# Why Don't the Modules Dominate?

Investigating the Structure of a Well-Known Modularity-Inducing Problem Domain

Anonymous Author(s)

# ABSTRACT

2

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

Wagner's modularity inducing problem domain is a key contribution to the study of the evolution of modularity (for both evolutionary theory and evolutionary computation). We study its behavior under classical genetic algorithms.

Unlike what we seem to observe in nature, the emergence of modularity is highly conditional, dependent for example on the eagerness of search. In nature, modular solutions generally dominate populations, whereas in this domain, modularity – when it emerges – is a relatively rare variant. Emergence of modularity depends heavily on random fluctuations in the fitness function; with a randomly varied but unchanging fitness function, modularity evolved far more rarely.

Interestingly, high-fitness non-modular solutions could frequently be converted into even-higher-fitness modular solutions by manually removing all inter-module edges. Despite careful exploration, we do not yet have a full explanation of why the genetic algorithm was unable to find these better solutions.

# **CCS CONCEPTS**

• Applied computing → Biological networks; • Computing methodologies → Artificial intelligence; • Software and its engineering → Abstraction, modeling and modularity;

# KEYWORDS

Biological networks, Evolution of modularity

### ACM Reference Format:

Anonymous Author(s). 2018. Why Don't the Modules Dominate?: Investigating the Structure of a Well-Known Modularity-Inducing Problem Domain. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18)*. ACM, New York, NY, USA, 8 pages. https://doi.org/ 10.1145/nnnnnnnnnnn

### **1 INTRODUCTION**

Adaptability is an essential property of both biological and artificial evolutionary systems [30]. Biological organisms have already solved this problem through evolutionary adaptations, of which modularity is believed crucially important [10], providing hope for artificial evolutionary systems to also generate modular, adaptable systems [21]. In artificial systems, modularity is also desirable for comprehensibility/engineering reasons, a good example being

Unpublished working draft. Not for distribution.

2018-02-05 18:11. Page 1 of 1-8.

the "high cohesion, low coupling" principle in software engineering [6]. This emphasis on modularity was key to the software engineering boom over the last few decades [12].

Lack of modularity has been shown as a key limitation of artificial biological systems in scaling to higher complexity [15, 20]. Artificial neural networks are usually densely connected [13], whereas human brains have modular components taking different responsibilities, such as the hippocampus for novel situations and amygdala for emotional controls [7]. Thus it is important to understand the conditions leading to the spontaneous biological emergence of modularity. Engineers may leverage them to generate modular systems, able to solve more complex problems and to autonomously adapt to new environments. Conversely, such understanding can help to winnow the plethora of evolutionary theories in biology.

Formally, modularity is the divisibility of structures or functions into sub-units that perform autonomously [23]. Thus a module is a group of elements which associate preferentially within the group [8, 17]. Many biological activities and structures can be modeled in the form of networks – animal brains, signaling pathways, etc. [2]. A network is modular if it can be partitioned into highly connected components, and between these components there are only sparse connections [5, 9]. Elements within a module will tend to undertake coherent functions independent of outside elements [8, 16]. Such modules appear everywhere in biology [7], at multiple levels of biological organisation [7, 8]. Modularity can promote the evolvability of organisms, defined as the ability to rapidly adapt to novel environments [22]. Modular networks allow changes in one module without disturbing other modules; and modular structures can be reutilised and recombined to perform new functions [8, 26].

Despite decades-long research interest in modularity [28], there is no consensus on its biological origin [8, 27]. Two theories stand out, because their preconditions commonly arise in nature [28]: modularly-varying evolutionary goals [15] and specialisations in gene activity patterns [8]. In the former, modular changes in environments may generate an impetus toward modularity [15]. Organisms whose environmental sub-components change repeatedly show more modularity than those from stable environments [19]. Fluctuations are omnipresent in real environments [8, 29]. However it is unclear to what extent these fluctuations are modular [8].

Espinosa-Soto and Wagner<sup>1</sup> studied the emergence of modularity in an artificial environment requiring gene specialisation [8], in which gene regulatory networks need to be able to regulate toward multiple different patterns. This is a common occurrence in nature: the same collection of genes frequently exhibits different activity patterns at different phases of development or in different locations in an organism [14]. In ES&W' model, the network was

114

115

116

59

60

61 62 63

64 65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

for profit or commercial advantage and that copies bear this notice and the full citation

For all other users, construct the armon (author())

CECCO '19 Lul. 15 10 2010 Varia Tarres

<sup>© 2018</sup> Convright held by the owner/orthor

ACM ISBN 978-x-xxxx-x/YY/MM.

https://doi.org/10.1145/nnnnnnnnnnnn

<sup>&</sup>lt;sup>1</sup>Since we repeatedly need to refer to this crucial paper, we abbreviate it as ES&W

117 initially encouraged to regulate toward a single target. As evolu-118 tion proceeded, further targets were added. The targets were mod-119 ularly structured, sub-components changing regularly but modu-120 larly. They argued that too many interconnections between sub-121 components of a network following independently changing tar-122 gets would hamper the ability of each sub-component to follow 123 its own target. They were able to demonstrate the emergence of modularity under such conditions, using a recombination-free evo-124 125 lutionary algorithm. The results were persuasive, but also exhib-126 ited one key difference from nature: although modular structures 127 emerged, they did not dominate populations, whereas in biology, 128 once modular structures emerge, they become universal across a 129 species. Larson et al. [16] subsequently extended this work (within 130 a modified environment) to explore the effects of recombination on modularity emergence.

