

# Exploiting the relationship between structural modularity and sparsity for faster network evolution

Anton Bernatskiy  
Dept. of Computer Science  
University of Vermont  
anton.bernatskiy@uvm.edu

Josh C. Bongard  
Dept. of Computer Science  
University of Vermont  
josh.bongard@uvm.edu

## ABSTRACT

A network is structurally modular if it can be divided into tightly connected groups which are weakly connected or disconnected from each other. Such networks are known to be capable of fast evolutionary adaptation, which makes modularity a highly desired property of networks involved in evolutionary computation. Modularity is known to correlate positively with another network parameter, sparsity. Based on this relationship, we hypothesize that starting evolution with a population of sparse networks should increase the modularity of the evolved networks. We find that this technique can enhance the performance of an existing technique for modularity evolution, multiobjective performance-connection cost (P&CC) optimization, and enable a multi-objective algorithm which maximizes performance and minimizes time since last mutation to produce modular solutions almost as efficiently as the P&CC optimization does.

## Categories and Subject Descriptors

G.1.6 [Optimization]: Global optimization; G.2.2 [Graph theory]: Graph algorithms

## General Terms

Theory

## Keywords

Genetic algorithms, Theory, Working principles of evolutionary computing

## 1. INTRODUCTION

Many problems in engineering can be reduced to global optimization of a function  $f(G)$  mapping a set of networks (possibly directed and/or weighted)  $\mathcal{G}$  onto some set of real numbers  $\mathcal{R} \subset \mathbb{R}$ . Suppose  $\mathcal{G}$  is a set of all possible networks with  $N$  nodes and up to  $\sim N^2$  edges. Further, each edge has one of  $C$  possible weights associated with it (absent connections are defined to have a weight of 0). There are on

the order of  $C^{N^2}$  such networks. In the worst case, solving the optimization problem involves computing  $f(G)$  for all  $G \in \mathcal{G}$ , i.e. an exponential number of operations.

Two approaches are utilized to cope with this complexity. First, instead of looking for the global optimum, a reasonably good solution which can be obtained in a reasonable time is sought. To this end, metaheuristic methods such as evolutionary computation are utilized. The second approach involves constraining or biasing the search towards some small subset of  $\mathcal{G}$ . The subset is selected heuristically in such a way that it is likely to contain or to be close to some reasonably good solution.

Indirect encodings (e.g. [11]) and additional optimization objectives (e.g. [5]) have been developed which bias the search towards small subsets of the search space. These methods perform well on a wide range of tasks.

A lot of attention has been attracted to the subset of networks possessing structural modularity, a widespread property of biological networks [7]. A network is modular if it can be divided into subgraphs with strong connections within them, but with little or no connections between them. It has been shown that many practical tasks have modular solutions which are reasonably good [5, 6, 3, 8, 4]. It was also found that modular solutions evolve more rapidly than their nonmodular counterparts in nonstationary environments [6] and that they generalize better [1, 12].

A variety of techniques for evolving modular networks has been suggested [6, 5, 8, 4]. One trait that many of these techniques share is an explicit [5] or implicit [6] bias of the evolutionary process towards sparse networks.

Clune *et al* [5] established that modularity evolves if connection cost is minimized while the network performance is maximized in a Pareto front-based multiobjective evolution. They also demonstrated that such an algorithm produces solutions with higher fitness compared to the case when the performance of a network is the only objective, and that this fitness was arrived at in fewer generations. They concluded that the influence of the connection cost pushed the population towards the region of the search space with sparser networks. Since sparse networks have less connections and, correspondingly, weights to be optimized, they adapt more rapidly than dense networks.

To explain the increase in modularity, they examined the search space and found a negative correlation between network density and modularity both in random and in highly fit networks.

Alternatively, one can think of this in terms of graph sets: the bias towards sparse networks causes search to optimize

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'15, July 11-15, 2015, Madrid, Spain.  
Copyright 2015 ACM TBA ...\$15.00.

fitness on the set of sparse networks first, and this set is much smaller than the set of all possible networks. Due to the relationship between modularity and sparsity mentioned above, the set of sparse networks happens to contain many modular networks, which makes this approach even more efficient.

