A generalised dropout mechanism for distributed systems

Larry Bull & Haixia Liu

Computer Science Research Centre

University of the West of England, Bristol BS16 1QY UK

Larry.Bull@uwe.ac.uk

Abstract

This letter uses a modified form of the NK model introduced to explore aspects of distributed control. In particular, a previous result suggesting the use of dynamically formed subgroups within the overall system can be more effective than global control is further explored. The conditions under which the beneficial distributed control emerges are more clearly identified and the reason for the benefit over traditional global control is suggested as a generally applicable dropout mechanism to improve learning in such systems.

Keywords: rugged fitness landscape, multi-agent systems, neural networks, NKD model, search.

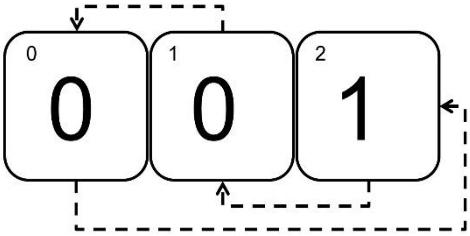
Introduction

Kauffman and Levin (1987]) introduced the NK model to allow the systematic study of various aspects of organisms evolving on rugged fitness landscapes and, given its abstract nature, the model has also been used widely within complex artificial systems. Distributed control is becoming increasingly significant in many areas, including edge computing, collective robotics, power grids, ad hoc networks, etc. Typically, the underlying functional dependencies which exist between the constituent parts of a system are only partially known or understood and hence the utility of different distributed control structures can be unclear, potentially leading to sub-optimal overall performance. A version of the NK model through which to systematically explore the general properties of distributed control systems has been introduced – the NKD model (Bull, 2020). Initial results showed that equivalent performance to global control exists within a significant proportion of the attribute space of the model and, more significantly, that *dynamically grouping elements within the control structures can prove beneficial*. This letter seeks to further explore and explain this finding, drawing an analogy with dropout in neural networks.

The NK Model

In the standard NK model (Kauffman & Weinberger, 1989), the features of the fitness landscape are specified by two parameters: N, the length of the genome; and K, the number of genes that has an effect on the fitness contribution of each (binary) gene. Thus, increasing K with respect to N increases the epistatic linkage, increasing the ruggedness of the fitness/problem landscape. The increase in epistasis increases the number of optima, increases the steepness of their sides, and decreases their correlation. The model assumes all intragenome interactions are so complex that it is only appropriate to assign random values to their effects on fitness. Therefore, for each of the possible K interactions a table of

$2^{(K+1)}$ fitnesses is created for each gene with all entries in the range 0.0 to 1.0, such that there is one fitness for each combination of traits (Figure 1). The fitness contribution of each gene is found from its table. These fitnesses are then summed and normalized by *N* to give the selective fitness of the total genome.



n	k	f_0	f_1	f_2
0	0	**0.76**	0.43	0.99
0	1	0.23	**0.74**	0.83
1	0	0.56	0.30	**0.02**
1	1	0.86	0.56	0.38

$$\text{Fitness} = 1/N \sum f_n$$

$$\begin{aligned}\text{Fitness(001)} &= (f_0 + f_1 + f_2) / 3 \\ &= (0.76 + 0.74 + 0.02)/3 \\ &= 0.51\end{aligned}$$

Figure 1: An example NK model where *N*=3 and *K*=1.

Kauffman (1993) used a mutation-based hill-climbing algorithm, where the single point in the fitness space is said to represent a converged species, to examine the properties and evolutionary dynamics of the NK model. That is, the population is of size one and a species evolves by making a random change to one randomly chosen gene per generation. The “population” is said to move to the genetic configuration of the mutated individual if its fitness is greater than the fitness of the current individual; the rate of supply of mutants is seen as slow compared to the actions of selection. Ties are broken at random.