132 Our original intention was to further extend this stream, particu-133 larly emphasizing the effects of two ubiquitous biological phenom-134 ena, diploidy and crossover. To establish a baseline, we initially ex-135 perimented with variants of standard genetic algorithms, yielding 136 anomalous and difficult-to-explain results. Until these are fully ex-137 plored, our original goal had to be abandoned, as we could not trust 138 any results that might be obtained. The remainder of this paper 139 details our deeper exploration of the fitness landscape of this in-140 triguing environment. Some understanding has emerged, though we cannot yet say that we have fully understood the environment, 141 142 and some paradoxical results remain. We do not see this as a criti-143 cism of the original use of the model, as it was eminently successful in challenging the view that cyclic repetition of modularly struc-144 tured environments was essential for the emergence of modularity. 145 146 It is its extended use as a test-bed for understanding modularity 147 that is at issue here: whether further investigation of the compar-148 ative behavior of different algorithms on this problem gives any real insight into the equivalent behavior either in biology, or in 149 150 real-world application of evolutionary algorithms.

### 2 METHODS

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

We use genetic algorithms as our evolutionary simulation tools. The gene regulatory network that we used in this paper was originally proposed by Wagner [25] and customized by ES&W [8] and Larson et al. [16]. All simulation code was implemented in Java 1.8.0 and Python 2.7.10. They are all publicly available at (omitted for anonymity). Modularity was evaluated using the NetworkX package with the community API [11]. All the generated data can be downloaded at: (omitted for anonymity).

### 2.1 Model

Cells in an organism display heterogeneity in functionalities and morphologies, yet generally contain the same set of genes. Cells interpret the same genetic material in different ways so that their behaviors and structures vary. These distinct interpretations are due to regulation, among other mechanisms via the activation and repression of genes by other genes [25]: the effects of different genes are not mutually independent. A protein that is generated by one gene may activate or repress another. This mechanism can be usefully abstracted by a weighted directed graph. In this graph, 175

176

178

179

180

181

182

183

184

absence of an edge denotes lack of interaction, while a further abstraction limits the weights to +1 (activation) or -1 (repression) [25]. The term "gene activity pattern" describes the activeness status of the entire set of genes. Different activity patterns generally imply distinct cellular functions and forms [8].

We used ES&W's [8] representation of a gene regulatory network. The genotype of a gene regulatory network with N genes is represented as an  $N^2$  adjacency matrix  $A = a_{ji}$ . Each entry  $a_{ji}$ is restricted to be either 1, 0 or -1, represents activation, independence or repression of gene *i* by gene *j* to. The gene activity pattern of this network at time *t* is a Boolean row vector  $s_t = [s_t^0, ..., s_t^{N-1}]$ . Gene *i* can either be active  $(s_t^i = 1)$  or inactive  $(s_t^i = -1)$ . The state transition is modeled by:

$$s_{t+1} = \sigma[\sum_{j=1}^{N} a_{ji} s_t^j] \tag{1}$$

where  $\sigma(x)$  equals 1 if x > 0 and is -1 otherwise.

### 2.2 Fitness

A common role for GRNs is to maintain specific activation states in cells in the face of random external perturbations [1]. ES&W abstracted this by defining a sequence of two (or more) target states.

The ability of a GRN to robustly maintain a specific state was measured by randomly generating a set of *P* perturbations of the target, with each gene in the target having a 0.15 probability of being mutated to the opposite state (ES&W used *P* = 500, and Larson et al. *P* = 300). To each perturbation, the GRN was recursively applied. Preliminary experiments indicated that it normally took fewer than 20 transitions to reach an attractor [25]. If the GRN reached a stable attractor in fewer than 20 GRN steps from the perturbation, the Hamming Distance *D* between the attractor and the target state was returned; if 20 steps was insufficient, the maximum possible Hamming distance  $D_{max}$  was returned. In either case, the value  $\gamma_i = (1 - D/D_{max})^5$  was computed for each perturbation *i*, with  $1 \le i \le P$ . Finally, the mean value  $\bar{\gamma}$  over all  $\gamma_i$  was used to compute the fitness of the GRN *g* over a particular target *t*as:

$$f_t(g) = 1 - e^{-3\bar{\gamma}}$$
 (2)

This process, of randomly sampling a set of perturbations of the target, and evaluating the GRN's ability to robustly return them to the target, was repeated each generation.

