Evaluating Resilience Strategies Based on an Evolutionary Multi-agent System

Kazuhiro Minami, Tomoya Tanjo, and Hiroshi Maruyama
Institute of Statistical Mathematics, Tokyo, Japan
Email: {kminami, tanjo, hm2}@ism.ac.jp

***Abstract*—Many social infrastructures fail in an unexpected way, and thus it is important to make a system resilient such that it can recover from various damages in a dynamic and flexible way. There seems to be a consensus among many researchers in this field that redundancy, diversity, and adaptability are the three key characteristics of resilient systems that can absorb shocks and recover from damages gracefully. However, it is still not clear how we quantitatively combine those strategies to build a resilient system of a given domain. In this paper, we present an evolutionary multi-agent system for evaluating the effectiveness of those strategies and show several insights drawn from our preliminary results.**

I. Introduction

After the 3.11 earthquake of Japan in 2011, many people realized that there are events that cannot be reasonably anticipated. These unexpected events occur as an outside of the anticipated envelope (e.g., Tsunami of 14m high vs the anticipated max of 5.7m), or something completely unheard of (e.g., Tokyo subway gas attack in 1995). We recognize that these unexpected events do happen, but because they are unexpected, we cannot prepare for the event and protect our systems. The only thing we can do is to give resistance that contains damages from the event locally and recover from those damages as quickly and as inexpensively as possible. We call this combination of resistance and recovery the *resilience* of the system.

Many researchers have recognized the importance of establishing a new research discipline concerning the resilience of complex systems to provide a set of general principles for building resilient systems in various fields. We have seen many examples of a resilient system in computer science and biology, and identified three major characteristics (i.e., redundancy, diversity, and adaptability) common in resilient systems [1]. However, building a resilient system requires additional cost, and it is still not clear how we should combine those three strategies to build a resilient system in a given domain subject to a given budget constraint. For example, Toyota auto company has been achieving the resilience of its supply management system by putting precedence on adaptability over redundancy [2].

Therefore, we evaluate combinations of different design strategies for resilient systems based on an evolutionary multi-agent system. Each agent in the system is a digital organism [3] that can self-replicate, mutate, or evolve, so that we can perform experiments on scale that are beyond reach with any biological entity. We choose such a multi-agent system as our testbed since many complex systems in various domains, such as the Internet, have been implemented as a set of autonomous self-organized entities to achieve its resilience.

We conduct various multi-agent simulations while changing system parameters that quantify the three design strategies of resilient systems to obtain detailed measurements of resilience properties of time-changing populations. Our focus is to identify key parameters that make an agent population, which represents a decentralized complex system, resilient to a changing environment rather than to study evolution processes of an individual agent. We report from our experiments initial insights regarding design strategies for building resilient systems.

The rest of the paper is organized as follows. Section II describes our multi-agent system and introduce important resilience properties of the system. Section III presents our preliminary results based on multi-agent simulation. Section IV discusses related work and Section V concludes.

II. Evolutionary multi-agent system

In this section, we describe our evolutionary multi-agent system and our metrics for resilience properties of the system. Our formalism is based on the dynamic constraint satisfaction problem described in [4].

A. System structure

A multi-agent system consists of multiple agents representing a digital organism [3]. The system incorporates evolutionary processes of agents in a population [5], [6]. Each agent a_i is encoded as a binary string b_i of a given length l, and each bit of the string represents some phenotype (e.g., 2-leg or 4-leg) of that agent. We refer to the binary string of an agent as its *configuration* throughout the rest of the paper. We denote by $\mathcal{A}$ the set of all agent configurations $\{0,1\}^l$. Each agent a_i maintains a resource r_i, which is a numerical value in $\mathcal{N} = \{0, 1, \ldots\}$, Each agent can use its resource to survive a severe environment. The notion of resouces in our system is the similar to that of *energy* in [5].

We represent an environment of agents by a constraint that determines whether a given agent a_i fits into the corresponding environment. A constraint c is a subset of the set of all agent configurations $\mathcal{A}$; i.e., $c \subset \mathcal{A}$. We consider that an agent a_i is *fit* with a constraint c if $a_i \in c$; otherwise, we say that a_i is *unfit*. That is, a constraint in our system corresponds to a binary fitness function in genetic algorithms [7].

