Self-Protection and Diversity in Self-Replicating Cellular Automata

Hiroki Sayama
Department of Human Communication
University of Electro-Communications
1-5-1 Chofugaoka
Chofu, Tokyo 182-8585, Japan
sayama@hc.uec.ac.jp

Abstract The concept of *self-protection*, a capability of an organism to protect itself from exogenous attacks, is introduced into the design of artificial evolutionary systems as a possible method to create and maintain diversity in the population. Three different mechanisms of self-protection are considered and implemented on a cellular-automaton-based evolutionary system, the *evoloop*. Simulation results imply a positive effect of those mechanisms on diversity maintenance, especially when the self-protection is moderate so that it conserves both the attacker and the attacked. This letter briefly reports the models and the simulation results obtained using those models.

Keywords
Self-protection, self-replication, cellular automata, evoloop, diversity

1 Introduction

The maintenance of diversity is as crucial for the design of artificial evolutionary systems as for conservation efforts in real ecology, because of its importance to the evolutionary robustness and exploration capability of the system [5, 8]. It is often found in various artificial systems that a simple homogeneous environment typically results in simple dominance by one and only one optimal type of organism [20]. A key to the creation and maintenance of diversity is interaction between organisms that permits them to change each other's fitness landscapes. Several mathematical models of such processes have been studied in statistical physics [2, 4], where interdependence between different types (genotypes and/or phenotypes) may create a multimodal distribution of organisms in type space. Similar diversification phenomena have also been realized in artificial evolutionary systems [14, 19]. In those studies, the mechanism of interaction between different types is introduced as previously defined operations or computational procedures explicitly given in the models.

However, things become much more complicated when one tries to introduce such interdependence into much simpler models where organismal actions are implemented bottom-up by a coordination of tiny lower-level rules, as in cellular automata. It is indeed a challenging question how one can construct an evolutionary system with complex interdependent fitness landscapes using cellular automata, as originally discussed by von Neumann [17, 18] and more recently studied with other, smaller models [3, 7, 11, 12]. This question is challenging because the atomic components of organisms in those models have only localized capability of communication and computation. Interaction between organisms therefore must be an emergent phenomenon realized at a far higher scale than the component scale.

While the above-mentioned question remains an interesting topic to address, here we present a rather different approach. We introduce the concept of what we call *self-protection*—the capability of an organism to protect itself from exogenous attacks—into the design of artificial evolutionary systems as a possible method to create and maintain diversity in the population. We then apply this concept to a simple evolutionary system based on cellular automata, the *evoloop* [11], to see its effect on the diversity emerging in the system. Simulations are conducted for three different versions of self-protection mechanisms, with several variations added to experimental settings to make sure of the robustness of the effects observed. This letter reports the outline of the models and the results obtained so far, with a brief discussion of possible implications of this study.

2 Self-Protection

Here we define self-protection, quite intuitively for now, as an action or a set of actions taken by an organism to protect its structure and/or function from attacks that may come from the external world, such as the nonliving environment or other organisms in competition. Organisms with this capability should generally have an evolutionary advantage over those without it.[1] To support this conjecture, some of the major evolutionary innovations seen in the actual history of life, such as evolution of templates, error correction, or partitioning (membranes), seem to have direct or indirect relevance to this concept [9].

Self-protection may be achieved in several distinct ways, including:

Barrier: Create a strong static barrier that isolates the internal self from the external world and stabilizes local processes.

Camouflage: Avoid being a target of attacks by adopting a specially designed appearance on the interface with the external world.

Reaction: Actively detect attacks and take necessary reactions to them (counterattacks, self-repair, etc.) to revert internal conditions to the original state.

Prevention: Communicate with other organisms in advance to cause them not to attack.

Redundancy: Maintain spatio-temporal redundancy in its structure and/or function so that local damage by an attack will not cause a serious problem to the entire organism.

Note that the listing above is just a collection of examples, and there must be many other ways to attain the same self-protection goals. The aim of this letter is not to discuss details of these strategies; rather, our focus is on how such mechanisms would affect the diversity emerging in artificial evolutionary systems. We examine this using simple cellular automaton models in the following section.

3 Self-Protecting Evoloop

We apply the proposed concept of self-protection to a simple self-replicating and evolving cellular automaton model, the evoloop [11], invented by the author after Langton's famous self-replicating loop [6]. It is constructed on a nine-state deterministic cellular automata space with von Neumann neighborhoods. Table 1 shows the names and functions of the states used in the evoloop.

An evoloop individual is composed of two basic structures: an inner and outer sheath of square or rectangular shape, and a gene sequence of moving signal states.

1 This is related to the concept of *workplace construction* we recently presented in [13], where we focused particularly on the construction process of self-reproducing automata.

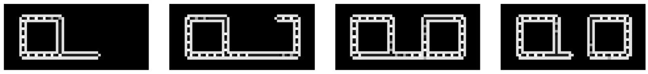


Figure 1. Self-replication of an evoloop.