In the first stage, the system was evolved to regulate the first target state alone. In subsequent stages, the fitness function rewarded regulation of newly introduced states, while maintaining pressure to regulate earlier states, by computing the overall fitness f(g) as the arithmetic mean of  $f_t(g)$  over all targets.

We followed this strategy, using only two targets: evolving for 500 generations with a single target, then introducing the second target for a further 1500 generations. We based our choice of the number of perturbations (75) on a trade-off between Totten's observation [24] that 75–100 perturbations are sufficient for emergence of modularity, and the practical need to minimize runtime.

Larson et al. applied a different approach to evaluating the fitness of networks [16]. Instead of sampling a new set of perturbations each generation, the instead sampled a static (but larger) set of perturbations at the beginning of each run, and utilised this

2018-02-05 18:11. Page 2 of 1–8.

231

Why Don't the Modules Dominate?

same set of corrupted targets whenever network fitness was calculated. This method has important computational cost advantages,<br/>since the fitness value of a given GRN on a given target remains<br/>fixed from generation to generation, so that cacheing and hash-When two matric<br/>i, matrices of their<br/> $C_1[0:i-1,:] = A$ 

ing methods can be used to give substantial speedups However it converts the original stochastically dynamic fitness evaluation into a static, deterministic one. ES&W's fitness landscape fluctuates stochastically each generation, whereas Larson et al.'s remains fixed. This has potential implications for search.

## 2.3 Evolutionary Simulations

p

ES&W imposed a mutation bias towards a specifc, relatively low, link density gene regulatory networks [8]. A node in the network has a probability  $\mu = 0.05$  to mutate every generation; if it mutates, it can either lose or gain an interaction. The probability for a node to lose an interaction is defined as

$$(u) = \frac{4r_u}{4r_u + N - r_u} \tag{3}$$

where *N* is the number of nodes in the network, and  $r_u$  is the number of regulators of gene u [8] – that is, the number of genes that exert effects on gene u. Complementarily, the probability for a gene u to gain an interaction is defined to be 1 - p(u). This bias acts to preserve the sparseness of the network, which computational biology research suggests is necessary for modularity to emerge.



## Figure 1: Modular and Non-modular networks. Different node colors represent distinct modules (based on the modular changes in target values). Green and red edges mean activation and repression.

ES&W defined modules in terms of the components of targets that followed similar activation pattern histories. In the most-used example, which used two targets of length ten, the activations of the first five locations in both targets were identical, while the activations of the second five were inverted between the targets. Thus the modules were assumed to be the connected components in the GRN involving nodes 1–5 and nodes 6–10. Figure 1 illustrates typical examples of modular and non-modular networks.<sup>2</sup>

ES&W did not use a crossover mechanism in their simulation [8]. In the reconstructed model of Larson et al., they limited crossover to nine possible partition locations of a 10-node network, corresponding to nine possible rows for splitting the adjacency matrix of a network horizontally [16]. We call this horizontal crossover.

When two matrices  $A_1$  and  $A_2$  are selected for crossover at index *i*, matrices of their children will be produced as

$C_1[0:i-1,:] = A_1[0:i-1,:]$	
$C_1[i:9,:] = A_2[i:9,:]$	
$C_2[0:i-1,:] = A_2[0:i-1,:]$	
$C_2[i:9,:] = A_1[i:9,:]$	

However this horizontal crossover may make the parental networks exchange not only modular clusters (defined by the nodes that follow a similar pattern in the activation targets [8, 16]), but also some interactions between the two modules. This may corrupt modularity. In contrast, we use a crossover mechanism that swaps interactions between modules in a gene regulatory network with connections between modules in another network. We refer to this as diagonal crossover. Compared with horizontal crossover, this approach, as Figure 2 illustrates, should better preserve the community structure, since this approach will lead to higher eventual modularity Q score (Wilcoxon signed-rank test; p < 0.0019).



Figure 2: Illustration of diagonal crossover

## 2.4 Modularity Metric

We adopted the Q scoring system to quantify modularity in a network, based on the algorithm proposed by Newman [18]. Briefly, this approach is defined as the difference between the ratio of the number of edges in the network connecting nodes within a module over the number of all the edges, and the same quantity when assigning the nodes into the same modules yet edges are assumed to be randomly connected in the network [15]. Formally, Q is calculated as

$$Q = \sum_{i}^{K} \left[ \frac{l_i}{L} - \left( \frac{d_i}{2L} \right)^2 \right]$$
(4)

 <sup>&</sup>lt;sup>2</sup>These color conventions are used throughout this paper. While color in the images conveys additional information, the key distinctions are still observable in black and white.

where *i* represents one of the *K* potential modules within a network, *L* is the total number of connections in a network,  $l_i$  stands for the number of interactions in the module *i*, and  $d_i$  is the sum of degrees of all the nodes in module *i* [8]. In other words, *Q* considers the two ratios of both intra-module connection density and inter-module connection density [18]. For a network to be considered high modularity, it must consist of as many within-module edges and as few inter-module edges as possible. Conversely, it will result in Q = 0 if all the nodes are partitioned into the same module.

