

# The Role of Degenerate Robustness in the Evolvability of Multi-agent Systems in Dynamic Environments

James M. Whitacre<sup>1</sup>, Philipp Rohlfshagen<sup>1</sup>, Axel Bender<sup>2</sup>, and Xin Yao<sup>1</sup>

<sup>1</sup> CERCIA, School of Computer Science, University of Birmingham  
Birmingham B15 2TT, United Kingdom

{j.m.whitacre,p.rohlfshagen,x.yao}@cs.bham.ac.uk

<sup>2</sup> Land Operations Division, Defence Science and Technology Organisation  
Edinburgh, Australia  
axel.bender@dsto.defence.gov.au

**Abstract.** It has been proposed that degeneracy plays a fundamental role in biological evolution by facilitating robustness and adaptation within heterogeneous and time-variant environments. Degeneracy occurs whenever structurally distinct agents display similar functions within some contexts but unique functions in others. In order to test the broader applicability of this hypothesis, especially to the field of evolutionary dynamic optimisation, we evolve multi-agent systems (MAS) in time-variant environments and investigate how degeneracy amongst agents influences the system's robustness and evolvability. We find that degeneracy freely emerges within our framework, leading to MAS architectures that are robust towards a set of similar environments and quickly adaptable to large environmental changes. Detailed supplementary experiments, aimed particularly at the scaling behaviour of these results, demonstrate a broad range of validity for our findings and suggest that important general distinctions may exist between evolution in degenerate and non-degenerate agent-based systems.

## 1 Introduction

The field of evolutionary dynamic optimisation (e.g., [3,8]) is concerned with the application of evolutionary algorithms (EA) to dynamic optimisation problems (DOP). In DOP, conditions vary frequently, and optimisation methods need to adapt their proposed solutions to time-dependent contexts (tracking of the optimum). EA are believed to be excellent candidates to tackle this particular class of problems, partially because of their correspondence with natural systems – the archetypal systems exposed to inherently dynamic environments.

Here we examine the properties that are believed to facilitate the positive relationship between *mutational robustness* and *evolvability* that takes place in natural evolution. In computational intelligence, these issues relate directly to concepts of fitness landscape neutrality and the search for high-quality solutions. Fitness landscapes are used extensively in the field of combinatorial optimisation to describe the structural properties of the problem to be optimised. The fitness landscape results directly from the choice of representation as well as the choice of search operators. Subsequently, different representations lead to different fitness landscapes and hence to problems of different difficulty (see [9] for an overview). Much research has focused on developing and analysing

different problem representations. Inspired by earlier developments in theoretical biology, neutrality – the concept of mutations that do not affect system fitness – has been integrated into problem representations using various approaches such as polyploidy (see [1,18,10,7,6]). However, there are theoretical reasons as well as some experimental evidence to suggest that only particular representations of neutrality will support the discovery of novel adaptations. Edelman and Gally have proposed that degeneracy, a common source of stability against genetic mutations and environmental changes, creates particular types of neutrality that increase access to distinct heritable phenotypes and support a system's propensity to adapt [5]. Before describing Edelman and Gally's hypothesis on the mechanics of evolution, we first define some biological concepts – evolvability, robustness, redundancy and degeneracy – with special emphasis on their meaning to optimisation.

Evolvability in biology is concerned with the inheritance of new and selectively beneficial phenotypes. It requires 1) phenotypic variety (PV), i.e. an ability to generate distinct heritable phenotypes, and 2) that some of this phenotypic novelty can be transformed into positive adaptations [16,17,14]. Similarly, evolvability in optimisation describes an algorithm's ability to sample solutions of increasing quality.

Robustness has several meanings in optimisation that mostly relate to the maintenance of adequate fitness values. In robust optimisation, robustness refers to the insensitivity of a solution's fitness to minor alterations of its decision variables. In dynamic optimisation, robustness is often defined as the insensitivity of a solution's fitness to perturbations in the objective function's parameters over time.

In biology, redundancy and degeneracy often contribute to the robustness of traits [5]. Redundancy means 'redundancy of parts' and refers to identical components (e.g. proteins, people, vehicles, mechanical tools) with identical functionality (see Fig. 1b). Redundant components can often substitute for one another and thus contribute towards a 'fail-safe' system. In contrast, degeneracy arises when similarities in the functions of components are only observed for certain conditions. In particular, while diverse components sometimes can be observed performing similar functions (many-to-one mapping), components are also functionally versatile (one-to-many mapping) with the actual function performed at a given time being dependent on the context. For degeneracy to arise, a component must have multiple context-induced functions of which some (but not all) are also observed in another component type.