The gene sequence contains several state 7 genes for straight growth of a construction arm of the loop, and a pair of state 4 genes for left turning of the arm. After three such left turns, the arm collides with itself; this makes the tip and the root of the arm bond together to complete self-replication (Figure 1). Evoloops may mutate through direct interaction (collision) of their sheath structures, leading to a change in the gene sequence of offspring loops.

Note that the reflectional asymmetry inherent in the dynamics of the evoloop world is key to determining whether colliding patterns give rise to mutation or attack. Pointing upward, the left side of an arm is considered as inside, the right side considered as outside. Collisions on the left side therefore cause arm bonding to form a new shape, while collisions on the right side do not. Since all undefined neighborhood situations are set to generate a dissolving state 8 in the original evoloop rules, the loops are very fragile with respect to exogenous attacks.

We introduce a self-protection capability into the evoloop by adding a new state 9 and some patches to the transition rules that mainly describe the behavior of this new state. These patch rules are hand-designed by the author. The transition rules explicitly defined in the original evoloop are all inherited as is, so that the normal self-replicative behavior will not change at all. With this modification applied, the model is granted three essential capabilities—self-replication, self-dissolution, and self-protection. We call this new model the *self-protecting evoloop*, or the *SP evoloop*.

As reviewed in the previous section, there could be many ways to implement self-protection in a system, and so in the SP evoloop in particular. We have so far tested three different mechanisms as listed below:

Shielded: The attacked loop generates a dissolving state 8 at the tip of the attacker's arm. Since the dissolving state usually becomes canceled by the gene flow in the arm, most likely the attacker suffers from only a partial dissolution of the tip of its arm, and the same attack occurs repeatedly.

Deflecting: The attacked loop generates an umbilical cord dissolver 6 at the tip of the attacker's arm. The umbilical cord dissolver goes back to the attacker, removing the entire arm, and then deceives the attacker into believing that self-replication has been completed. The attacker then starts another attempt of self-replication in a different direction rotated by 90 degrees counterclockwise.

Poisoning: The attacked loop generates a poison 9 at the tip of the attacker's arm. The poison works as a kind of dissolving state with extra strength that will never be canceled until it deletes the whole contiguous structure. This may be one of the most radical and merciless mechanisms of self-protection.

Their behavioral difference on collision is shown in Figure 2. See Appendix for details of how to derive a complete set of transition rules for each model.

These protection mechanisms are implemented so as to work properly anywhere on the right side (outside) of an arm or a body of evoloops including T-connections, except for their corners. This incompleteness is left unpatched intentionally, based on the assumption that most attackers hitting corners are *kin* of the attacked. Without such

Table 1. Names and functions of the states in the evoloop. States and functions newly introduced in this letter are shown in boldface.

State	Name	Function
0	Background	Quiescent state.
1	Core	Fills in sheath structure and conducts genes in it.
2	Sheath	Forms sheath structure.
3	Left indicator	Supports left turning of an arm.
	Bonder	Supports bonding of two arms.
	Sprout generator	Supports germinating of a new sprout.
	Sprout capper	Caps a tip of a sprout.
	Sprout finisher	Finishes growth of a sprout.
4	Gene	Keeps genetic information for left turning of an arm and finishes growth of a sprout.
	Sprout guide	Supports growth of a sprout.
5	Messenger	Points where a loop should germinate a new sprout.
6	Umbilical cord dissolver	Dissolves an umbilical cord between parent and offspring.
7	Gene	Keeps genetic information for straight growth of an arm or a sprout.
8	Dissolving state	Dissolves sheath structure.
	Attack detector	**Detects attacks coming from outside (for poisoning SP evoloop).**
9	**Attack detector**	**Detects attacks coming from outside (for shielded and deflecting SP evoloops).**
	Poison	**Dissolves the attacker completely (for poisoning SP evoloop).**