The value Q will sit in the range of  $\left[-\frac{1}{2}, 1\right]$ . Nodes in the gene regulatory network are partitioned into different groups according to their regulating gene activity patterns.



# 3 PRELIMINARY EXPERIMENTS: MODULARITY SURPRISES

# Figure 3: Modularity decreased after target change (marked by the vertical line).

Our baseline-setting experiments, using a standard genetic algorithm with an elite of 10 and tournaments of size 3, revealed surprising differences in the emergence of modularity from the results of ES&W and Larson et al. Recall that in their experiments, overall modularity increased after a second [3]d target was added to the fitness function. In our initial experiments (using the Louvain metric [3] rather than the Q metric we use elsewhere), we instead observed a decrease immediately following the addition of the second target, with the overall modularity eventually stabilising to a level substantially below that of the first phase (see Figure 3).

Our settings differed from those of ES&W and Larson et al. in the following ways:

- (1) Use of crossover (difference from ES&W only)
- (2) Tournament vs proportional selection
- (3) Incorporation of elitism
- (4) Omission of the age-fitness Pareto mechanism (difference from Larson et al. only, [4])

Of these differences, item 1 could probably not explain our result since Larson et al. also used crossover, while items 3 and 4 both increase the relative eagerness of our search. Item 2 is more complex, since a tournament of size 3 exerts relatively weak selection pressure, but the relative pressure of tournament and proportional selection varies with the stage of the algorithm. Proportional selection depends on relative differences in fitness, so it typically exerts fairly strong pressure in early stages of search, but as the population fitness converges and differences reduce, pressure weakens; by contrast, tournament selection, being dependent only on fitness rank order, exerts a relatively constant selection pressure throughout. In particular, when populations are relatively converged (as at the time of the target switch), we would expect even relatively small tournaments to be more selective than proportional.



# Figure 4: Without elitism, modularity increased after target change (marked by the vertical line).

Based on these considerations, we decided to test the joint effects of elitism and tournament selection. The results bore out this hypothesis: the same algorithm and settings, with elitism eliminated and proportional substituted for tournament selection, led to the emergence of modularity, see Figure 4.

# **4 EXPERIMENT SETTINGS**

### Table 1: Gene Activity Patterns

Target Pattern	Generation where Pattern is added
1, -1, 1, -1, 1, -1, 1, -1, 1, -1	0
1, -1, 1, -1, 1, 1, -1, 1, -1, 1	500

Tables 1 and 2 show the gene activity patterns and the essential parameters of our evolutionary simulations. Unless otherwise specified, all experiments use the stochastic fitness evaluation of ES&W [8]. The only parameters that will vary from the tables are the selection type (tournament) and size, and the elite size. The detailed explanations of these parameters are given in Table 3.

2018-02-05 18:11. Page 4 of 1-8.

Edge Size	Number of	Per-location
	Perturbations	Perturbation Rate
20	75	0.15
Mutation Rate	Population Size	Selection Type
0.05	100	Proportional
Reproduction Rate	Maximum	Elite Size
	Generation	
0.9	2000	0 or 10
Trials per	Significance Test	
Treatment		
40	Wilcoxon	
	Signed Rank	
Table 2. Eval	anations of simula	tion noromotors
Table 3: Expla	<b>anations of simula</b> patterns that are p	<b>tion parameters</b> erturbed, and toward
<b>Table 3: Expl</b> Target Patterns	anations of simular patterns that are p which gene regulat	tion parameters erturbed, and toward ory networks evolve
Table 3: Expl Target Patterns Target Addition	anations of simular patterns that are p which gene regulat the generations wh	tion parameters erturbed, and toward ory networks evolve ere new targets are i
Table 3: Expl Target Patterns Target Addition Generations	anations of simular patterns that are p which gene regulate the generations wh troduce	tion parameters erturbed, and toward ory networks evolve ere new targets are i
Table 3: Explanation         Target         Patterns         Target Addition         Generations         Edge Size	anations of simular patterns that are p which gene regulate the generations wh troduce the initial number o	tion parameters erturbed, and toward ory networks evolve ere new targets are i f edges in the gene re
Table 3: Explanation         Target         Patterns         Target Addition         Generations         Edge Size	anations of simulat patterns that are p which gene regulate the generations wh troduce the initial number o ulatory network	tion parameters erturbed, and toward ory networks evolve ere new targets are i f edges in the gene re
Table 3: Expla         Target         Patterns         Target Addition         Generations         Edge Size         Perturbation	anations of simulat patterns that are p which gene regulate the generations wh troduce the initial number o ulatory network the number of pert	tion parameters erturbed, and towar ory networks evolve ere new targets are i f edges in the gene re urbed versions of eac
Table 3: Expla         Target         Patterns         Target Addition         Generations         Edge Size         Perturbation         Number	anations of simular patterns that are p which gene regulate the generations wh troduce the initial number o ulatory network the number of pert gene activity patter	tion parameters erturbed, and towar ory networks evolve ere new targets are i f edges in the gene re urbed versions of eac n
Table 3: Expla         Target         Patterns         Target Addition         Generations         Edge Size         Perturbation         Number         Perturbation Rate	anations of simular patterns that are p which gene regulate the generations wh troduce the initial number of ulatory network the number of pert gene activity patter the expected prop genes in a pattern	tion parameters erturbed, and towar ory networks evolve ere new targets are i f edges in the gene re urbed versions of eac n portion of corrupte
Table 3: Expla         Target         Patterns         Target Addition         Generations         Edge Size         Perturbation         Number         Perturbation Rate         Mutation Rate	anations of simulat patterns that are p which gene regulate the generations wh troduce the initial number of ulatory network the number of pert gene activity patter the expected prop genes in a pattern the probability of a lose an interaction	tion parameters erturbed, and towar ory networks evolve ere new targets are i f edges in the gene re urbed versions of eac n portion of corrupte