In a landmark paper [5], Edelman and Gally present numerous examples where degeneracy contributes to the stability of biological traits. They hypothesize that degeneracy may also fundamentally underpin evolvability by supporting the generation of PV. In particular, degenerate components stabilize conditions where they are functionally compensatory, however they also retain unique structural characteristics that lead to a multiplicity of distinct functional responses outside of those conditions. These differential responses can occasionally have distinct phenotypic consequences [16] that may emerge as selectively relevant adaptations when presented with the right environment, cf [5,16,14,15]. Edelman and Gally's hypothesis describes degeneracy as a mechanistic facilitator of both robustness and adaptation that, in principle, could be applied outside biological contexts [16]. As described in [5,17], degeneracy is ubiquitous throughout natural systems that undergo parallel problem-solving. Yet until recently, it has not

informed the design and development of nature-inspired algorithms. Here we present evidence that degeneracy may provide a new (representational) approach to improve evolvability throughout EA execution in both static and dynamic environments. This approach could be applicable for many problems that are naturally modeled by systems with autonomous and functionally versatile agents that must survive within a heterogeneous environment.

## 2 The Role of Degeneracy in Evolution

When considering discrete local changes (mutations) in the decision variables of a single solution, the number of distinct accessible solutions is trivially constrained by the dimensionality of the solution space. Under these conditions, any increase in fitness neutrality – i.e. *mutational robustness* – will reduce PV. While more explorative/disruptive variation operators can increase PV, nature almost always takes a different approach. In gene regulatory networks and other biological systems, mutational robustness often creates a neutral network that improves access to PV over long periods of time, e.g. by drifting over neutral regions in a fitness landscape [4]. With PV being a prerequisite of evolutionary adaptability, a strong case has been made that this positive correlation of mutational robustness and PV is important to the evolvability of biological systems [4,16,14].

Inspired by these developments, some computational intelligence studies have investigated whether increasing neutrality (e.g. designing a many-to-one mapping between genotypes and phenotypes) influences the evolvability of a search process [1,18,10,7,6]. A common approach is to introduce genetic redundancy so that more than one copy of a gene performs the same function [1,18]. Although some researchers have indicated that redundant forms of neutrality improve evolvability, others have questioned the utility of fitness landscape neutrality generated through redundant encodings [7,15,16].

In the next section we describe, in detail, the computational study used to evaluate Edelman and Gally's hypothesis, including the details for the experimental setup. The proposed model provides the basis for simulating the evolution of a population of multi-agent systems (MAS) and depends on a minimal set of parameters that provide sufficient degrees of freedom to study the system properties – redundancy, degeneracy, robustness and evolvability – that we are interested in. The model (including the fitness function) is formally the same as the one developed in [15]. The study in [15] investigated degeneracy's relationship to genetic neutrality and evolvability and found that degenerate forms of genetic neutrality increase PV while neutrality from redundancy does not. In [16] we expanded on these results and found evidence that neither mutational robustness nor the size of the neutral network in a fitness landscape guarantees high PV, unless degenerate neutrality is present.

The studies in [15,16] investigated PV only within the local vicinity of a static neutral network. While this allowed for comparisons with recent biologically-inspired models (e.g. [4]), it was not within their scope to assign a selective relevance to heritable phenotypic variations. Thus, while previous studies were promising for Edelman and Gally's hypothesis, there has yet to be direct evidence that PV facilitated by degeneracy leads to higher rates of adaptive improvement. In the following we outline a set of experimental

conditions that allow us, for the first time, to evaluate Edelman and Gally's claim that degeneracy facilitates evolvability (and not just PV).