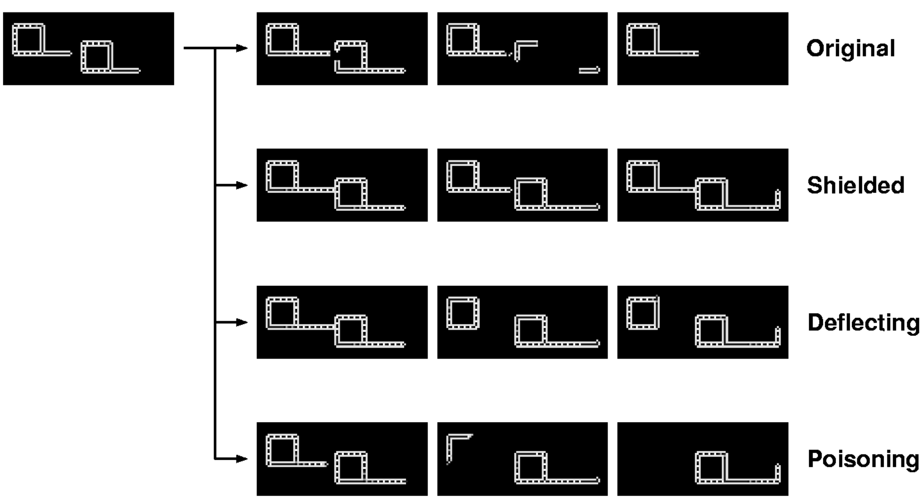


Figure 2. Original behavior and three different versions of self-protection mechanisms implemented on the evoloop. Original: The attacked loop dissolves, and the attacker continues to produce its offspring where the attacked loop was. Shielded: The tip of the attacker's arm dissolves while the attacked receives no damage, so the same attack continues to occur repeatedly. Deflecting: The attacked loop generates an umbilical cord dissolver 6 at the tip of the attacker's arm. The attacker is deceived and begins to produce its offspring in a different direction rotated by 90 degrees counterclockwise. Poisoning: The attacked loop generates a poison 9 at the tip of the attacker's arm to kill the attacker completely.

incompleteness of self-protection, the entire system would be more prone to fall into limit cycles with no generation turnover. This issue will be revisited in the last section.

4 Results

We have conducted simulations of evolution of the proposed SP evoloops, aiming at the evaluation of how much diversity increase the introduction of self-protection brings to the system. The cellular automata space used for simulations consists of 400 × 400 sites with periodic boundary conditions applied to the edges. Simulations start from a quiescent initial configuration with a single ancestor loop of size 13 that has a genome 77447777777777 (two state 7 genes preceding a pair of state 4 genes). Evolutionary dynamics is traced over 1,000,000 updates for each run, where square loop structures in the space are identified and counted according to their size using the method proposed in [11].

To characterize the diversity within the population, we define a *trait distribution entropy*

$$H = -\sum_i \frac{n_i}{N} \log \frac{n_i}{N} = \log N - \frac{1}{N} \sum_i n_i \log n_i,$$

where n_i is a population of loops of size i, and $N = \sum_i n_i$ (total population). This entropy is zero when the system is filled with loops of the same size only, and takes its maximal value $\log N$ when every loop in the system differs from each other (i.e., $n_i = 0$ or 1 for all i). This characterization takes into account the size difference only. We have reported elsewhere [1, 10] the emergence of huge genetic and behavioral diversity in the evoloop world using a more sophisticated identification scheme, which is not discussed here in detail.

The results of simulation runs are presented in Figures 3, 4, 5, and 6 for original, shielded, deflecting, and poisoning SP evoloops, respectively. Each figure shows time evolution of populations, average and dominant traits, and trait distribution entropy in separate charts. While the basic direction of evolutionary paths leading to the dominance by the smallest species (size 4) does not change for any of the four cases, variations can be seen in the sustainability of larger loops (e.g., size 5). To confirm this observation, the difference in average trait distribution entropy between the four cases is summarized in Figure 7, showing the distribution of entropy values sampled at regular intervals between 200,000 and 1,000,000 updates for each run. A diversity increase is clearly seen for the deflecting and poisoning cases; actually all three self-protection mechanisms show a statistically significant increase of entropy compared to the baseline (original), since the number of sample points is large (20,000 points).

To check the robustness of the observed diversity increase, we have also conducted the same simulations with variations added to the initial configuration, the transition rules, or both. For a variation of the initial configuration, we choose another type of size 13 loop with a different genome, 77777777777744. While this loop is also self-replicative, its evolvability is significantly smaller than that of the loop with genome 77447777777777 [11], making the final population composition significantly different. For a variation of transition rules, we additionally introduce a small set of rules listed in Table 7 in the Appendix. This tiny patch enables state 4 genes to trigger arm bonding at the final stage of self-replication, which was originally possible only by state 7 genes. This modification slightly improves the adaptability of loops but does not change their behavior fundamentally.

The results are shown in Figure 8. Note that the magnitudes of the average and standard deviation of the entropy are larger in cases with the different ancestral loop.