In [5], evolution was initialized with a population of networks generated by assigning weights to all possible connections at random. Such networks are dense. From the optimization point of view, they are probing the unconstrained set of networks, which is inefficient. They are also not likely to be modular. Plots of density versus modularity versus generation provided in [5] show that it takes a number of generations for the population to reach a region of the search space with sparser and more modular networks.

The fact that the population was seeded with networks with  $\mathcal{O}(N^2)$  connections leads us to believe that this evolutionary transient becomes longer for larger networks. Here we show that seeding evolution with a population of sparse networks can remove this transient, which results in more rapid adaptation and increased modularity, especially for larger values of  $N$ . We also demonstrate that, given that the evolution starts with a population of sparse networks, it is possible to replace the connection cost objective with another objective of minimizing the time since the last mutation without significantly affecting the speed of adaptation. This, however, comes with the expense of destabilizing the process of modularity growth, resulting in a decrease of the final modularity metric  $Q$ .

## 2. METHODS

**Task:** We use evolution of attractors in boolean networks as an example problem for our study. The networks and their dynamics are identical to the ones described in [6].

Boolean networks are dynamical systems which are often used as simple models of gene regulatory networks (GRNs) found in biological cells. The state of each gene  $j$  is represented by a variable  $s_j$  which can be equal to either -1 or 1. The complete state of the network at time  $t$  is a vector of states of  $N$  individual genes,  $\mathbf{s}(t) = (s_1(t), s_2(t), \dots, s_N(t))$ . The state of the network at the next time step is determined as follows:

$$s_i(t + \tau) = \sigma \left[ \sum_{j=1}^N a_{ij} s_j(t) \right]. \quad (1)$$

Here,  $a_{ij} \in \{-1, 0, 1\}$  is the strength of connection from gene  $i$  to gene  $j$ . The function  $\sigma(x)$  is defined to be -1 if  $x < 0$  and +1 otherwise.

Such a network has a point attractor (hereafter referred to as just “attractor”) at state  $\mathbf{s}'$  if for some set  $S$  of two or more initial conditions the state of the network converges over time to  $\mathbf{s}'$  and then stops changing altogether.

The task for the evolutionary algorithms described in this work is to find a matrix of connection strength values  $A$  which describes a network with a desired attractor.

**Fitness:** The fitness of a network of  $N$  genes is defined against a target attractor  $\mathbf{s}'$ . For every target attractor  $\mathbf{s}'$  we tested whether the network dynamics converges to it if it starts at a close initial state. We began by generating a set  $S'$  of  $N$  perturbed attractor states, each of which differs from  $\mathbf{s}'$  at exactly one gene. Then we carried out the net-

work dynamics (1) starting from every state  $\mathbf{s}$  in  $S'$  for 20 iterations or until convergence to some attractor  $\mathbf{t}$ .

The fitness of the network was then computed as

$$f(\mathbf{s}') = 1 - e^{-3g}, \quad (2)$$

where

$$g = \frac{1}{N} \sum_{\mathbf{s} \in S'} (1 - D(\mathbf{s})/N)^5. \quad (3)$$

$D(\mathbf{s})$  is the Hamming distance between  $\mathbf{s}$  and  $\mathbf{t}$  if the convergence did happen and  $N$  otherwise.

Note that this fitness function differs from the one used in [6] only in the way in which the set  $S'_j$  is generated. The difference was introduced to reduce the time of computation.

**Optimization objectives:** We compare the performance of several multiobjective optimization techniques. For the sake of uniformity we reduce all objectives to a minimization of some function or property of a network. Three objectives are used throughout the paper:

1. *Performance (P)* objective, implemented as minimization of  $-f$ , where  $f$  is the fitness function.
2. *Connection cost (CC)* objective, implemented as minimization of the total number of connections.
3. *Time-since-mutation (TSM)* objective, which minimizes the number of generations since the last mutation of the network. The term is chosen to contrast with an established “age” objective [10], which minimizes the number of generations since the emergence of network’s family tree. The objective was chosen as to test whether diversity-promoting objective can facilitate the evolution of modular networks and because it does not require a metric in the space of possible behaviors (a requirement for using the novelty objective [9]) nor periodic injections of random genomes into the population ([10]).