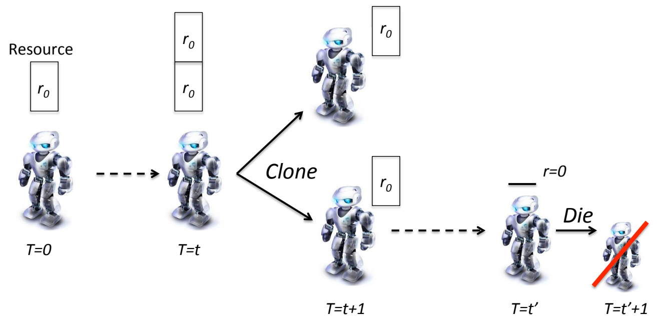


Fig. 1. An example life cycle of an agent. A agent is given an initial resource r_0 at time t_0. When the agent doubles its resource, it creates its clone and splits the resource accordingly. When one of the agents uses up its resource at time t', it dies.

B. System transition

Each agent of the system evolves itself over time. The state of an agent a_i at time t consists of a pair (b_i^t, r_i^t) of a configuration and a resource. Similarly, a constraint c changes over time. We denote a constraint at time t by c^t. We consider such a dynamic constraint to study the resilience of an agent population to a changing environment, while the majority of previous research on evolutionary systems (e.g., genetic algorithms) focuses on solving an optimization problem subject to a static constraint.

Each agent a_i changes its configuration b_i^t at every time t by flipping some of the bits in its configuration, in order to adapt to a changing constraint. This adaptation process corresponds to mutations on genomes of a biological organism. However, since we aim to study a population of any artificial artifacts (e.g., robots), we assume that an agent can apply any adaptation strategy that intelligently considers the surrounding environment. Each agent applies any adaptation strategy of its choice at time t if $a_i^t \notin c^t$. Thus, the mutation rates that we use in our experiments in Section III are much higher than those of biochemical organisms. The goal of each agent is to survive in the population as long as possible by adapting to a dynamically changing environment.

Unlike other computational approaches to the study of evolutionary systems (e.g., genetic algorithms [8]), agents in our system representing digital organisms explicitly create a copy of their own to reproduce. Each agent a_i's resource changes dynamically. If a_i^t belongs to c^t, r_i^{t+1} will be incremented as a reward from r_i^t by a specified delta d; otherwise, r_i^t is decremented by d. An agent with a sufficient resource at time t can make its clone and give the half of its resource to the newly created clone. Those two identical agents apply the adaptation strategy to themselves and prepare their configuration for time $t+1$. If an agent uses up all resources at time t, that agent dies at time $t+1$. Figure 1 shows an example life cycle of an agent in our system. Notice that the number of agents in our system could grow or shrink dynamically.

C. Resilience metrics

We now describe how we quantify resilience properties of the system, redundancy, diversity, and adaptability. First, we consider the amount of a resource owned by an agent as the redundancy factor. An agent can remain alive until it uses up its resources even if it does not satisfy a constraint for a certain period. Second, we measure the diversity of a population consisting of multiple agents with the Simpson's diversity index [9] below.

$$D = 1/\Sigma_{i=1}^{M}(n_i/N)^2$$

where M is the number of different configuration of agents in the population, N is the total number of agents, and n_i is the number of agents with the same configuration i. Third, we quantify the speed of an adaptation by the number of bits an agent can flip at a time.

III. Simulation results

We conduct various multi-agent simulations changing various system parameters including those regarding the resilience properties described in Section II-C. We use a multi-agent system written in D programming language.

A. System parameters

Table I shows a list of system parameters in our experiments. We start each simulation with the initial number of 100 agents. Each agent, which is encoded as a 8-bit or 16-bit string, owns ten resources initially. We randomly choose a given number of configurations as the members of an initial constraint.

We consider two types of constraints. One is a *connected* constraint in which all member configurations are connected as a graph. We consider that each configuration is a node in a graph and two nodes are connected if the Hamming distance of the two corresponding configurations is one. We say that a constraint is connected if every two configurations c_1 and c_2 in the constraint has a path from c_1 to c_2. Intuitively, we can visualize a connected constraint as a single island in the agent configuration space. The other is a *unconnected* constraint in which configurations do not necessarily compose a connected graph.

Further, we consider two types of constraint transitions that specify how a constraint changes over time. One is called a *random* transition where a new constraint of the initial size is created randomly replacing the previous constraint. Such a random transition represents a sudden disruptive change of an environment. The other is called a *continuous* transition where a new constraint for time t is created from the previous constraint by adding or removing a certain number of constraints.