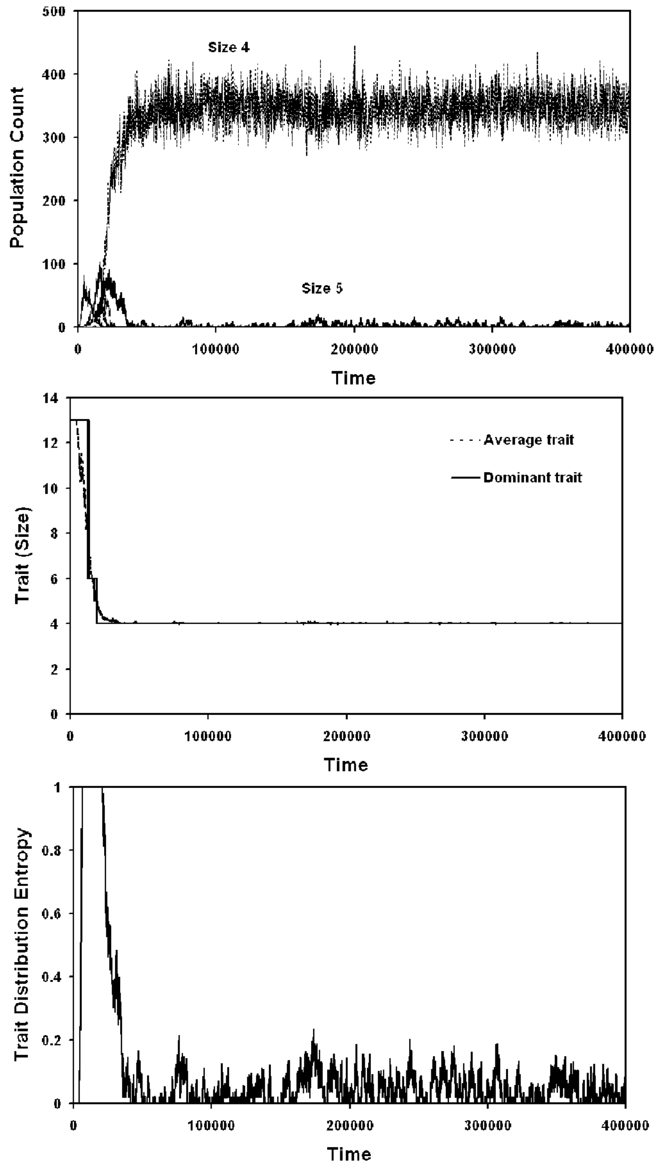


Figure 3. Simulation result of the original evoloop. Ancestor: size 13 evoloop with genome 774477777777777. Space: 400 × 400 sites with periodic boundary conditions.

Since this species does not evolve to smaller size, the population remains dominated by large loops (e.g., sizes 13, 14, 15). Thus there is room left for a variety of non-self-replicative smaller variants to emerge, increasing the trait distribution entropy values.

These plots tell that the effect of poisoning is rather sensitive to the variation in initial condition (and thus in population composition). This observation implies that the significant increase of diversity for poisoning seen in Figure 7 may be an artifact specific to model settings, and that the effectiveness of self-protection may depend on the mechanism chosen for its implementation. On the other hand, the diversity increase in the deflecting and shielded cases seems fairly robust to model variations;

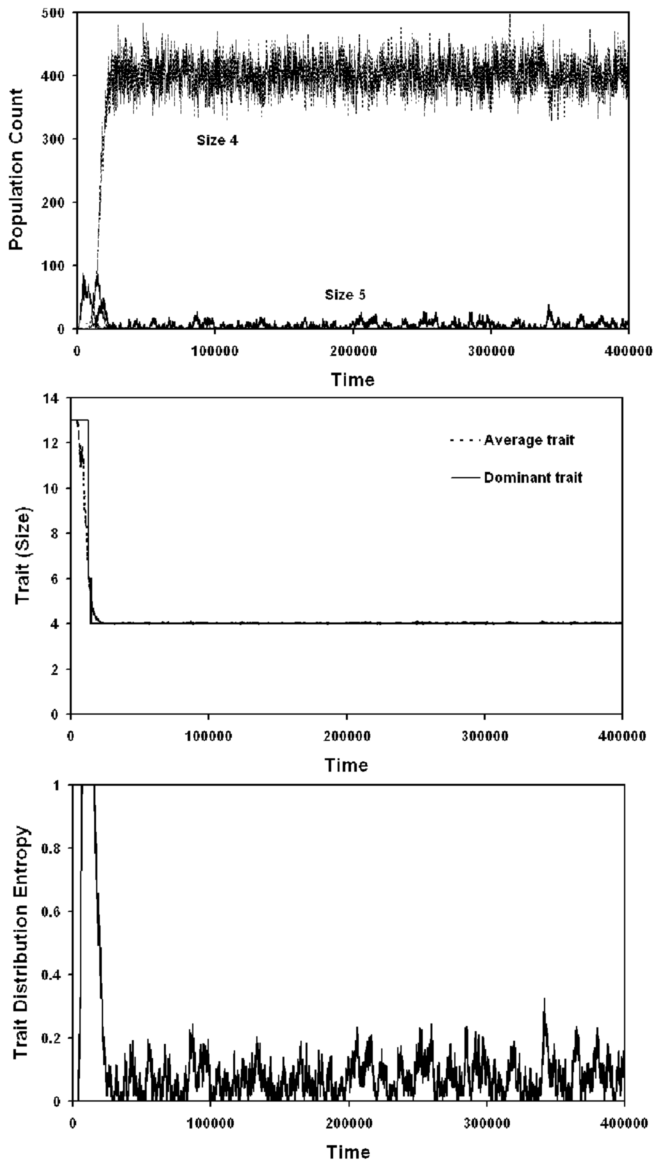


Figure 4. Simulation result of the shielded SP evoloop. Ancestor: size 13 evoloop with genome 7744777777777777. Space: 400 $\times$ 400 sites with periodic boundary conditions.

the deflecting mechanism in particular displays significant, consistent diversity increase. An interesting finding is that neither of these robust mechanisms ever kills the attackers, which is good for diversity maintenance in the entire population. In contrast, too strong self-protection like that implemented in poisoning promotes dominance by a single type, causing a negative effect on diversity.