The NKD Model

The NK model can be cast as an abstract distributed system containing *N* interacting agents/elements under global control: any effect upon fitness caused by a mutation/change considers all *N* genes/elements in the decision as to whether to accept that mutation/change. The NKD model was introduced as a generalisation to enable the exploration of distributed control structures within the space of NK models. Here each gene/element in the traditional NK model is extended to include *D* connections to other genes/elements (Figure 2). In the simple case these are assigned at random for each gene/element. When a mutation/change is made to a given gene/element, the decision as whether to accept that change is based upon the effect on fitness of the set of *D*+1 genes/elements. Hence the traditional case exists when *D*=(*N*-1). The networks of decision making typically overlap here. All results reported in this paper are the average of 10 runs (random start points) on each of 10 NK functions, ie, 100 runs, for 5000 generations. Here 0≤*K*≤15, for *N*=20 and *N*=100.

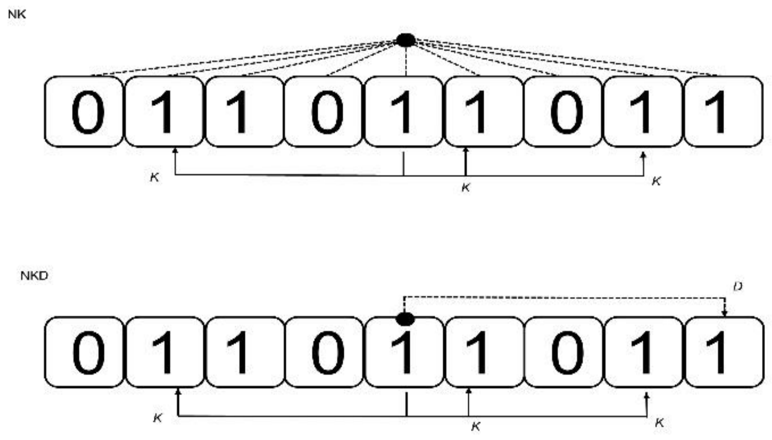


Figure 2: Showing the centralised, global control of the traditional NK model (*N*=9, *K*=3) and the fully distributed control structure of the NKD model (*N*=9, *K*=3, *D*=1). Connections shown for one gene/element only for clarity.

Figure 3 shows examples of how, with *N*=20, fitness is the same as *D*=(*N*-1) when *D*≥12 for 0<*K*<10 (T-test, $p<0.05$). For *K*>6 this is true for *D*≥8 and when *K*=15, ie, as *K*→*N*, the highest fitness seen is at *D*=8 (T-test, $p<0.05$). Figure 3 also shows similar results for *N*=100, although relative fitness is much worse for *D*<60 when 2<*K*<10. For higher *K*, fitness remains lower than at *D*=*N*-1 when *D*<80 (T-test, $p<0.05$).