All algorithms considered in this paper use exactly two objectives.

**Evolutionary algorithm:** We employ a simple biobjective evolutionary algorithm which relies on the concept of stochastic Pareto dominance. For a pair of networks  $(A, B)$  and a pair of minimizable functions  $(f, g)$ , we determine whether  $A$  stochastically dominates  $B$  by first generating a uniformly distributed random number  $r \in [0, 1)$  and comparing it to a user-defined constant  $p \in [0, 1]$ . If  $r > p$ , only the first objective  $f$  is taken into account and the dominance is established if  $f(A) < f(B)$ . If  $r \leq p$ , both objectives are taken into account and  $A$  dominates  $B$  if either of the following conditions holds:

1.  $f(A) < f(B)$  and  $g(A) \leq g(B)$ ,
2.  $g(A) < g(B)$  and  $f(A) \leq f(B)$ ,
3.  $f(A) = f(B)$  and  $g(A) = g(B)$  and  $ID(A) < ID(B)$ .

Here,  $ID(X)$  refers to the identification number of the network, the value of a global integer which is incremented every time any network in the population is created or mutated, starting from 0. The value is recorded at the moment of the  $X$  network’s mutation or creation. Thus,  $ID(A) < ID(B)$  indicates that the network  $A$  was either generated or mutated before network  $B$ .

The constant  $p$  describes the probability that the second objective is taken into account. If  $p = 1$ , stochastic Pareto dominance becomes deterministic, making the comparison between the networks simpler. However, it has been shown in [5] that evolution with the objectives of performance and connection cost (P&CC) has the best convergence rate when  $p$  is distant from both 0 and 1. Despite the differences in the stochastic Pareto dominance definition and in the selection strategy, we were able to confirm this result in our preliminary trials (data not shown). Hence, we chose not to switch to deterministic Pareto dominance in our comparison of the P&CC approach to other approaches.

The Pareto front is defined as a subset  $P'$  of a population  $P$  consisting of all elements of  $P$  which are not stochastically dominated by any network in  $P$ . At every generation increment, the algorithm finds the Pareto front  $P'$  in the current population and adds it to the new population. When done, the algorithm selects a network from the Pareto front at random, makes a copy, mutates it and appends the resulting offspring to the new population. This cycle is repeated until the sizes of the populations become equal, at which point the new population replaces the old one.

In all of our experiments the population was composed of 100 networks.

**Mutation operator:** The mutation operator acts on network nodes, having a fixed probability of 0.05 to change the set of strengths of incoming connections to any given node. One of the following operations may be performed on the node:

1. *Insertion* adds an incoming connection with a strength randomly selected from  $\{-1, 1\}$ . The gene at the tail of the new connection is selected at random among the genes which do not yet have a connection from them to the current gene.
2. *Deletion* deletes a randomly selected incoming connection of the current gene by setting its weight to 0.
3. *Density-preserving mutation*, which is a deletion event followed by an insertion event.

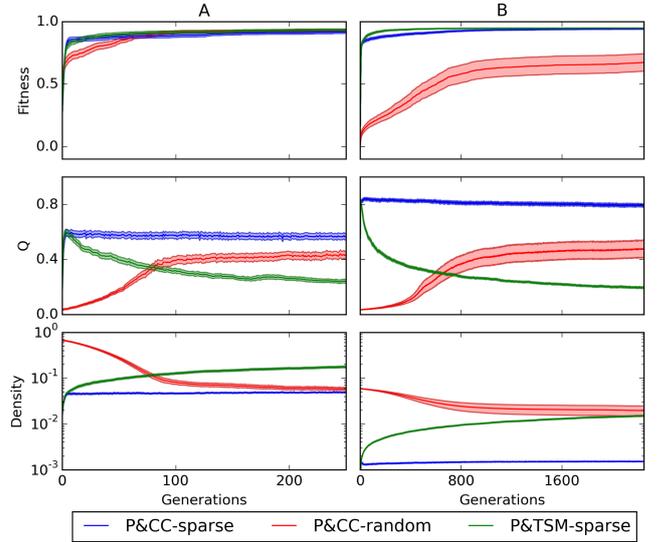
The probability of the density-preserving mutation  $p_{dpm} = 0.5$  in all our experiments. Probabilities of insertion  $p_{ins}$  and deletion  $p_{del}$  are controlled using their ratio  $r_{insdel} \equiv p_{ins}/p_{del}$ . In all of our experiments this ratio was set to 1.