The difference in performance between the deflecting and shielded cases can be explained as follows: The deflecting mechanism diverts attackers to a different place, where they may be able to find room to produce their offspring, while the shielded mechanism simply blocks the attackers attempt at self-replication so they will remain

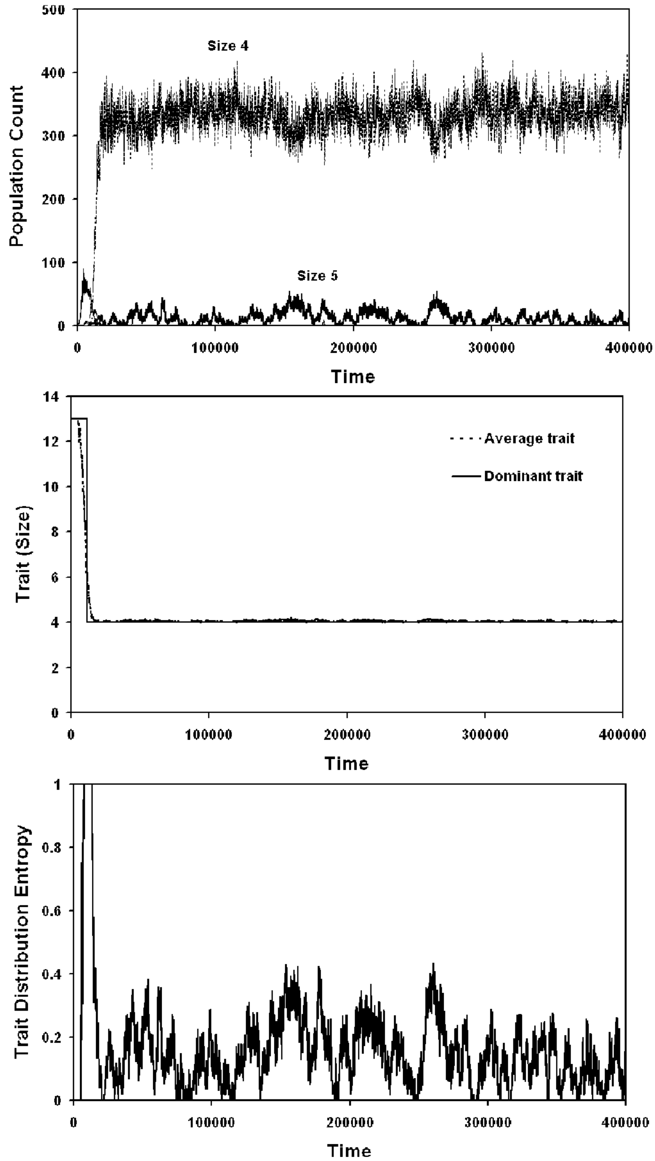


Figure 5. Simulation result of the deflecting SP evoloop. Ancestor: size 13 evoloop with genome 774477777777777. Space: 400 × 400 sites with periodic boundary conditions.

unable to produce offspring unless some external factor changes the situation. Such a simple blocking strategy generally results in a greater likelihood for a system to fall into a confined region of phase space (limit cycle).

5 Discussion

We have introduced a self-protection capability into a simple artificial evolutionary system, the evoloop, to make their population more diverse. Simulation results have generally indicated a positive effect of the self-protection mechanisms on diversity cre-