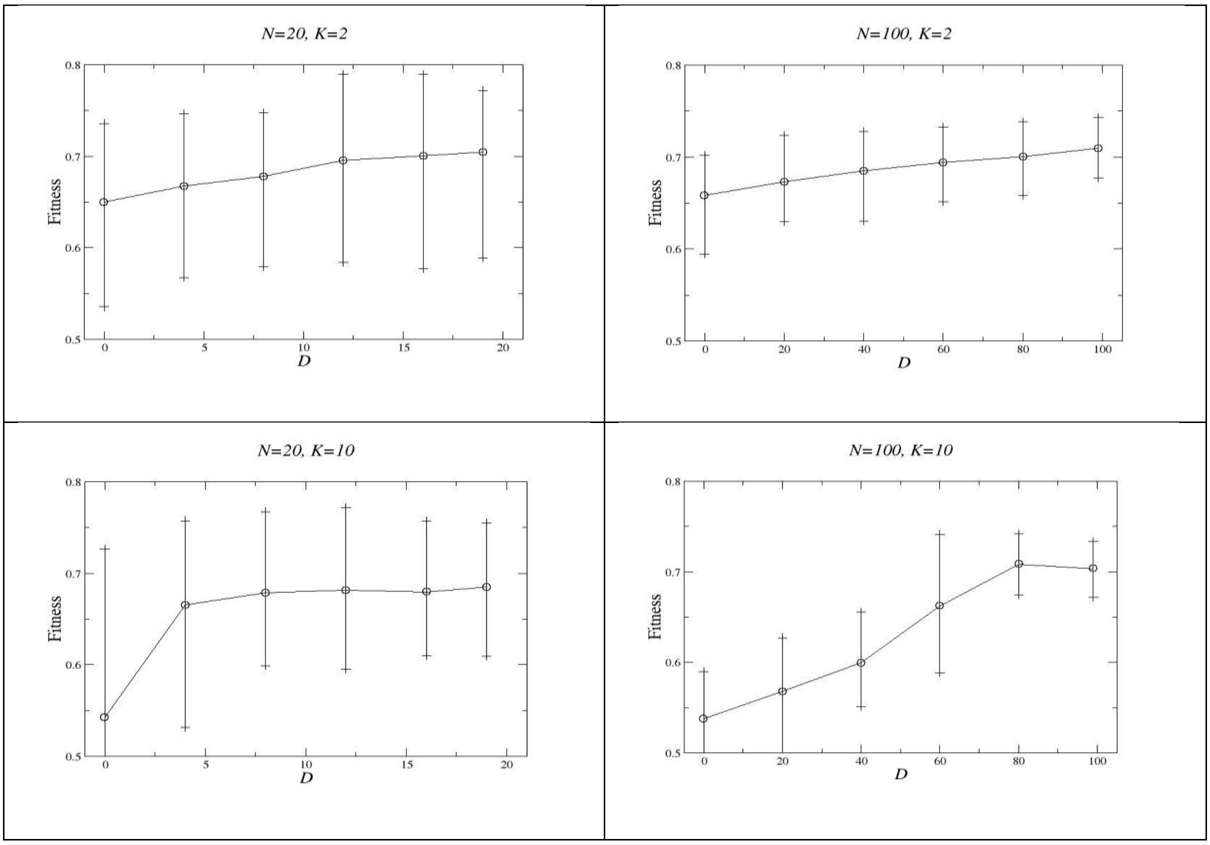


Figure 3: Showing the fitness reached after 5000 generations/iterations on landscapes of varying ruggedness (*K*) and degrees of control (*D*). Error bars show min/max values.

Hence whilst the best control structure varies with both *N* and *K*, it is never only at *D*=(*N*-1), ie, many distributed structures exist which give equivalent performance to the traditional centralised scheme.