### 3 Computational Study and Experimental Setup

Each MAS  $\mathcal{M} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$  consists of  $n = 30$  agents and each agent is able to perform two types of tasks  $\mathbf{a}_i = (a_{i1}, a_{i2})$  where  $0 < a_{i1} < a_{i2} \leq m$ . We have chosen a value of  $m = 20$ . This simple model is sufficient to allow for measurable degrees of redundancy and degeneracy: Any two agents  $\mathbf{a}_i$  and  $\mathbf{a}_j$ ,  $i \neq j$ , are considered unique with respect to one another if  $\forall a_{ik} \in \mathbf{a}_i \Rightarrow a_{ik} \notin \mathbf{a}_j$ . Redundancy with respect to two agents, on the other hand, is defined as  $\forall a_{ik} \in \mathbf{a}_i \Rightarrow a_{ik} \in \mathbf{a}_j$ . If a pair of agents is neither unique nor redundant, it is considered degenerate. A system-wide measure of degeneracy (redundancy) of  $\mathcal{M}$  then corresponds to the fraction of all unique pair-wise comparisons of all agent pairings that are degenerate (redundant).

Each agent may devote its resources (e.g., time or energy) to the two tasks it is able to carry out. For instance, if agent  $\mathbf{a}_i$  is able to carry out tasks 1 and 2, it could devote 30% of its resources to task 1 and 70% to task 2. We subsequently define a global resource allocation vector  $\mathcal{R} = (\mathbf{r}_1, \dots, \mathbf{r}_n)$ , where each resource allocation  $\mathbf{r}_i$  is a pair  $(r_{i1}, r_{i2})$  with  $0 \leq r_{ij} \leq 1$  and  $r_{i1} + r_{i2} = 1$ ; the number  $r_{ij}$  denotes the fraction of resources that agent  $\mathbf{a}_i$  devotes to its task  $a_{ij}$ .

The available resources may be allocated dynamically using a *local decision-making process* without global control. In order to do so efficiently, we discretise the continuous range of each element  $r_{ij}$  into 11 segments  $\{0, \frac{1}{10}, \dots, 1\}$ . For each iteration of this procedure, we consider every element  $\mathbf{r}_i$  (without replacement) and perform a local search that systematically increases or decreases the value  $r_{i1}$  by  $\frac{1}{10}$ , doing the opposite for  $r_{i2}$  (such that  $r_{i1} + r_{i2} = 1$ ). We do this as long as the fitness of the MAS (see below) improves (or the bounds of  $r_{ij}$  have been reached). This step is repeated until no further improvements may be made across all elements of  $\mathcal{R}$ .

Each MAS is exposed to  $s = 10$  distinct scenarios at any one time: each scenario  $\mathbf{s}_i$  specifies a set of demands for each of the  $m$  task types,  $\mathbf{s}_i = (s_{i1}, \dots, s_{im})$  where  $0 \leq s_{ij} \leq n$ . We also impose that the sum of all demands equals the size of the MAS:  $\sum_{j=1}^m s_{ij} = n$ . In order to generate the  $s$  scenarios, a *seed scenario*  $\mathbf{s}_0$  is generated randomly and the remaining  $s - 1$  scenarios are then generated by means of a random walk of length 10 (volatility) that always starts from  $\mathbf{s}_0$ . For each step of the random walk, a pair of task-types is chosen uniformly at random and the demand for one of the chosen task-types is increased by a value of 1, the other is decreased by a value of 1 (subject to staying within bounds; if this operation should be unsuccessful, a new pairing of task-types would be chosen). It follows that the total demand of the scenario remains constant but its distribution changes. The set of environments changes every 200 generations (of the genetic algorithm; see below) either *moderately* or *drastically*. For moderate changes, the seed for the new set of scenarios is randomly selected from the previous set (excluding the original  $\mathbf{s}_0$ ). For drastic changes, on the other hand, a new seed scenario is generated uniformly at random. The remaining scenarios are generated as before.

The distribution of resources within a MAS (as described above) occurs as a direct response to the environmental conditions (i.e., demands) experienced by the system. We denote the output of a MAS by the vector  $\mathcal{O} = (o_1, \dots, o_n)$  where  $o_i$  is the sum of resources dedicated to task-type  $i$ :  $o_i = \sum_{j=1}^n \sum_{k=1}^2 r_{jk} \cdot [a_{jk} = i]$  where  $[\cdot]$  returns 1 if the containing statement is true. The *fitness* of a MAS under environment  $\mathbf{s}_i$  is then the difference between its output  $\mathcal{O}$  and the demand imposed by the environment  $\mathbf{s}_i$ :  $F(\mathcal{M}, \mathbf{s}_i) = \sum_{j=1}^m \max\{0, s_{ij} - o_j\}^2$  where  $o_j \in \mathcal{O}$  approximates an optimal allocation of resources under  $\mathbf{s}_i$  given the capabilities of  $\mathcal{M}$ . The *robustness* of the MAS is subsequently defined as the average fitness across all scenarios,  $R(\mathcal{M}, \{\mathbf{s}_1, \dots, \mathbf{s}_s\}) = \frac{1}{s} \sum_{j=1}^s F(\mathcal{M}, \mathbf{s}_j)$ . This measure was chosen for simplicity, although we found that robustness measurements that incorporated fitness thresholds did not appear to alter our basic findings.