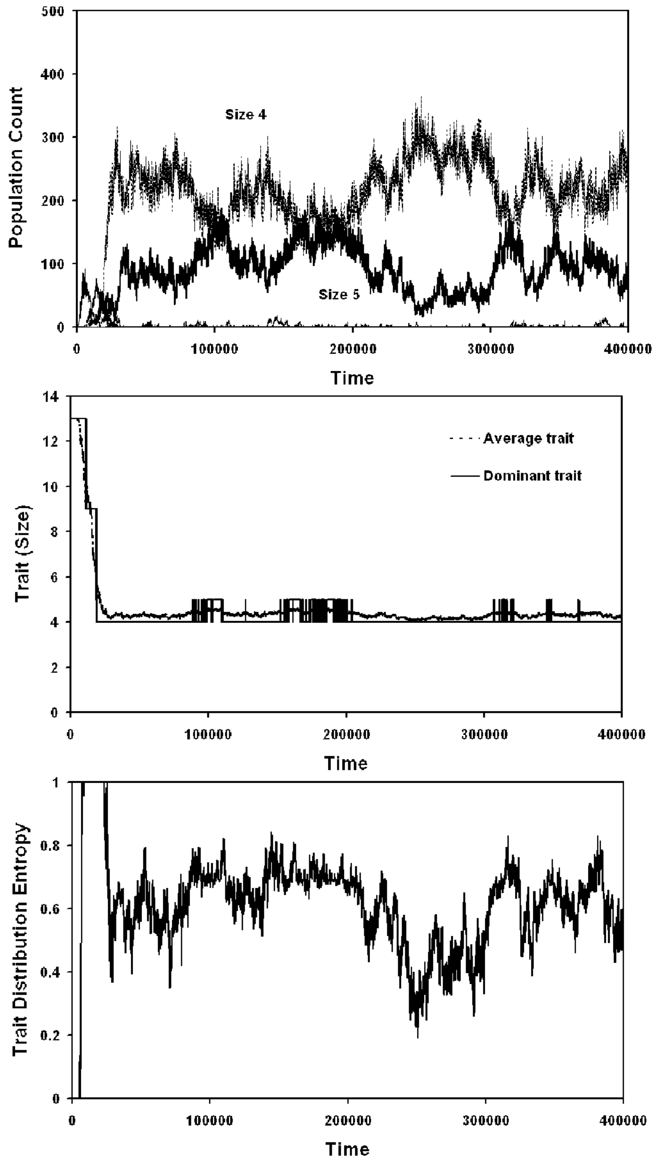


Figure 6. Simulation result of the poisoning SP evoloop. Ancestor: size 13 evoloop with genome 774477777777777. Space: 400 × 400 sites with periodic boundary conditions.

ation and maintenance in the evoloop world. In the meantime, however, it is also suggested that the effectiveness of self-protection may depend on the specific implementation of its mechanisms. Among the mechanisms we have tested so far (shielded, deflecting, and poisoning), the deflecting one seems most robust in its effect on the diversity increase.

We note that implementing self-protection is generally much easier than introducing explicit interspecific interactions if systems are made up in a bottom-up way like cellular automata. For example, the deflecting mechanism has been introduced by adding just one new state and 73 related transition rules (see Tables 3 and 5 in the Appendix). This is mainly due to the fact that the self-protection can be achieved with local efforts

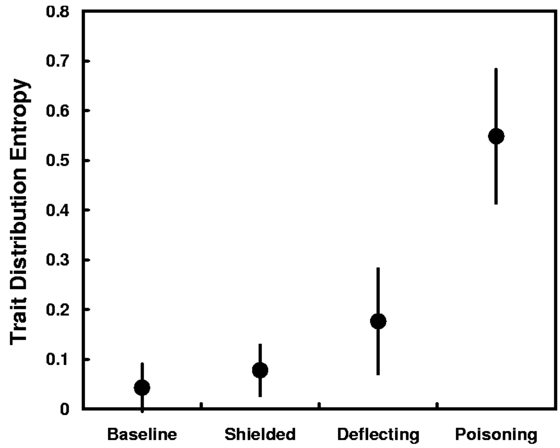


Figure 7. Comparison of trait distribution entropy between the original and three SP evoloops shown in Figures 3 to 6. The original evoloop case is shown as "baseline." Data sampled at 40 updates interval during 200,000 to 1,000,000 updates for each run. Filled circles represent average values, with the solid line showing corresponding standard deviations.

only, that is, everything in the protection process happens within a local region at the interface between organisms and external environment. If complex interspecific interactions among artificial organisms were to be embedded in such distributed models, far more extra states and rules would be needed to hierarchically construct higher-order functions out of local behaviors. We thus expect that the concept of self-protection, still quite abstract and theoretical, may give a general and practical solution to the problem of diversity increase in the design of artificial evolutionary systems. This could be of particular relevance in the context of physical implementation of artificial self-replicator models, which is now occurring in evolvable hardware [15].

Finally, we refer to a study recently and independently conducted by Suzuki [16], which is of direct relevance to the subject we have discussed. In his work, Suzuki systematically classified possible modes of interaction between self-replicators into five (inroad, offensive, cancel, defensive, and counter), and examined the effect of each on the evolutionary dynamics of self-replicating cellular automaton models. Our original, deflecting, and poisoning mechanisms correspond to the offensive, defensive, and counter modes in Suzuki's classification, respectively. There is no counterpart of our shielded mechanism in his classification.

One difference between Suzuki's and our models is that Suzuki's defensive mode makes replicators too stable and thus the system tends to fall into rather static patterns, which is not the case with our deflecting SP evoloop. The key to understanding this difference is the incompleteness of self-protection intentionally left in our model construction. Namely, our self-protection mechanisms cannot deal with attacks that hit corners of a loop. This weak point makes it possible for generation turnover to last. We actually have tested other models with *perfect* self-protection that covers corners as well, and find that the system always loses evolutionary dynamics due to the same reason as mentioned above. This, together with the diversity loss seen for too strong self-protection as in our poisoning model, leads us to the observation that a balance between stability and instability of organisms bears key importance for the evolvability and sustainability of systems. The issue of such balance may deserve more attention in the design of artificial evolutionary systems in the future.