If any operation is impossible, e.g. if there are no incoming connections at this node and deletion is invoked, the node's incoming connections remain unchanged. Thus, density-preserving mutation only really preserves density when it is applied to a node with one or more incoming connections.

**Initial populations:** We consider two types of initial populations of the networks. We will say that an initial population is composed of *random* networks if the networks are generated by choosing connection strength from  $\{-1, 0, 1\}$  at random for every possible connection in the network.

The alternative to this is to build the initial population out of randomly generated *sparse* networks. To obtain such networks, we create a network without any connections and mutate it once. In the resulting network, every node has at most one incoming connection and possibly multiple outgoing connections.

**Modularity metric:** We quantify the modularity of evolved networks using the  $Q$  metric (e.g. [5, 6]). For a given decomposition of a network into modules it measures the difference



**Figure 1: Comparison of parameters of the most fit networks evolved with different approaches. Columns A and B correspond to tasks A and B in the text. The lines represent mean values over 100 runs; bands are 95% confidence intervals for Student  $t$ -distribution.**

between the actual fraction of the edges within modules and the expected fraction of edges for a random network of the same density.  $Q$  is defined to be the maximum value of such a difference across all possible decompositions of a network into modules. To find the optimal decomposition, we use the Fast Unfolding of Communities method [2].

**Density:** The density of the network is defined as the number of connections in the network divided by the total number of possible connections,  $N^2$ .

### 3. RESULTS

We investigated the performance of our multiobjective evolutionary algorithm (see Methods) under the following three sets of objectives and conditions:

**P&CC-random** This setup is similar to [5]. Following [5], the probability that the connection cost objective is taken into account was set to  $p = 0.25$ . Evolution starts with an initial population of random networks.

**P&CC-sparse** Same as P&CC-random, but the initial population is composed of randomly generated sparse networks.

**P&TSM-sparse** The two objectives of performance and TSM are taken into account deterministically ( $p = 1$ ). Evolution starts with a population of sparse networks.

The performance of the approaches was measured using the task of finding a network with  $N$  nodes and  $N^2$  possible connections that settles into the attractor in which neighboring gene values are maximally different:

$$s' = (1 - 2(i \bmod 2)) \text{ for } i = 1, 2, \dots, N).$$

Two variants of this task were considered: variant **A** with  $N = 10$  and variant **B** with  $N = 30$ .

The comparison is presented in Fig. 1. In both tasks, the P&CC-random approach led to slower adaptation than both the P&CC-sparse and P&TSM-sparse approaches. For task A, the average fitness for P&CC-random across 100 runs was significantly lower than the fitness for both P&CC-sparse ( $p < 6 \cdot 10^{-5}$  with the Mann-Whitney U test implementation from `scipy.stats`) and P&TSM-sparse ( $p < 2 \cdot 10^{-6}$ ) at generation 25. Later in the evolutionary history P&CC-random reaches the same values of fitness as P&TSM-sparse does, marginally surpassing the P&CC-sparse approach ( $p < 3 \cdot 10^{-3}$  at generation 125).

For the more complex task B, the speedup caused by seeding the initial population with sparse networks is greater. Here, P&CC-random’s fitness was worse than the fitness of the two other approaches throughout the whole run of 2250 generations ( $p < 6 \cdot 10^{-14}$ ). The two approaches which start with populations of sparse networks – P&CC-sparse and P&TSM-sparse – show similar adaptation curves for both tasks, with P&TSM-sparse performing slightly better ( $p < 4 \cdot 10^{-5}$  for generations 50-800; the highest ratio of the mean fitnesses is 1.054).