We consider two adaptation strategies: one is a *random* adaptation strategy where an agent chooses a specified number of bits in its configuration randomly and flips them to create its new configuration at the next time. The other is an *intelligent* adaptation strategy where an agent flips a specified number of bits in its configuration to move to the closest point to the nearest configuration in the current constraint. The speed of

TABLE I
SYSTEM PARAMETERS

Parameter	Value
Initial population size	100
Agent bit length	{8,16}
Initial resource	10
Initial constraint size	{13,26}
Constraint connectivity	{true, false}
Constraint transition	{random, continuous}
Constraint change range	$[n, n]$ where $n \in \mathcal{N}^+$
Adaptation strategy	{random, intelligent}
Adaptation speed	1
Diversity index	{1,10,50}

an agent's adaptation corresponds to the number of bit flips in those two strategies.

Finally, we specify the diversity index of an *initial* agent population. We randomly generate an initial set of agents with a given value of the diversity index.

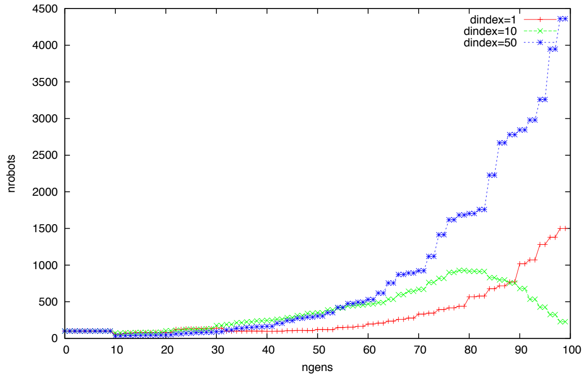
B. Effects of population diversity

We examine how the diversity index of the set of agents in a given initial population affects its resilience against a time-varying constraint. Each agent with a 8-bit string applies a random adaptation strategy that flips a single bit at a time. We use a connected constraint, which changes continuously over time. The size of an initial constraint is 26, which is roughly 10% of the set of all agents $\mathcal{A}$. We create a new constraint at time t by adding n new configurations or removing n configurations from the constraint at time $t-1$. We randomly pick such n from the range of integers in $[0, 3]$.

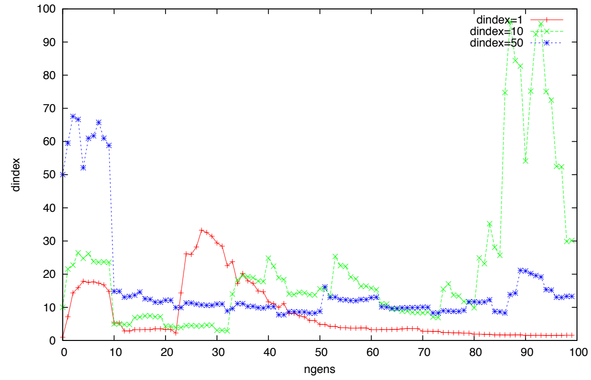
Figure 2 compares experiments with three different values of initial diversity indexes, 1, 10, and 50. Figure 2(a) shows the transition of the number of agents over time in each case. We compute 100 time steps in each simulation. The run with the largest diversity index of 50 shows the fastest growth of an agent population while the run with that of 10 shows the near extinction of its agent population. We observe that, if an initial population of agents has sufficient diversity, it is highly likely that enough number of agents belong to an initial constraint such that they accumulate their resources and make their clones quickly.

The run with the diversity index of 1 performs better than that of 10. We suspect that this is because a sufficient number of agents in the population of the diversity index 1 happen to belong to the initial constraint by chance. Figure 2(c) shows that the ratio of fit agents in the population of the diversity index 1 is almost same as that of diversity index 10 at the early stages of those runs.

