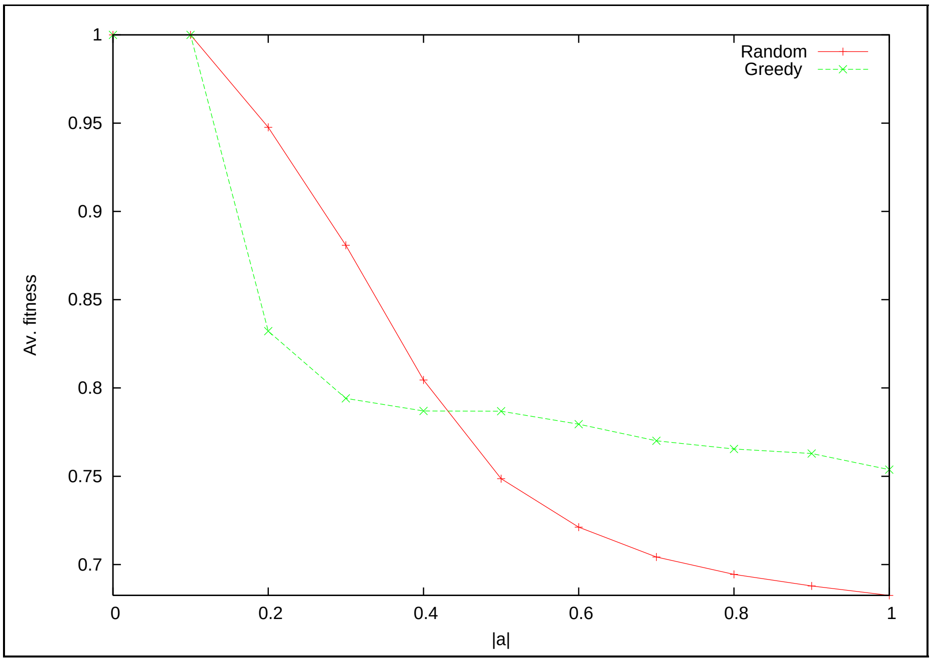

Figure 5: Average final fitness over 30,000 searches (random and greedy) on 11 landscapes with $a_{i,j}$'s ranging from 0.0 to 1.0.