The evaluation metrics for experiments include both the eventual fitness values and modularity Q scores from the last generation. All are significance tested using Wilcoxon's Signed-Rank Test.

the type of selection used, and where tour-

the proportion of the population repro-

duced without change, vacancies being

the generation when the simulation will

nament, the size of the tournament

filled by the selection mechanism

#### **EXPERIMENTS AND RESULTS**

tion

terminate

#### 5.1 **Diagonal Crossover Promotes Modularity**

We ran trials comparing horizontal, diagonal and no crossover, in all cases without elitism. As Tables 4 and 5 show, diagonal crossover generated significantly higher fitness and Q score than horizontal crossover, which in turn generated significantly higher Q score, though non-significantly lower fitness, than absence of crossover.

2018-02-05 18:11. Page 5 of 1-8.

Selection Type

Reproduction

Maximum genera-

Rate

tion

### Table 4: Final Generation Best Fitness and Q Score with No, Horizontal and Diagonal Crossover

	No Crossov	er Horizontal	Diagonal
Fitness	0.9476	0.9446	0.9488
Q Score	0.1961	0.2919	0.3386
Table 5: Wilcoxon Ranked Sign Values for Table 4         Fitness P       Q Score P			
No < Horizo	ntal	(Horizontal <no) 0.0882</no) 	1.7090e-6
Horizontal <	Diagonal	0.0006	0.0019

This Boolean model was proposed by Wagner [25], who showed that random recombination made no difference to evolution of stability. Our experiments suggest that more structured forms of recombination (which occur in biology) can contribute to evolvability. Diagonal crossover can preserve underlying network modules. Although horizontal crossover did not preserve community structures as well as diagonal, its partitioning is still based on a modular structure, and thus partially preserves modularity.

### **Greedier Search Hampers Modularity** 5.2

5.2.1 Elitism Hampers Modularity.

Table 6: Final Generation Best Fitness and Q Score with and without Elites

2	Without Elites	With 10 Elites
Fitness	0.9488	0.9472
Q Score	0.3386	0.2735

### Table 7: Wilcoxon Ranked Sign Values for Table 6

	Fitness P	Q Score P
With 10 Elites < Without Elites	0.0003	0.0022

We trialled simulations with an elite of 10 against simulations with no elites. Significantly lower best fitness (regulatory capability) and modularity arose when an elite was used (Tables 6 and 7).

5.2.2 Proportional Selection generates Better Fitness than Tournament Selection, but Lower Modularity than Small Tournaments.

At least when populations are approaching convergence, tournament selection, especially with larger tournaments, imposes stronger selection pressure than proportional selection. In these results, we see in Tables 8 and 9 fairly much the anticipated decline in ultimate fitness with increasing selection pressure (though none of the differences are significant at the 1% level). Interestingly, proportional selection exhibits an unexpected (but non-significant) lower modularity than tournaments of size 2 or 3, but generates significantly more modularity than tournaments of size 10.