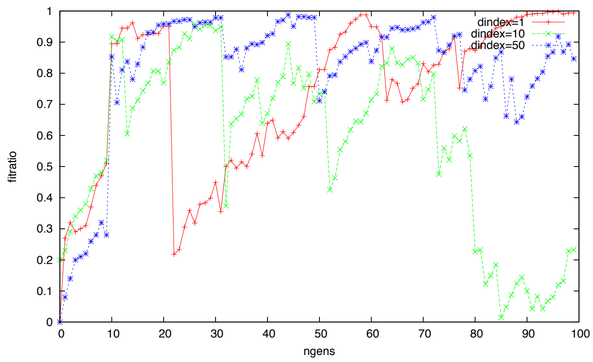
Figure 2(b) shows the transition of the diversity index of an agent population. We see that an agent population with the initial diversity index of 50 quickly loses its diversity because agents that do not belong to a constraint die at an early state of the run. The loss of diversity in the agent population does not do any harm on its sustainability since the constraint changes continuously so that agents in the current constraint can easily adapt to a new constraint.



(a) Transition of the number of agents.



(b) Transition of the diversity index.



(c) Transition of the ratio of fit agents in the population.

Fig. 2. Comparison with different initial diversity indexes. Constraint connectivity: true, Constraint transition: continuous, Constraint change range: [-3, 3], #constraints: 26, Agent bit length: 8, Fit reward: 1, Adaptation strategy: random, Adaptation speed: 1.

The agent population with the initial diversity index 10 has a high diversity index between time 80 and 100. This is possibly because the majority of agents in that population do not belong to the constraints during that period and thus they need to adapt themselves by flipping their configurations randomly causing the increase of their diversity. Our agents do not apply the adaptation strategy if they are fit belonging to the current constraint.

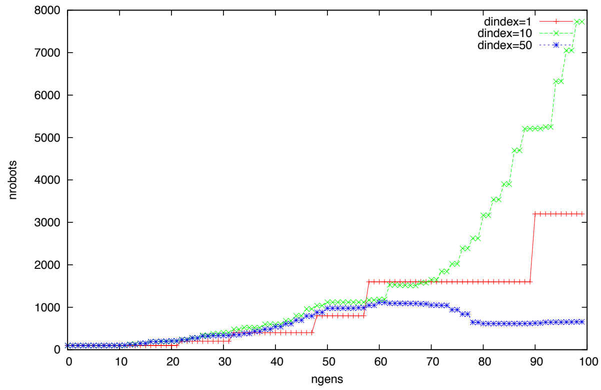


Fig. 3. Improvement with the intelligent adaptation strategy. Constraint connectivity: true, Constraint transition: gradual, Constraint change range: [-3, 3], #constraints: 26, Agent bit length: 8, Fit reward: 1, Adaptation strategy: intelligent, Adaptation speed: 1.

C. *Effects of adaptation strategies*

Figure 3 shows how results would be different if each agent adopts the intelligent strategy. We use the same set of system parameters as in the previous experiments in Figure 2. The results show that, regardless of the value of the diversity index, an agent population grows roughly as twice as fast as that in the previous cases with the random strategy.

D. *Effects of resource sharing*

We also consider the case where all the agents share the common resource, and evaluate how this resource sharing scheme affects the resilience of an agent population [1]. We set the value of the initial common resource to 1000, which is the multiplication of an individual's separate resource (i.e., 10) and the initial number of agents (i.e., 100). If the value of the common resource is increased by 10, which is the initial value for an agent's separate resource, we randomly pick one agent from the current population and create its clone. Similarly, if the common resource value becomes less than that of the initial value for the current number of agents, we remove an agent randomly chosen from the current population. That is, if the common resource becomes less than $10n$ where n is the size of current population, we remove one agent from the population.

We select different system parameters from those in the previous experiments in order to arrange a much severe environment for an agent population. That is, the length of an agent's binary string is 16 rather than 8, which roughly doubles the time to reach a configuration in a constraint. The size of an initial constraint is 13 rather than 26, which is about 0.0012% of all agent configurations. A constraint, which is not connected, changes randomly once every ten time steps and changes continuously within the range of [-7, 7] at the other times. A random constraint transition represents a disruptive change of an environment.

[1]Since agents in the real world share distributed resources locally, we plan to extend our resource sharing scheme to a more distributed setting as our future work.

Figure 4 compares the results of the two resource management schemes. We use the same set of system parameters mentioned above except for the resource management scheme. Figure 4(a) shows the case where each agent separately manages its resource. Regardless of the value of initial diversity index, an agent population goes extinct quickly. Figure 4(c) shows the case where all agents use the common resource. The agent populations with the diversity indexes of 10 and 50 survive periodic disruptive changes of their environments and start growing their populations after time 60.