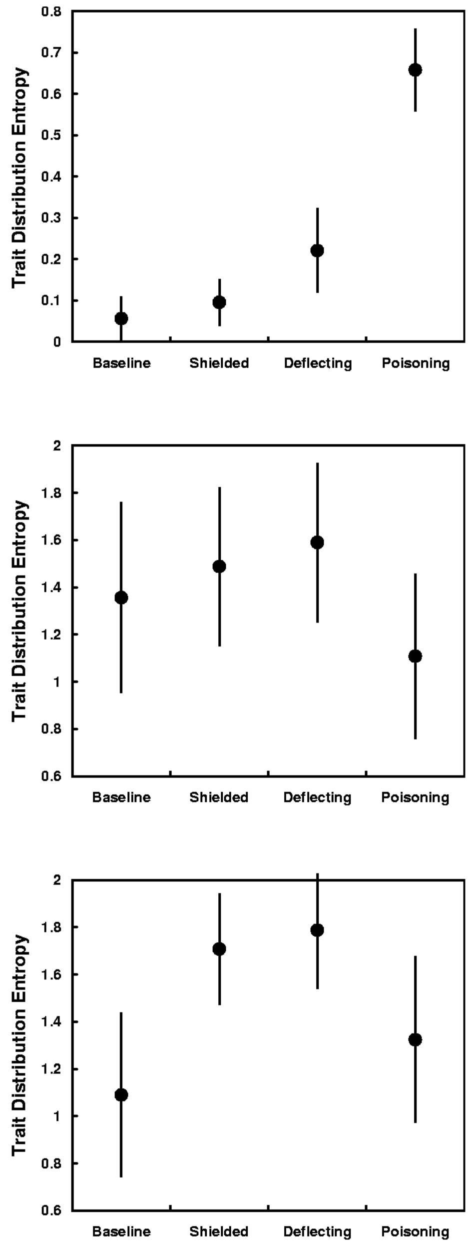


Figure 8. Comparison of trait distribution entropy conducted in the same way as in Figure 7 under three other conditions. Top: Additional transition rules (Table 7) added. Middle: Ancestor genome changed to 7777777777777744. Bottom: Additional transition rules added and ancestor genome changed. The sampling period is shortened to 200,000 to 400,000 updates for shielded evoloops in the last setting, because it fell in a limit cycle after 450,000 updates.

Acknowledgments

This work is supported in part by a grant from Hayao Nakayama Foundation for Science, Technology, and Culture.

References

1. Antony, A., Salzberg, C., & Sayama, H. A closer look at the evolutionary dynamics of self-reproducing cellular automata. Accepted pending revisions.
2. Bagnoli, F., & Bezzi., M. (1997). Speciation as pattern formation by competition in a smooth fitness landscape. *Physical Review Letters, 79*, 3302–3305.
3. Chou, H. H., & Reggia, J. A. (1997). Emergence of self-replicating structures in a cellular automata space. *Physica D, 110*, 252–276.
4. Drossel, B., & McKane, A. (2000). Competitive speciation in quantitative genetic models. *Journal of Theoretical Biology, 204*, 467–478.
5. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Reading, MA: Addison-Wesley.
6. Langton, C. G. (1984). Self-reproduction in cellular automata. *Physica D, 10*, 135–144.
7. Lohn, J. D., & Reggia, J. A. (1997). Automatic discovery of self-replicating structures in cellular automata. *IEEE Transactions on Evolutionary Computation, 1*, 165–178.
8. Margules, C. R., & Pressey, R. L. (2000). Systematic conservation planning. *Nature, 405*, 243–253.
9. Maynard Smith, J., & Szathmáry, E. (1995). *The major transitions in evolution*. New York: Oxford University Press.
10. Salzberg, C., Antony, A., & Sayama, H. (2003). Visualizing evolutionary dynamics of self-replicators using graph-based genealogy. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, & J. Ziegler (Eds.), *Advances in artificial life: Proceedings of the Seventh European Conference on Artificial Life (ECAL 2003), Lecture Notes in Artificial Intelligence 2801* (pp. 387–394). Berlin: Springer-Verlag.
11. Sayama, H. (1999). A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. *Artificial Life, 5*, 343–365.
12. Sayama, H. (2000). Self-replicating worms that increase structural complexity through gene transmission. In M. A. Bedau, J. S. McCaskill, N. H. Packard, & S. Rasmussen (Eds.), *Artificial life VII: Proceedings of the Seventh International Conference on Artificial Life* (pp. 21–30). Cambridge, MA: MIT Press.
13. Sayama, H. (2002). von Neumann's machine in the shell: Enhancing the robustness of self-replication processes. In R. K. Standish, M. A. Bedau, & H. A. Abbass (Eds.), *Artificial life VIII: Proceedings of the Eighth International Conference on Artificial Life* (pp. 49–52). Cambridge, MA: MIT Press.
14. Standish, R. K. (2002). Diversity evolution. In R. K. Standish, M. A. Bedau, & H. A. Abbass (Eds.), *Artificial life VIII: Proceedings of the Eighth International Conference on Artificial Life* (pp. 131–137). Cambridge, MA: MIT Press.
15. Stauffer, A., & Sipper, M. (2002). An interactive self-replicator implemented in hardware. *Artificial Life, 8*, 175–183.
16. Suzuki, K. (2003). *Interaction and dynamics of self-replicating CA* (in Japanese). Master's thesis, Graduate School of Arts and Sciences, University of Tokyo.
17. von Neumann, J. (1951). The general and logical theory of automata. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior—The Hixon symposium* (pp. 1–41). New York: Wiley.
18. von Neumann, J. (1966). *Theory of self-reproducing automata*. Urbana, IL: University of Illinois Press. Edited and completed by A. W. Burks.
19. Wilke, C. O., & Adami, C. (2002). The biology of digital organisms. *Trends in Ecology and Evolution, 17*, 528–532.