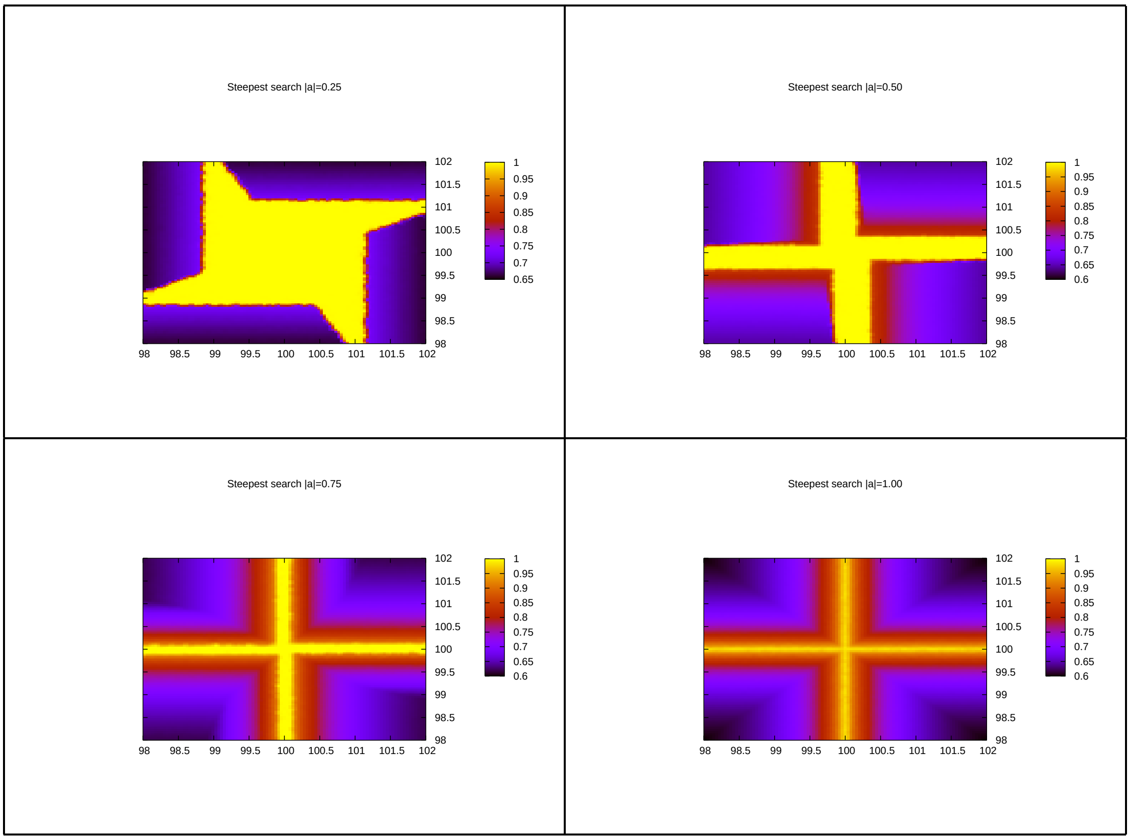

Figure 6: Final fitness produced by the greedy search strategy for different values of $|a_{i,j}|$.

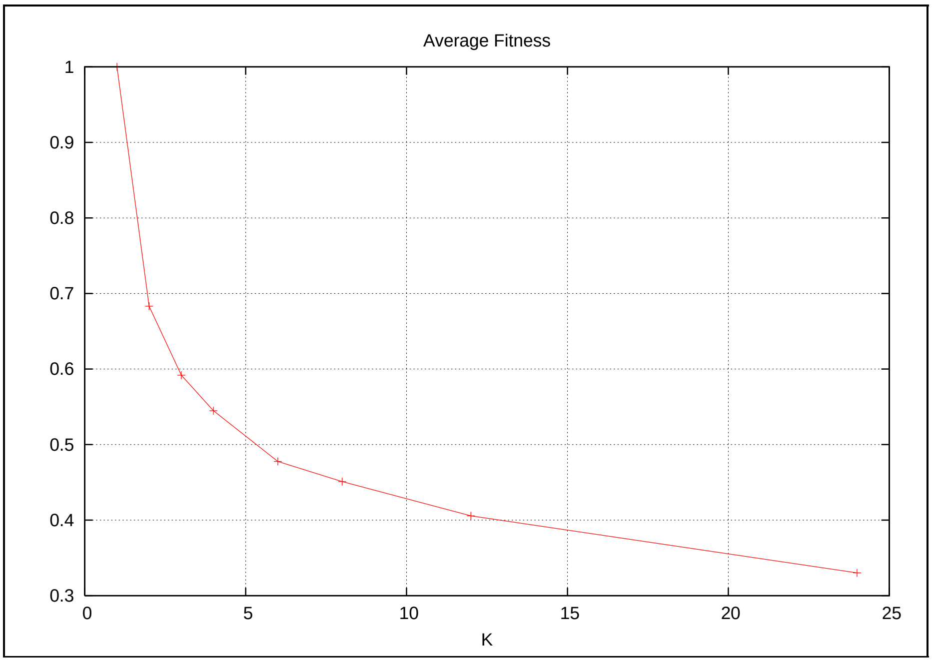

Figure 7: Average fitness produced at the end of 300 searches for each of the eight landscapes with $K = \{1, 2, 3, 4, 6, 8, 12, 24\}$. Interdependent dimensions have $|a_{i,j} = 1|$ while independent ones have $a_{i,j} = 0$