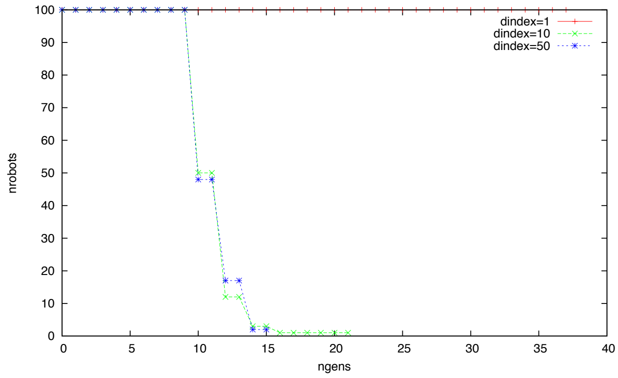
There are two possible reasons that explain the difference of the two resource management schemes. First, the value of the common resource changes over time much slowly than that of an individual resource since agents both in a constraint and outside the constraint cancel out their updates in a mutual way. Second, we choose the liberal way of adding or removing an agent randomly chosen from the population as the value of the common resource is changed. Since we choose an agent to be added or removed without considering its fitness, the agent population maintains a relatively high degree of diversity, which makes the system more adaptive to a sudden change arranged in this experiment, as shown in Figure 4(d). On the other hand, Figure 4(b) shows that the agent populations lose their diversity quickly when each agent maintains its resource separately.

IV. Related work

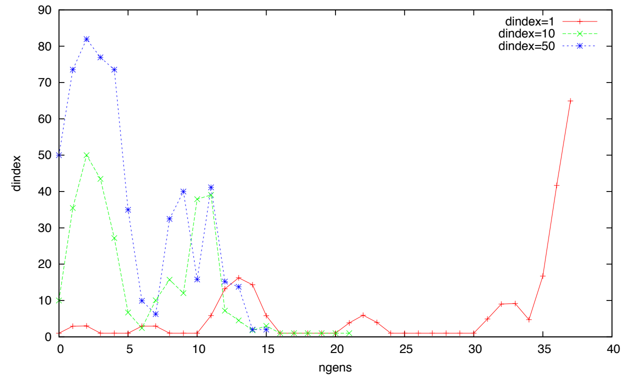
Many researchers have applied an evolutionary approach to study the dynamics of digital organisms [3] or virtual robots [10] in a population. However, the main focus of those research is the evolution of the fittest individual, which corresponds to a solution of a given optimization problem. Such an optimization problem is usually represented as a static constraint. On the other hand, our focus is an evolutionary ecology of artificial artifacts; we study desirable global properties of such an ecological system that makes it resilient to dynamically changing environmentals as opposed to static environments representing constraints of an optimization problem.

Avida [11] is a software platform for experiments with self-replicating and evolving computer programs. In Avida, a digital organism is represented as a collection of computer instructions. Several researchers conduct simulations based on Avida to study evolutionary processes of digital organisms. For example, Lenki et al. [12] studied the origin of complex organismic features, such as the eye, in the process of biological evolution, by conducting simulations on digital organisms. They show that a digital organism that performs a complex logical operation is derived through those with more primitive operations. Their focus is to study mechanisms for deriving complex features in evolutionary processes, not the resilience of an agent population to a changing environment.

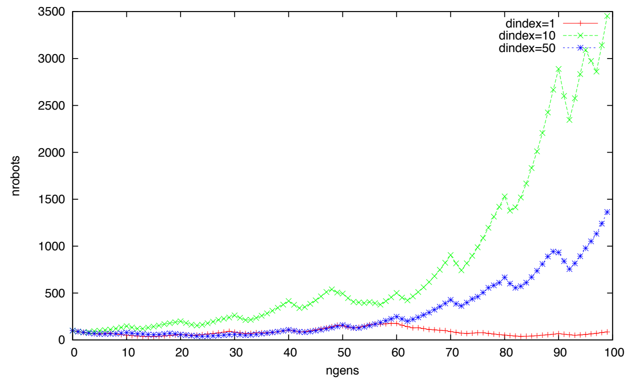
Researchers in the field of evolutionary robotics have been studying evolutionary algorithms for a population of virtual robots to automate design processes of robots. Lipson et al.[13] conduct simulations of an evolutionary process of robots that