The patterns of variation of the  $Q$  value are different for all approaches. For both tasks the connection cost-based techniques both evolve networks with high  $Q$  which is maintained among the champions of subsequent generations. However, it takes less generations for P&CC-sparse than it does for P&CC-random. The number of the required generations has increased approximately ninefold for task B, which is in agreement with our  $\mathcal{O}(N^2)$  hypothesis. The final  $Q$  value achieved by the P&CC-sparse approach is higher than that of P&CC-random for both tasks ( $p < 4 \cdot 10^{-6}$ ).

For both tasks the  $Q$  metric of P&TSM-sparse follows the same rising pattern as it does for P&CC-sparse during the first few generations. Both approaches develop highly modular solutions at this point, but for subsequent generations modularity of P&TSM-sparse solutions falls rapidly while the modularity of P&CC-sparse solutions remains the same. This ultimately causes P&TSM-sparse to produce the least modular solutions for both tasks ( $p < 9 \cdot 10^{-5}$ ).

The changes in density follow the changes in  $Q$  values. For task A, P&CC-random and P&CC-sparse stabilize at similarly low density, although it takes longer for P&CC-random to reach this state. For task B, the P&CC-random method evolves networks whose density stabilizes at a much higher value, perhaps due to the algorithm becoming trapped at local optima. For the P&TSM-sparse approach, the density keeps growing, but growth slows over generations.

## 4. DISCUSSION

Our findings confirm that seeding evolution with a population of randomly generated sparse networks can facilitate the evolution of modularity and increase the rate of adaptation. We found that this approach works especially well when optimizing large networks.

We demonstrated that this effect is present for a biobjective performance plus connection cost (P&CC) algorithm similar, but not identical, to the one described in [5]. However, when we seeded a multiobjective algorithm which minimized time since the last mutation instead of connection cost (P&TSM), we found that this initial increase in modularity decays over generations. Despite this, the new algorithm adapts approximately as fast as (and sometimes faster than) the P&CC-sparse algorithm for our task.

These results suggest that it is possible to replace the connection cost objective with another objective and still obtain, at equivalent evolutionary rate, networks of equivalent performance, possibly at the expense of some penalty to modularity. We speculate that replacement of connection cost with another diversity-promoting objective such as age [10] or novelty [9] may be beneficial for some harder tasks.

## 5. ACKNOWLEDGMENTS

We thank John Long, Ken Livingston, Nick Livingston, Mark Smith and Jodi Schwarz for fruitful discussions, and in particular, for asking the right questions. We thank Jim Bagrow for consultations in network theory. We thank the anonymous reviewers of GECCO’15 track “Multiobjective optimization” for useful criticism. This work was supported by the NSF award INSPiRE 1344227. The computational resources provided by the Vermont Advanced Computing Core which is supported by NASA (NNX-08AO96G) are gratefully acknowledged.

## 6. REFERENCES

- [1] W. R. Ashby. *Design for a brain*. Wiley, 1952.
- [2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
- [3] J. C. Bongard. Spontaneous evolution of structural modularity in robot neural network controllers. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 251–258. ACM, 2011.
- [4] B. Calcott. Chaining distinct tasks drives the evolution of modularity. In *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, pages 701–702, 2014.
- [5] J. Clune, J.-B. Mouret, and H. Lipson. The evolutionary origins of modularity. *Proceedings of the Royal Society B: Biological sciences*, 280(1755):20122863, 2013.
- [6] C. Espinosa-Soto and A. Wagner. Specialization can drive the evolution of modularity. *PLoS computational biology*, 6(3):e1000719, 2010.
- [7] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
- [8] N. Kashtan and U. Alon. Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(39):13773–13778, 2005.
- [9] J.-B. Mouret. Novelty-based multiobjectivization. In *New horizons in evolutionary robotics*, pages 139–154. Springer, 2011.
- [10] M. Schmidt and H. Lipson. Age-fitness pareto optimization. In *Genetic Programming Theory and Practice VIII*, pages 129–146. Springer, 2011.
- [11] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.
- [12] G. P. Wagner and L. Altenberg. Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, pages 967–976, 1996.