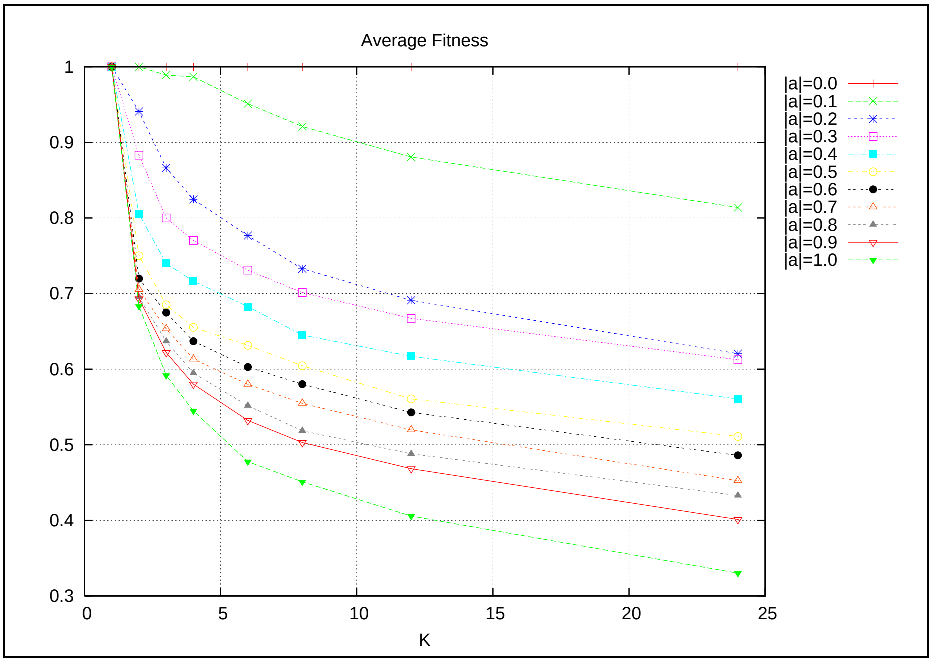

Figure 8: Average fitness produced at the end of 300 searches for each of the eight landscapes with $K = \{1, 2, 3, 4, 6, 8, 12, 24\}$ and for various value of $a_{i,j}$. Interdependent dimensions have $|a_{i,j}|$ from 0.0 to 1.0 while independent ones have $a_{i,j} = 0$