 Table 8: Final Generation Best Fitness and Q Score for

 Proportional Selection and Different Sized Tournaments

	Proport	Tourn	Tourn	Tourn	
		Size 2	Size 3	Size 10	
Fitness	0.9488	0.9404	0.9404	0.9371	
Q Score	0.3386	0.3697	0.3623	0.2783	

### Table 9: Wilcoxon Ranked Sign Values for Table 8

	Fitness P	Q Score P
Proportional > Tourn Size 2	0.7401	0.0467
Tourn Size 3 < Size 2	0.9313	0.7881
Tourn Size 10 < Size 3	0.0164	0.0015
Tourn Size 10 < Proportional	0.0227	0.0054

# 5.3 Dynamically Stochastic Fitness Evaluation generates Higher Fitness and Modularity than Static

### Table 10: Final Generation Best Fitness and Q Score for Dynamic and Static Stochastic Fitness Evaluation

	Dynamic	Static	
Fitness	0.9488	0.9379	
Q Score	0.3386	0.2948	$\mathbf{\lambda}$

### Table 11: Wilcoxon Ranked Sign Values for Table 10

	Fitness P	Q Score P
Dynamic < Static	3.5669e-8	0.0167

We compared dynamically stochastic fitness evaluation (in which the perturbations are generated anew each generation) and static fitness evaluation (the perturbations are generated once for all at the start of each run). Dynamic fitness evaluation outcompeted static on both survivability and modularity (Tables 10 and 11), though the latter result is not significant at the 1% level.

# 6 ANALYSIS AND DISCUSSION

# 6.1 Modular systems did not gain dominance via selection for robust target recovery

While some algorithm variants permitted modularity to emerge and survive under selection for robust target recovery, it did not (in extreme contrast to natural systems [23]) come to dominate populations. We saw some indication that greedier mechanisms, including elitism and tournament selection scheme, impeded the emergence of modularity. This implies that individuals who performed best, particularly in early stages of evolution, might not be especially modular. In general, the most competitive elites in each generation did not have the most modular gene regulatory networks. Overall, this suggests that while the ES&W framework has been useful to demonstrate particular properties of the emergence of modularity, in particular from pressure on gene specialization, it may not be sufficiently nature-like to function as a useful testbed to explore algorithmic effects on modularity emergence. If we see differences in algorithm behavior on this testbed, we will not know whether they arise from the abstractions from the natural environment (in which case they may usefully explain the behavior of natural environments, and suggest how to extend that behavior to artificial systems), or whether they arise from the substantial differences between the abstraction and reality (in which case any extrapolations would be moot).

Table 12: Modulari	ty domi	nance f	or data	from
sul	osection	5.1		

Generation Range	Modularity	Fitness
(500, 2000)	0.5000	0.9482
6	0.1736	0.9502

To further investigate the behavior of this system, we took more detailed measurements from the simulations in subsection 5.1, using the diagonal crossover. Specifically, from each run, we collected the fittest gene regulatory network among networks that were the most modular; and conversely, the network that was least modular among those that had the greatest fitness value. We expected the mean fitness of the latter to be lower than the former. Surprisingly, the situation was reversed: less modular networks generally recovered the target more robustly than more modular, as shown in Table 12. This does not reflect biological observations.

### Table 13: Modularity dominance for extended runs from more complex environments

Generation Range	Modularity	Fitness
(26000, 35000)	0.5506	0.9100
	0.4151	0.9419

We wondered whether this inconsistency could arise from insufficient complexity in the targeted gene activity patterns. The number of genes in patterns might be too simple, or the number of targets might be too few, to reflect natural environments. Perhaps modular network might give great performance on complex tasks, but worse than non-modular ones for simple tasks. Thus using the basic set-up of subsection 5.1, we ran more complicated evolutionary simulations using patterns comprising 15 nodes, encountering a sequence of seven different targets. Evolution was extended to 35,000 generations and during the final epoch from (26000  $\rightarrow$  35000) generations, it was evolving to robustly recover all seven targets. We repeated the preceding analysis; the results in Table 13 resemble those of Table 12. Overall, the number and complexity of targets could not resolve the issue: less modular networks still recovered the target more robustly than more modular.

2018-02-05 18:11. Page 6 of 1-8.

Why Don't the Modules Dominate?



Figure 5: Illustration of inter-module connection removal.

# 6.2 Inter-Module Connections Can Hamper Network Fitness

To further investigate this phenomenon, we took the 40 networks that were least modular among those having the greatest fitness value from the first experiment in subsection 6.1. We wondered what would happen if we simply removed all non-modular interconnections. So we did so with all 40 networks, and measured fitness after this removal (modularity was, of course, perfect). Of these 40 relatively low-modularity (Q) but near-optimal fitness networks, 24 exhibited even higher fitness after manually deleting inter-module edges. That is, more than half these originally nonmodular networks exhibited better fitness performance after removing all the inter-module connections. For example, the right network in Figure 5 was the consequence of removing inter-module connections of the network in the left. The fitness value of the latter was 0.9502 after removing 6% of the connections of the former, whose fitness was 0.9472. So these 24 much-improved networks (higher fitness than any found in the run, but also much higher modularity) were not only available to the evolutionary algorithm - they were even relatively nearby (a few edge deletion mutations away). Yet the algorithm reliably did not find them!

Originally, we suspected this anomaly might result from their lower edge density – perhaps they were too much below the edge density targeted by the biased mutation operator, so that this soft constraint eliminated them from the search. Further investigation revealed that on average, they still retained approximately 30 edges, above the targeted density of the mutation operator (see subsection 2.3), so far from being difficult to reach, the biased mutation operator favored moving toward them.