The vector  $\mathcal{O}$  is obtained on-the-fly with respect to each  $\mathbf{s}_i$  encountered. However, the optimality of the resource allocation is strictly dependent on the task-types contained within the MAS. We thus use a genetic algorithm (GA) based on deterministic crowding to evolve a population of MAS (i.e.,  $\mathcal{M}$ ) towards a specific set of scenarios. Prior to the algorithm's execution,  $\frac{m}{2}$  unique agent-types (i.e., pairing of task-types) are constructed from the  $m = 20$  task-types and stored in a set  $T$ . The initial population  $P$ , of size  $N = 20$ , is then created by sampling (with replacement) from  $T$  to obtain a MAS that consist exclusively of pairwise unique or redundant agent-types.

During evolution, two parents are randomly selected from the population (without replacement) and subjected to uniform crossover (element-wise probability of 0.5) with probability 1. Each resulting offspring has exactly one element (agent-type) mutated and then replaces the genotypically more similar parent if its fitness is at least as good. Mutation changes the functional capabilities of a single agent and thereby determines whether degeneracy may arise during evolution. The mutation operator has been designed with the following considerations in mind: (a) the search space is to be of the same size in all experiments; (b) in some experiments both redundancy and degeneracy can be selected for during the evolutionary process.

Each position in  $\mathcal{M}$  is occupied by a specific agent-type and the mutation operator replaces exactly one such agent-type with a new one. The agent-types available at each position are determined *a priori* and remain constant throughout the execution of the algorithm. In the fully restricted case (no degeneracy), the options at *each* position are given by the set  $T$  (which was also used to initialise the population). It follows that a purely redundant MAS remains redundant after mutation. For experiments in which the MAS can evolve degenerate architectures, each position  $i$  has a *unique* set of options  $T_i'$  which closely resembles the set  $T$  but allows for a partial overlap in functions: Each  $T_i'$  contains the same task-types as  $T$  but half its members (chosen randomly) have exactly one element per task-type pairing altered randomly. The mutation operator is illustrated in Fig. 1b: agents from both system classes have access to the same number of task type pairings (mutation options are shown as faded task type pairings), hence the search space sizes are identical. In the redundant case, mutation options are defined in order to prevent degeneracy. In the degenerate case, it is evident that the agents' capabilities may be unique, redundant, or may partially overlap due to slightly altered task type pairings for each agent.

## 4 Experimental Results

In our experiments, a MAS architecture (i.e. the specification of agent task capabilities) evolves to maximise robustness within a set of environmental scenarios. To evaluate Edelman and Gally’s hypothesis, we place different restrictions on the architectural properties that can evolve in a MAS (see mutation operator in Section 3), preventing degeneracy from arising in some cases. We then evaluate if degeneracy improves adaptation properties during static and dynamic environmental conditions.