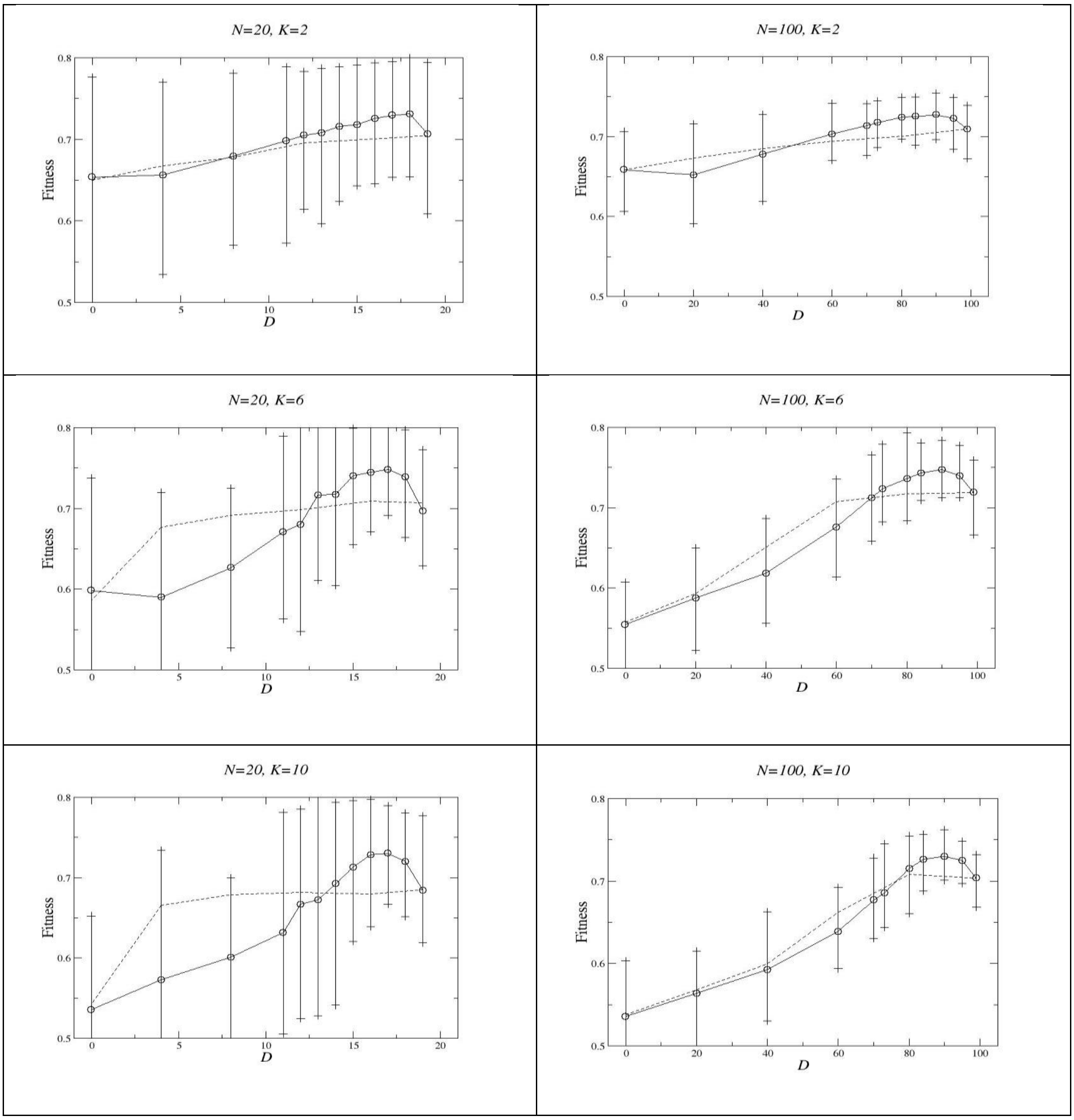
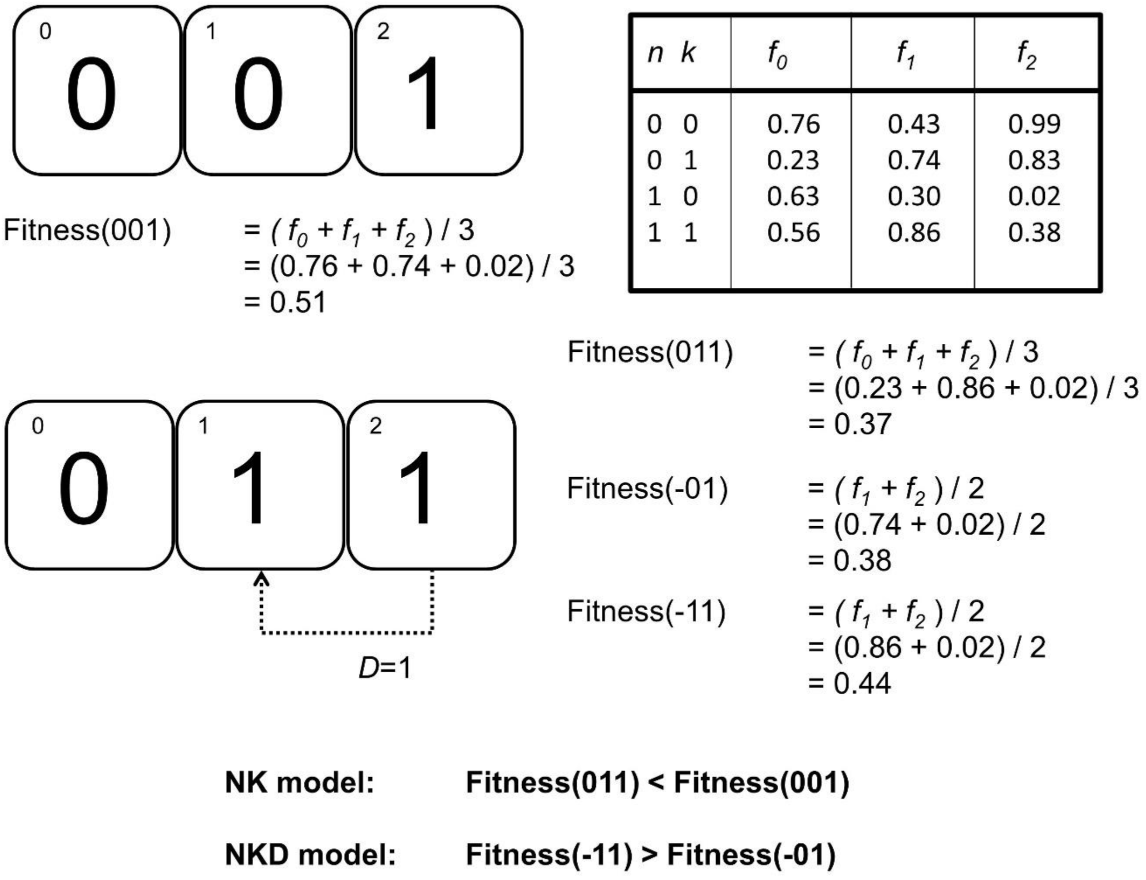