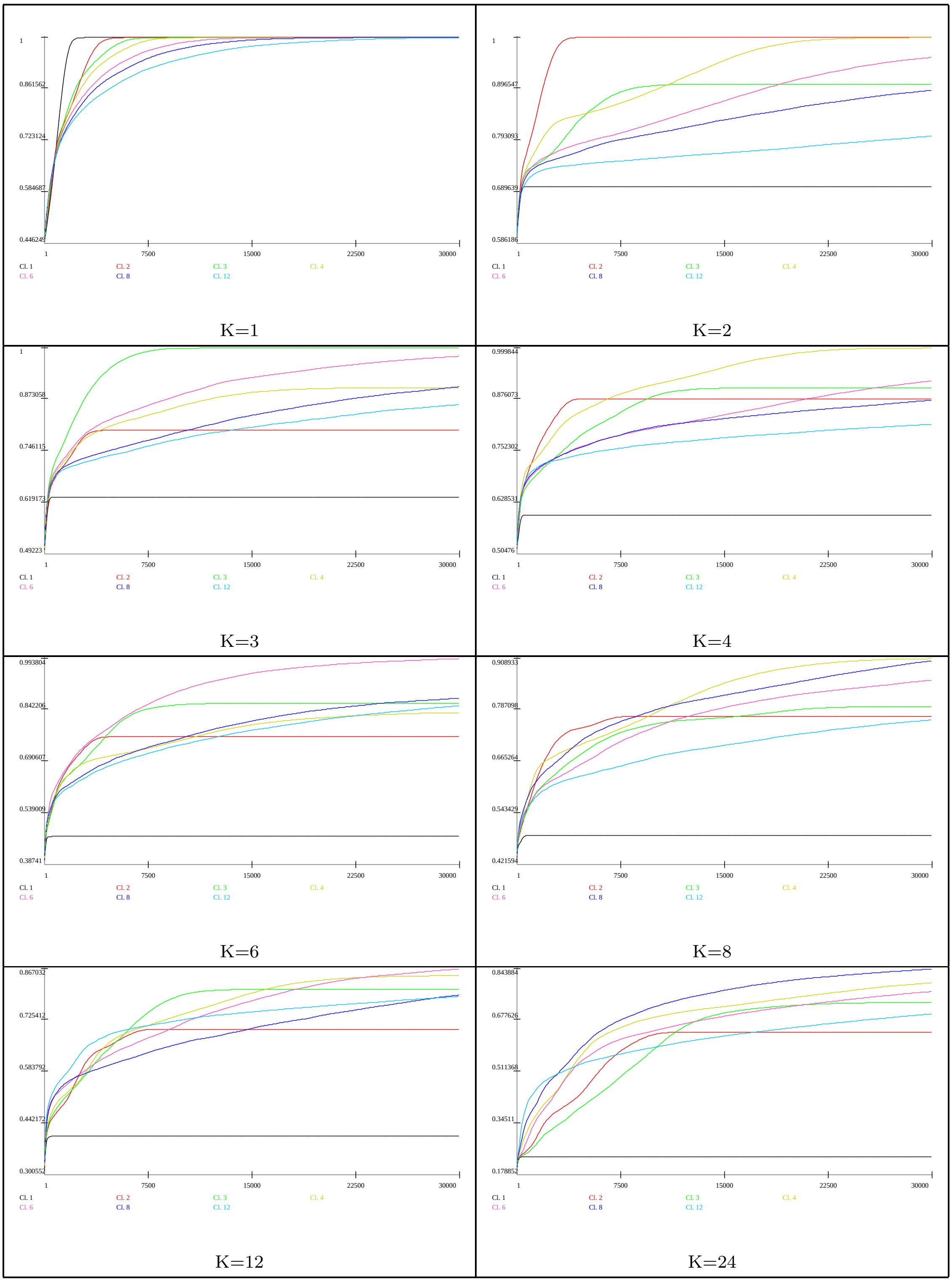

Figure 9: Average fitness across time for classes of agents mutating blocks made of C ={1,2,3,4,6,8,12} dimensions. Simulations reported for landscapes with K={1,2,3,4,6,8,12,24}.

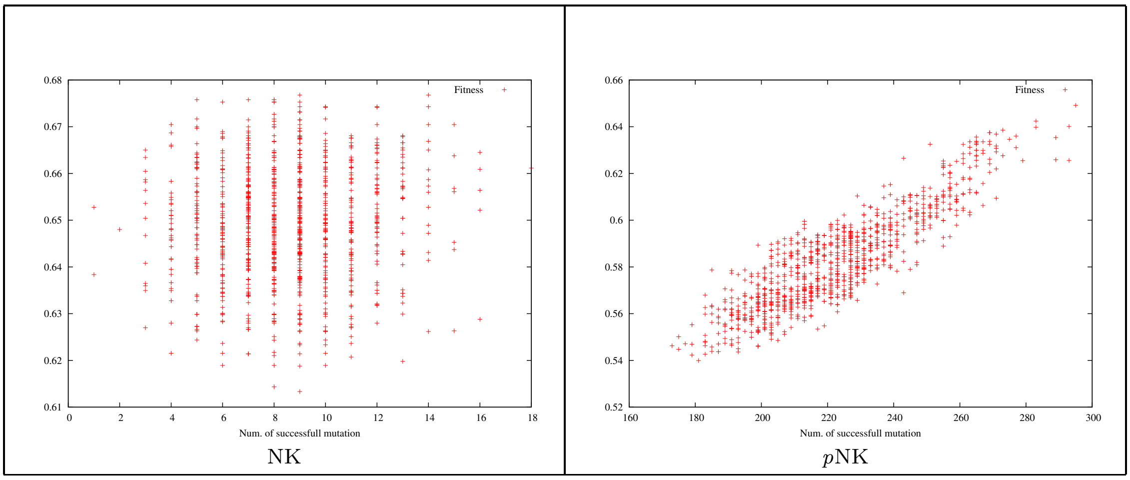

Figure 10: Values of fitness and number of successful mutations for NK and pNK. In both cases the graphs report the values for $N = 24$, $K = 3$ and agents adopting a one-bit mutation strategy. The data are obtained by 1,000 independent runs on the same landscapes, each starting from the same location. Values collected at the end of exploration, when a local peak is reached.