### 4.1 Robustness, Evolvability in Static (Heterogeneous) Environments

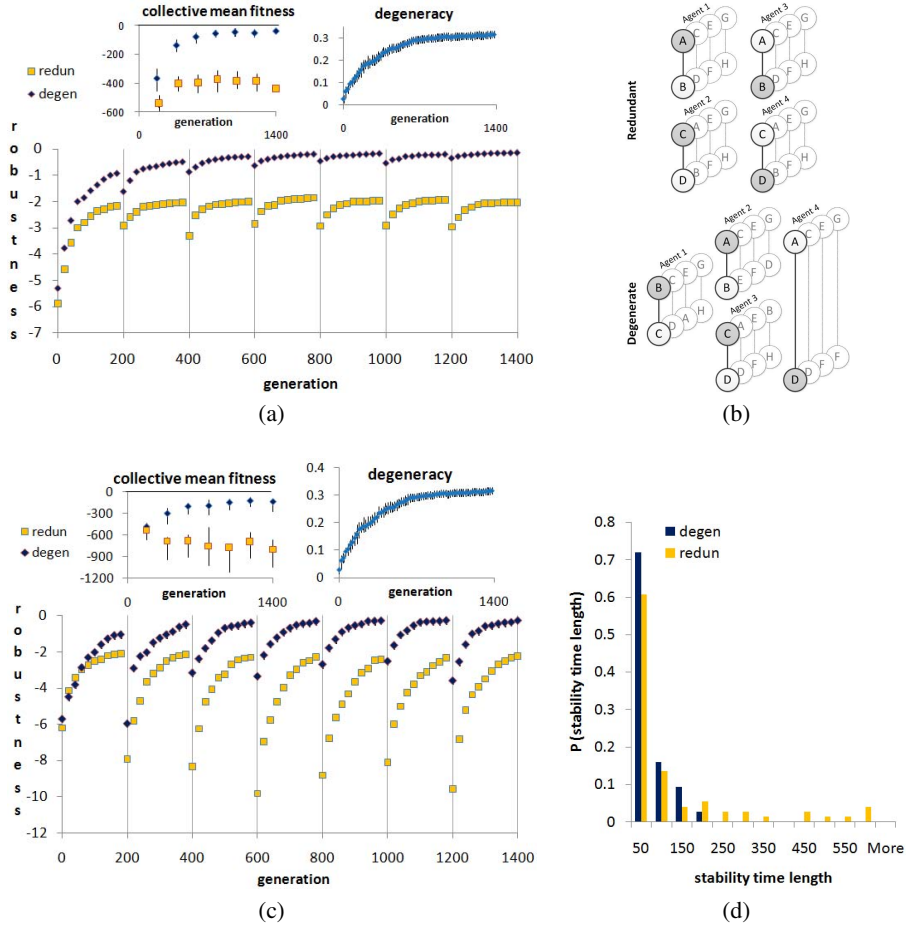
In Fig. 1a, for the 200 generations before the environment changes we see that, when degeneracy is allowed to emerge, the MAS evolves higher robustness towards the set of environmental scenarios. This finding is not intuitively expected considering that: systems are the same size (and solution spaces are constrained to identical sizes), MAS are presented with the same scenarios, agents have access to the same task types and, within a noiseless environment, all MAS evolve within a unimodal fitness landscape that contains the same optimal fitness value. In our view, there are two factors that primarily contribute to these observed differences in evolved robustness: 1) evolvability within the static noisy environment (discussed below) and 2) differences in the robustness potential of a system (discussed in the networked buffering hypothesis in [17]).

Conceptually, evolvability is about discovering heritable phenotypic improvements. In Fig. 1d, we record the probability distribution for the time it takes the MAS population to find a better solution. As can be seen, degenerate architectures are finding adaptive improvements more quickly. An analysis of improvement size vs fitness finds this relationship is similar for the two types of MAS, thus suggesting the faster adaptation rate is largely responsible for the divergence in fitness from generation 0 to 200.

### 4.2 Evolvability in Dynamic Environments

In a dynamic environment, evolvability is no longer merely about a propensity for discovering improvements but is also about sustaining high fitness throughout and after the environment changes. As can be seen from Fig. 1a and c, in both redundant and degenerate MAS, robustness drops every 200 generations when environmental change is imposed. This drop reflects declines in fitness across the population. However, we can make the following noteworthy observations.

When evolution of degeneracy is enabled, MAS populations can adapt better to change than MAS with purely redundant architectures. Except for the decline of fitness at generation 200, all subsequent drops (at generations 400, 600, etc.) are smaller in the ‘degen’ experiments than in the ‘redun’ experiments, irrespective of whether the scenario changes are moderate or drastic. With every change in the set of scenarios, MAS that cannot evolve degenerate architectures appear to drop in performance by similar amounts. The only exception is the first adaptation after a change of environmental conditions (i.e. the time period from generation 201 to 400) where there is some overall improvement when the environmental change is moderate (Fig. 1a), and some overall deterioration when the change is drastic (Fig. 1c). MAS that can evolve degeneracy, on the other hand, have some capacity to adapt to the nature of change. From



**Fig. 1.** Figure 1 (a): When degeneracy is/is not permitted in the MAS architecture, we label these as ‘degen’/‘redun’. The main graph plots robustness evolved over time with smaller graphs of collective mean fitness (top-left) and degeneracy (for the MAS where it is allowed to emerge, top-right). Environmental changes (every 200 gen.) are moderate (see Experimental Setup). (b): Degenerate and redundant MAS. Agents (depicted as pairs of connected nodes) can perform 2 different task types. Each MAS (top: redundant; bottom: degenerate) consists of 4 agents and the faded pairings indicate the predetermined set of options the mutation operator may choose from. (c): MAS evolve in conditions where environmental changes are dramatic. (d): histogram for the number of offspring sampled before an improvement is found (stability time length). Conditions are the same as (a) except environmental changes occur every 400 generations.

environmental change to environmental change, the drop in fitness/robustness becomes smaller.