Figure 4: Showing the fitness reached after 5000 generations on landscapes of varying ruggedness (*K*), degree of control (*D*), and length (*N*) when the control structure is assigned at random per generation/iteration. Fitness of equivalent static grouping shown as a dashed line.

In some distributed system scenarios the topology of the control structure can vary temporally, eg, due to changes in geographic location in systems containing mobile elements. From a distributed control strategy view, this can result in the dynamic formation of the subgroup of elements involved in each system update decision. Figure 4 shows examples when the topology of the D connections of each gene/element is randomly (re)created on *each* generation/iteration. For small N, and $K>0$, there is a significant drop in fitness for low D compared to the use of statically allocated control subgroups. Regardless of K, optimal performance is typically seen when $D \approx 0.9N$, independent of N (T-test, $p<0.05$). Perhaps somewhat counterintuitively, such temporarily dynamic distributed control structures appear to be beneficial over the static case roughly when $0.8N < D < N$.

That is, better than global control performance can be achieved when less than 100% of the system is being considered in an update/change iteration/generation. How can such partial information be beneficial? Dropout was introduced to neural networks with the aim of reducing overfitting (Hinton et al., 2012) and has been widely adopted thereafter. In the basic scheme, a node(s) in the hidden layer(s) is probabilistically removed/ignored from the forward and backward pass of a training cycle. The motivation being to stop unhelpful co-adaptation between nodes in the identification of robust features. Other variants include the removal of a small or large number of individual connections (dilution). Thus, when training under dropout, the global network is not experienced by the learning mechanism: a subset of the weight/error space is experienced and updated per cycle, with an overall averaging process happening.

This appears to directly correlate with the dynamic grouping scheme in the distributed control system presented here: some elements of the global system are not involved in a given learning step. The global control case above becomes trapped in local optima on the more rugged fitness landscapes, with the increase in performance from dynamic subgroups showing a tendency to increase with increasing K (Figure 4). Doing the learning step – here accepting or refusing a mutation - in a subset of the global problem space clearly helps to avoid local optima in the global space. That is, *a less fit solution in the global problem space can be accepted by making the decision within a subset of that space* (Figure 5). With the caveat the subset is varied appropriately. This finding is also loosely related to the NKP model (Barnett, 1998) wherein neutral regions are added to traditional NK fitness landscapes thereby reducing the number of local optima.



n k	f_0	f_1	f_2
0 0	0.76	0.43	0.99
0 1	0.23	0.74	0.83
1 0	0.63	0.30	0.02
1 1	0.56	0.86	0.38

Figure 5: Example contrasting the selection processes within the NK and NKD models.

With dropout viewed as a mechanism by which to escape local optima, the frequency with which it is applied to aid the search process potentially becomes significant. Figure 6 shows examples of how the benefit is seen even when the frequency of change in the control structure is very low in comparison to Figure 4.