In order to further comprehend this phenomenon, that our evolutionary simulations could not find a path to the trimmed networks, we recorded all fitness values that could be obtained by removing one inter-module edge in turn, until all have been deleted, and plotted them as graphs. Figure 6 is typical. We could usually see a steadily improving fitness as edges were deleted, along a path that was favored by the biased mutation operator, yet our genetic algorithm could not find these paths.



Figure 6: Fitness Values along all Inter-module Edge Removal Paths from a High Fitness, Low Modularity Network that results in Increased Final Fitness.

# 6.3 Temporally fluctuating landscapes seem essential for generating modularity with a standard genetic algorithm

Because ES&W [8] re-sampled the perturbations used for fitness evaluation each generation, their fitness values varied dynamically and stochastically. By contrast, Larson et al. generated their samples once for each run, so that the fitness, while still stochastic, was not dynamic. While they saw useful emergence of modularity with their age-Pareto algorithm, which certainly generates higher population diversity and greater exploration, we did not have access to such an algorithm during the research on this paper. In any event our primary aim was to gain understanding in the context of the immense research knowledge that has been accumulated about standard genetic algorithms. Additionally, to increase the emphasis on modularity, we made the crossover always occur at the boundary of the two modules. That is, crossover would always interchange inter-module connections. We did not see modularity emerge with a standard genetic algorithm on a static landscape.

### Table 14: Fitness and Q Scores for Neighbors of Final Generation Fittest Individuals in Static and Dynamic Environments

	Dynamic	Static
Original Fitness	0.9461	0.9323
Best Neighbor Fit- ness	0.9410	0.9323
Original Q	0.3374	0.1851
Best Neighbor Q	0.3791	0.2223

To further understand this phenomenon, we collected the gene regulatory networks of the final generation, and mutated each network 499 times to generate neighbors. . We measured the fitness

GECCO '18, July 15-19, 2018, Kyoto, Japan

2018-02-05 18:11. Page 7 of 1-8.

813 values of these neighbors with the target perturbations from this 814 generation, and determined their maximum. In this fashion, we 815 would have 40 neighborhood-maximum fitness values for both dy-816 namic and static fitness evaluation. We repeated this process for 817 the modularity O score. As Table 14 shows, the fittest individu-818 als in the final generation for the dynamic problem were generally 819 local optima, whereas for the static problem, they were generally 820 on a fitness plateau, with equally fit neighbors, and a substantially 821 lower fitness than found in the dynamic problem. In both scenarios, 822 there were neighbors of substantially higher modularity than the 823 original individual, but overall modularity, both of the final generation best individual and of its neighbors, were much higher in the 824 825 dynamic problem than in the static.

#### 7 CONCLUSIONS

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

It is of considerable importance to identify those characteristics of evolutionary algorithms that will lead to modular problem solutions. Our belief that this is generally feasible is heavily reliant on the ubiquity of modular solutions in biological evolution. So useful testbeds for exploring algorithms' propensity to generate modular solutions need to abstract the relevant aspects of the real world. We have seen that from this perspective, ES&W's environment exhibits a number of anomalies. It is highly sensitive to the eagerness of search (suggesting that differences observed in the behavior of different algorithms might be ascribed to propensity to generate modularity, while in fact merely reflecting algorithm eagerness). While it supports the emergence of modularity, it does not appear to support their dominance, in contrast to real-world behavior. The non-dominance of modularity is surprising. We can manually find high-fitness paths that lead to modular solutions of even higher fitness, and these paths are favored by the mutation bias, yet the algorithm does not find them. This appears to be a result of the dynamic nature of the fitness function: the paths to these modular solutions may not appear favorable to the algorithm because of the dynamic stochastic variations in the fitness landscape. Paradoxically, this dynamic aspect seems essential to the emergence of modularity at all. The static variant of this problem does not support the emergence of modularity under standard genetic algorithms (we accept that it may do so under strongly diversity-encouraging algorithms such as age-Pareto algorithms). This appears to be linked to the genetic algorithms being trapped at local optima, which the varying fitness landscape of the dynamic variant allows them to escape.

These surprising properties of the problem render it problematic as a benchmark test. While we have gained some understanding of the problem domain, it is clear that deeper understanding is necessary, if we are to either rehabilitate it as a useful benchmark, or to find variants that may not behave so paradoxically. We are pursuing further investigations into the properties of the problem domain.