When plotting the *collective mean fitness* (i.e. the area under the fitness/robustness curve between two consecutive environmental changes [8]), we do not only observe this adaptation in experiments with moderately changing environments (top-left graph

in Fig. 1a) but we also see overall adaptation levels improve over time even when the environmental changes are drastically different (top-left graph in Fig 1c). Comparing this with the amount of degeneracy integrated within the MAS architecture (top-right graphs in Fig. 1a and c), we see that the collective mean fitness improves as degeneracy is integrated into the system. Furthermore, the degeneracy-enabled capacity to adapt is better when changes in the environment are moderate or correlated; a proposed precondition for continuous adaptation in DOP (see [2]). It is admittedly difficult however to directly evaluate changes in the rate of adaptation (e.g. as we did for a static environment in Fig. 1d) in the dynamic case because fitness differences at the beginning of each epoch act to confound such an analysis. We note however that in somewhat similar MAS models, experimental conditions were established that can more clearly demonstrate an acceleration in adaptation rates during degenerate MAS evolution within a dynamic environment [12].

When we make the scale of our model larger (i.e. by increasing MAS size,  $T$ , and random walk size by the same proportion), the differences between degenerate and redundant MAS in robustness, evolvability and collective mean fitness become accentuated. Future studies guided by selected MAS application domains will aim to further investigate the generality and limitations of these findings by considering: restrictions in functional capability combinations in each agent, different classes of environment correlation, the speed of agent behavior modification, costs in agent behavior modification, and agent-agent functional interdependencies.

## 5 Discussion and Conclusions

In this paper, we investigated the potential for designing dynamic optimisation problem (DOP) representations that are robust to environmental conditions experienced during a solution's lifecycle and, at the same time, have the capacity to adapt to changing environments. Our investigation was motivated by a hypothesis formulated in the context of biological evolution – namely that degeneracy facilitates robustness and adaptation in time-variant environments. In simulation experiments we evolved populations of multi-agent systems (MAS) and compared the robustness and adaptation potentials of systems that could evolve degenerate architectures with those that could evolve redundant structures only. We found evidence that incorporating degeneracy into a problem's representation can improve robustness and adaptiveness of dynamic optimisation in ways that are not seen in purely redundant problem representations.

While our investigation was quite abstract, we can identify several features that make degeneracy suitable for dynamic optimisation. First, degenerate systems appear to exhibit a greater propensity to adapt. While we have not reported an analysis of fitness landscape neutrality here, previous studies on the ensemble properties of similar models have shown degeneracy creates neutral regions in fitness landscapes with high access to phenotypic variety [15,16]. In light of these earlier studies, the results presented here demonstrate that the discovery of adaptations in static neutral landscapes created by degeneracy can be surprisingly rapid. Theoretical arguments have suggested that long periods of time may be needed to discover a single adaptive phenotype from a neutral network [11], however the rapid adaptation in Fig. 1a,d suggests that little neutrality is



ever traversed in these experiments before an improvement is discovered. As believed to also take place in biology, this fast pace of adaptation likely reflects the existence of many alternative paths to adaptive change within neutral networks created by degeneracy. This means that little of the neutral network needs to be searched before new improvements are found, thus fitness barriers are not being replaced with large “entropic barriers” during evolution, cf [11]. While optimal solutions are not guaranteed, the propensity to adapt in evolved degenerate systems appears to allow such a strategy to quickly find highly fit and highly robust solutions – as needed when tackling DOP.

A second desirable feature of degenerate systems is their enhanced capacity to deal with novel conditions. Compared with redundant architectures, degenerate systems have a greater potential to evolve innovative solution responses that account for small variations in environmental conditions. In a supplemental analysis of these systems we have found this robustness potential can extend to moderate degrees of environmental novelty, thus helping to explain the differences between system classes immediately after a change in the environment (Fig. 1a,c). However, a further reason that degenerate MAS exhibited highly effective responses to immediate environmental change was the emergence of population properties known in evolutionary biology as cryptic genetic variation (CGV).