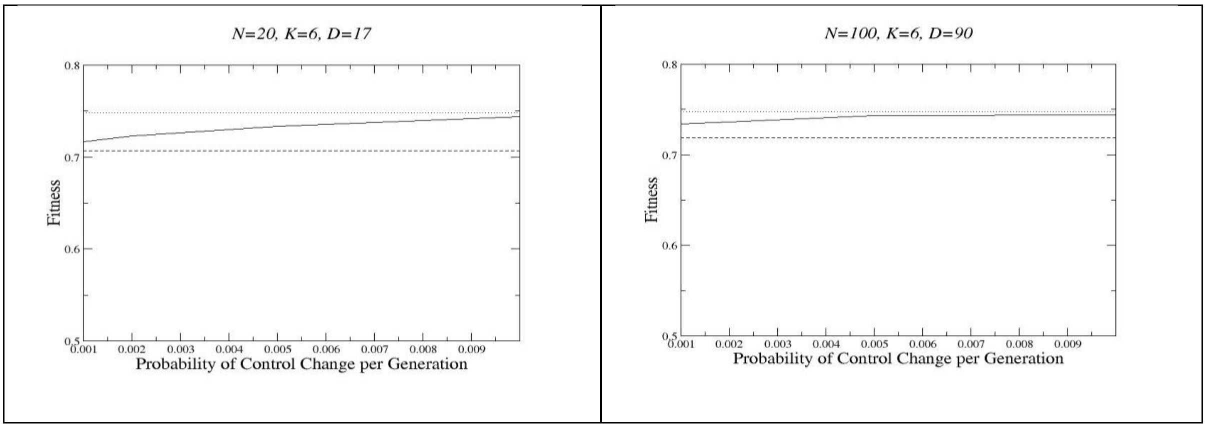


Figure 6: Showing the fitness reached after 5000 generations on landscapes of varying length (*N*) when the control structure is changed at random with different frequencies. Fitness of equivalent static grouping shown as a dashed line (prob change = 0) and with change per generation shown as dotted line (prob change = 1.0).

In the above, a single node/gene was chosen for changing/mutating and there is no reason to assume that is the optimal rate for the global search space, particularly for the more rugged landscapes. Indeed, the setting of such parameters is a major challenge in complex systems.

Figure 7 shows results from when two nodes are chosen at random to be updated per generation/iteration for the static grouping scenario. Here both changes must prove beneficial within each subset *D* to be accepted, with ties broken at random as before. As can be seen, performance is typically the same or significantly improved, particularly for higher *D*. Comparison with the dynamic grouping scheme results in Figure 4 finds the benefit of the mechanism is lost for all cases considered with *N*=20, and for *N*=100 except when *K*>6. That is, the dropout remains beneficial on the larger, more rugged landscapes. Performance with two mutations with dynamic grouping is not improved for high *D* (not shown). More mutations have not been explored here.