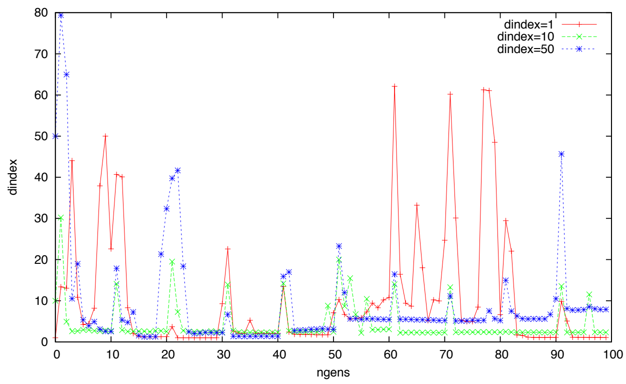
(a) Transition of the number of agents with the no resource sharing schemed.



(b) Transition of the diversity index with the no resource sharing schemed.



(c) Transition of the number of agents with the resource sharing schemed.



(d) Transition of the diversity index with the resource sharing schemed.

Fig. 4. Comparison of resource sharing schemes. Constraint connectivity: false, Constraint transition: (random: once at every ten time steps; continuous: the rest of the steps), Constraint change range: [-7, 7], #constraints: 13, Agent bit length: 16, Fit reward: 1, Adaptation strategy: intelligent, Adaptation speed: 1.

consist of basic building blocks (bars, actuators, and artificial neurons) to obtain an optimal design automatically. However, the focus of this line of research is limited to find a better design for a particular set of relatively static requirements.

V. Summary

We develop an evolutionary multi-agent system to evaluate the resilience of multi-agent systems. We identify three major strategies, redundancy, diversity, and adaptability, for building resilient systems and introduce corresponding quantitative system parameters to evaluate the effectiveness of those strategies.

Our preliminary results show that the diversity of an initial agent population is one of the key factors to make the system sustainable over time. We find that resource sharing among agents is an effective way to achieve a redundancy mechanism effectively and that the liberal way to selecting agents to be survived for the next generation is helpful to maintain the diversity of the population.

Acknowledgments

This research is partly supported by the Grants-in-Aid for Scientific Research B, 24300031, of Japan Society for the Promotion of Science.

References

[1] H. Maruyama, "Towards systems resilience," in *In Proceedings of the 1st Workshop on Systems Resilience (WSR 2013)*, 2013.
[2] A. Iyer, S. Seshadri, and R. Vasher, *Toyota Supply Chain Management: A Strategic Approach to Toyota's Renowned System*. McGraw-Hill.
[3] C. O. Wilke and C. Adami, "The biology of digital organisms," vol. 17, pp. 528–532, 2002.
[4] N. Schwind, T. Okimoto, K. Inoue, H. Chan, T. Ribeiro, K. Minami, and H. Maruyama, "Systems resilience: a challenge problem for dynamic constraint-based agent systems," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, ser. AAMAS '13. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 785–788. [Online]. Available: http://dl.acm.org/citation.cfm?id=2484920.2485043
[5] M. Kisiel-Dorohinicki, "Agent-oriented model of simulated evolution," in *Proceedings of the 29th Conference on Current Trends in Theory and Practice of Informatics: Theory and Practice of Informatics*, ser. SOFSEM '02. London, UK, UK: Springer-Verlag, 2002, pp. 253–261. [Online]. Available: http://dl.acm.org/citation.cfm?id=647012.712861
[6] P. J. t Hoen and E. D. de Jong, "Evolutionary multi-agent systems," in *In Proceedings of the 8th International Conference on Parallel Problem Solving from Nature – PPSN-04*, 2004, pp. 872–881.
[7] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
[8] J. A. Foster, "Evolutionary computation," *Nature Reviews Genetics*, vol. 2, pp. 428–436, June 2001.
[9] E. H. Simpson, "Measurement of diversity," vol. 163, p. 688, 1949.
[10] J. C. Bongard, "Evolutionary robotics," *Commun. ACM*, vol. 56, no. 8, pp. 74–83, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2493883
[11] C. Ofria, D. M. Bryson, and C. O. Wilke, "Avida: A software platform for research in computational evolutionary biology," *Artificial Life*, vol. 10, pp. 10–191, 2004.
[12] R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami, "The evolutionary origin of complex features," *Nature*, vol. 423, no. 6936, pp. 139–144, May 2003. [Online]. Available: http://dx.doi.org/10.1038/nature01568
[13] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, no. 406, pp. 974–978, 31 August 2000.