Many EA-based dynamic optimisation techniques aim to artificially control population convergence based on a general understanding that low genetic diversity limits a population’s adaptability when it encounters a changed fitness landscape. The resulting genetic and phenotypic properties of EA populations differ significantly however from that observed in natural populations. Genetic diversity within natural populations is maintained in a static environment by being phenotypically and selectively hidden. Trait differences across the population are mostly exposed only after an environment changes; a phenomena known as *cryptic genetic variation* (CGV). The present study focuses on how Edelman and Gally’s hypothesis is relevant when applying neutral evolution theories to the topic of evolvable problem representations. However in [13] we also analyze the population properties from these experiments and report evidence that degeneracy generates hide and release mechanisms for genetic diversity that are analogous to the natural CGV phenomena just described. This evidence of CGV is presented as a separate supplemental report in [13] due to space limitations as well as its distinctive theoretical relevance.

## Acknowledgements

This work was partially supported by a DSTO grant on “Fleet Designs for Robustness and Adaptiveness” and an EPSRC grant (No. EP/E058884/1) on “Evolutionary Algorithms for Dynamic Optimisation Problems: Design, Analysis and Applications.”

## References

1. Banzhaf, W.: Genotype-phenotype-mapping and neutral variation-a case study in genetic programming. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 322–332. Springer, Heidelberg (1994)

2. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Congress on Evolutionary Computation, vol. 3, pp. 1875–1882. IEEE Computer Society, Washington (1999)
3. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer, Dordrecht (2002)
4. Ciliberti, S., Martin, O.C., Wagner, A.: Innovation and robustness in complex regulatory gene networks. Proceedings of the National Academy of Sciences, USA 104(34), 13591–13596 (2007)
5. Edelman, G.M., Gally, J.A.: Degeneracy and complexity in biological systems. Proceedings of the National Academy of Sciences, USA 98(24), 13763–13768 (2001)
6. Jin, Y., Gruna, R., Paenke, I., Sendhoff, B.: Evolutionary multi-objective optimization of robustness and innovation in redundant genetic representations. In: IEEE Symposium on Computational Intelligence in Multi-Criteria Decision Making, Nashville, USA, pp. 38–42 (2009)
7. Knowles, J.D., Watson, R.A.: On the utility of redundant encodings in mutation-based evolutionary search. LNCS, pp. 88–98. Springer, Heidelberg (2003)
8. Morrison, R.W.: Designing Evolutionary Algorithms for Dynamic Environments. Springer, Berlin (2004)
9. Rothlauf, F.: Representations for Genetic and Evolutionary Algorithms, 2nd edn. Springer, Heidelberg (2006)
10. Rothlauf, F., Goldberg, D.E.: Redundant representations in evolutionary computation. Evolutionary Computation 11(4), 381–415 (2003)
11. van Nimwegen, E., Crutchfield, J.P.: Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths? Bulletin of Mathematical Biology 62(5), 799–848 (2000)
12. Whitacre, J.M.: Evolution-inspired approaches for engineering emergent robustness in an uncertain dynamic world. In: Proceedings of the Conference on Artificial Life XII (to appear)
13. Whitacre, J.M.: Genetic and environment-induced innovation: complementary pathways to adaptive change that are facilitated by degeneracy in multi-agent systems. In: Proceedings of the Conference on Artificial Life XII (to appear)
14. Whitacre, J.M.: Degeneracy: a link between evolvability, robustness and complexity in biological systems. Theoretical Biology and Medical Modelling 7(1), 6 (2010)
15. Whitacre, J.M., Bender, A.: Degenerate neutrality creates evolvable fitness landscapes. In: WorldComp 2009, July 13–16 (2009)
16. Whitacre, J.M., Bender, A.: Degeneracy: a design principle for achieving robustness and evolvability. Journal of Theoretical Biology 263(1), 143–153 (2010)
17. Whitacre, J.M., Bender, A.: Networked buffering: a basic mechanism for distributed robustness in complex adaptive systems. Theoretical Biology and Medical Modelling 7(20) (to appear 2010)
18. Yu, T., Miller, J.F.: Neutrality and the evolvability of boolean function landscape. In: Proceedings of the 4th European Conference on Genetic Programming, pp. 204–217. Springer, London (2001)