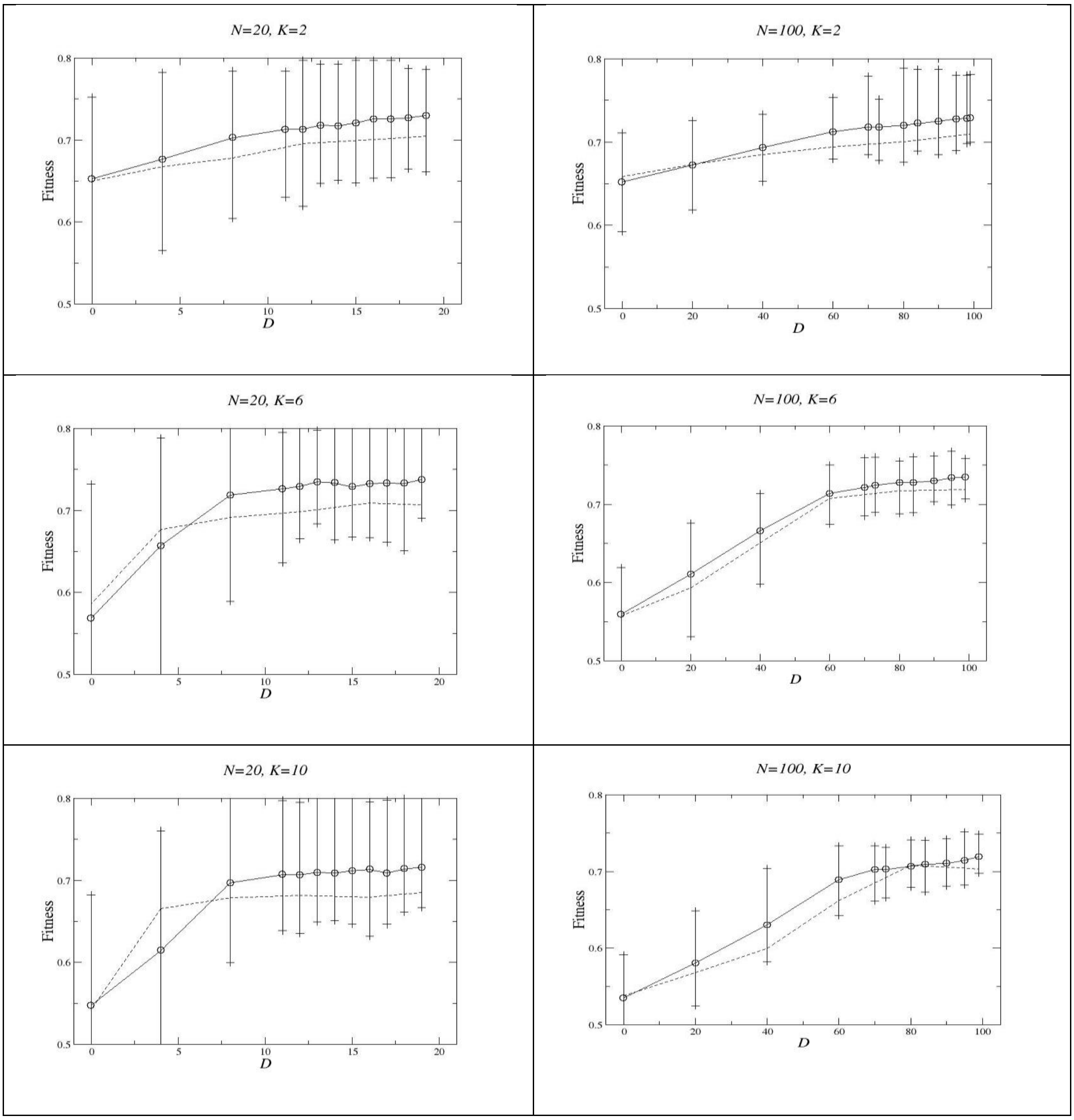


Figure 7: Showing the fitness reached after 5000 generations on landscapes of varying ruggedness (*K*), degree of control (*D*), and length (*N*) when the control structure is static but two nodes update per generation. Fitness of equivalent single node updating shown as a dashed line (Figure 3).

Conclusion

This letter has explored a recently presented result in which dynamic grouping improved performance across a range of distributed control conditions. It has been suggested that undertaking learning steps in a reduced set of dimensions of the problem improves the avoidance of local optima in the global space – so long as the set of reduced dimensions is varied to enable an averaging effect within the global space. The simple mechanism appears applicable to any system capable of functioning with components temporarily removed. Dropout as used in neural networks is a well-known example, although significantly it has not previously been described as working in this way.

References

Barnett, L. (1998) Ruggedness and neutrality - the NKp family of fitness landscapes. In C. Adami, R. K. Belew, H. Kitano and C. Taylor (Eds.), *ALIFE VI: Proceedings of the Sixth International Conference on Artificial Life* (pp. 18-27). The MIT Press.

Bull, L. (2020). Exploring distributed control with the NK model. *International Journal of Parallel, Emergent and Distributed Systems* 35, 413-422

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R.R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580

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

Kauffman, S. A., & Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128, 11–45

Kauffman, S. A., & Weinberger, E. (1989). The NK Model of rugged fitness landscapes and its application to the maturation of the immune response. *Journal of Theoretical Biology*, 141, 211–245