### REFERENCES

- [1] Alan Aderem. 2005. Systems biology: its practice and challenges. Cell 121, 4 (2005), 511-513.
- Albert-Laszlo Barabasi and Zoltan N Oltvai. 2004. Network biology: understand-[2] ing the cell's functional organization. Nature reviews genetics 5, 2 (2004), 101-113.
- [3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. Journal of statistical

	mechanics: theory and experiment 2008, 10 (2008), P10008.	871
[4]	Josh Bongard. 2017. Personal Communication. (2017).	872
[5]	gins of modularity. In Proc. R. Soc. B. Vol. 280. The Royal Society. 20122863.	873
[6]	Peter Coad, Edward Yourdon, and Peter Coad. 1991. Object-oriented design.	874
	Vol. 92. Yourdon press Englewood Cliffs, NJ.	875
[7]	L Andrew Coward. 2013. Towards a theoretical neuroscience. In <i>Towards a</i>	876
[8]	<i>Theoretical Neuroscience: from Cell Chemistry to Cognition.</i> Springer, 389–393. Carlos Espinosa-Soto and Andreas Wagner 2010. Specialization can drive the	877
[0]	evolution of modularity. <i>PLoS computational biology</i> 6, 3 (2010), e1000719.	077
[9]	Linton C Freeman. 1977. A set of measures of centrality based on betweenness.	8/8
[10]	Sociometry (1977), 35–41.	879
[10]	John Gernart and Marc Kirschner. 2007. The theory of facilitated variation. Proceedings of the National Academy of Sciences 104 suppl 1 (2007) 8582–8589	880
	https://doi.org/10.1073/pnas.0701035104	881
[11]	Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network struc-	882
	ture, dynamics, and function using NetworkX. Technical Report. Los Alamos Na-	883
[12]	Martin Hitz and Behzad Montazeri, 1995. Measuring coupling and cohesion in	884
	object-oriented systems. (1995).	885
[13]	Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. 1996. Artificial neural	886
[14]	networks: A tutorial. Computer 29, 3 (1996), 31–44. Peter A Jones and Shirley M Taylor 1980. Cellular differentiation cytidine	887
[14]	analogs and DNA methylation. Cell 20, 1 (1980), 85–93.	888
[15]	Nadav Kashtan and Uri Alon. 2005. Spontaneous evolution of modularity and	880
	network motifs. Proceedings of the National Academy of Sciences of the United	007
[16]	Silies of America 102, 39 (2003), 13/13-13/18. Ari Larson, Anton Bernatskiv Collin Cappelle Ken Livingston Nicholas Liv-	890
[10]	ingston, John Long, Jodi Schwarz, Marc Smith, and Josh Bongard. 2016. Recombi-	891
	nation Hotspots Promote the Evolvability of Modular Systems. In Proceedings of	892
	the 2016 on Genetic and Evolutionary Computation Conference Companion. ACM,	893
[17]	Mark El Newman, 2006. Modularity and community structure in networks. Pro-	894
[/]	ceedings of the national academy of sciences 103, 23 (2006), 8577–8582.	895
[18]	Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community	896
[10]	structure in networks. <i>Physical review E</i> 69, 2 (2004), 026113.	897
[19]	and modularity of bacterial metabolic networks. <i>BMC evolutionary biology</i> 7, 1	898
	(2007), 169.	899
[20]	Rolf Pfeifer and Josh Bongard. 2006. How the body shapes the way we think: a	900
[21]	new view of intelligence. MIT press. Rolf Pfeifer and Christian Schejer 2001. Understanding intelligence, MIT press	901
[22]	Massimo Pigliucci. 2008. Is evolvability evolvable? <i>Nature Reviews Genetics</i> 9, 1	002
	(2008), 75–82.	902
[23]	Gerhard Schlosser and Günter P Wagner. 2004. Modularity in development and	903
[24]	Mariko Totten, 2015. Exploring the Evolution of Modularity in Gene Regulatory	904
[]	Networks. (2015). https://scholarworks.uvm.edu/cgi/viewcontent.cgi?referer=	905
F	https://www.google.com.au/&httpsredir=1&article=1081&context=hcoltheses	906
[25]	Andreas Wagner. 1996. Does evolutionary plasticity evolve? Evolution 50, 3 (1996) 1008–1023	907
[26]	Günter P Wagner and Lee Altenberg. 1996. Perspective: complex adaptations	908
	and the evolution of evolvability. <i>Evolution</i> 50, 3 (1996), 967–976.	909
[27]	Günter P Wagner and JG Mezey. 2004. The role of genetic architecture con-	910
	evolution (2004) 338–358	911
[28]	Günter P Wagner, Mihaela Pavlicev, and James M Cheverud. 2007. The road to	912
[0:2]	modularity. Nature Reviews Genetics 8, 12 (2007), 921–931.	913
[29]	Shigeo Yachi and Michel Loreau. 1999. Biodiversity and ecosystem productiv-	914
	National Academy of Sciences 96, 4 (1999), 1463–1468.	017
[30]	Xin-She Yang, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet	915
	Karamanoglu. 2013. Swarm intelligence and bio-inspired computation: theory and	916
	applications. Newnes.	917
		918
		919
		920
		921
		922
		923
		924
		925
		926
		027
		72/
	2018-02-05 18:11. Page 8 of 1–8.	928