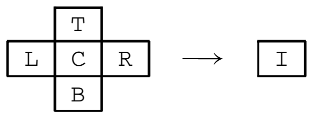
20. Yedid, G., & Bell, G. (2002). Macroevolution simulated with autonomously replicating computer programs. *Nature, 420*, 810–812.

Appendix

A.1 Transition Rules of the Original Evoloop

The transition rules of the original evoloop can be obtained by the operations below:

1. Define the rules listed in Table 2 and the rotationally symmetric rules. Each situation `CTRBL` and its image `I` listed in the tables is read as follows:



2. Let `8`→`0` with no condition.
3. To all the undefined situations in whose four neighbors (`TRBL`) there is at least one site in state `8`, apply the following:
   (a) Let `0,1`→`8` if there is at least one site in state `2,3,...,7` in its four neighbors (`TRBL`), otherwise let `0`→`0` and `1`→`1`.
   (b) Let `2,3,5`→`0`.
   (c) Let `4,6,7`→`1`.
4. Clear up all the undefined situations by letting `0`→`0` and `1,2,...,7`→`8`.

A.2 Transition Rules of the SP Evoloops

The transition rules of the proposed three SP evoloops can be obtained by the operations below.

A.2.1 Shielded and Deflecting SP Evoloops

1. Define the rules listed in Tables 2 and 3 and the rotationally symmetric rules.
2. Define the rules listed in either Table 4 for the shielded SP evoloop, or Table 5 for the deflecting SP evoloop, and the rotationally symmetric rules.
3. Let `8`→`0` with no condition.
4. To all the undefined situations in whose four neighbors (`TRBL`) there is at least one site in state `8`, apply the following:
   (a) Let `0,1`→`8` if there is at least one site in state `2,3,...,7` in its four neighbors (`TRBL`); otherwise let `0`→`0` and `1`→`1`.
   (b) Let `2,3,5,9,`→`0`.
   (c) Let `4,6,7`→`1`.
5. Clear up all the undefined situations by letting `0`→`0` and `1,2,...,7,9` →`8`.

A.2.2 Poisoning SP Evoloop

1. Define the rules listed in Tables 2, 3, and 6, and their rotationally symmetric ones.
2. Let `8`→`0` with no condition.

Table 2. Principal part of the transition rules of the original evoloop.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00001->2	10202->1	11272->7	20172->2	21322->2	40125->0
00004->3	10211->1	11273->5	20202->2	21422->2	40162->0
00012->2	10212->1	11322->1	20203->2	21622->2	40212->0
00015->2	10213->1	11332->1	20205->2	21722->2	40215->0
00021->2	10221->1	11542->4	20206->5	22224->2	40222->1
00024->2	10224->4	11572->7	20207->3	22227->2	40232->1
00042->2	10227->7	11624->4	20212->2	22234->2	40262->6
00045->2	10232->4	11627->7	20215->2	22237->2	40312->0
00075->2	10241->4	12224->4	20221->2	22243->2	40322->1
00102->2	10242->4	12227->7	20222->2	22244->2	50002->5
00214->1	10243->4	12243->4	20223->2	22273->2	50012->5
00217->1	10251->1	12273->7	20232->3	22277->2	50021->5
00232->2	10252->7	12324->4	20242->2	22324->3	50023->2
01122->1	10254->3	12327->7	20245->2	22327->3	50024->5
01212->1	10257->7	12426->6	20252->5	30001->3	50027->5
01232->1	10271->7	12433->3	20262->0	30002->2	50042->5
01242->1	10272->7	12627->6	20265->0	30003->2	50072->5
01245->1	10273->5	20001->2	20272->2	30004->3	50202->2
01252->6	10512->1	20002->2	20275->2	30007->4	50205->2
01262->6	10542->4	20004->2	20312->2	30012->3	50212->5
01272->1	10572->7	20005->2	20322->2	30032->2	50215->2
01275->1	10621->1	20006->0	20342->2	30042->1	50242->5
01342->1	10624->4	20007->1	20345->2	30102->1	50272->5
01372->1	10627->7	20012->2	20372->2	30125->0	50312->0
01422->1	11112->1	20015->2	20412->2	30212->3	60202->2
01425->1	11122->1	20021->2	20422->2	30242->3	60212->2
01432->1	11124->4	20022->2	20442->2	30252->1	60222->0
01435->1	11125->1	20023->2	20512->2	30272->3	60242->2
01442->1	11127->7	20024->2	20542->5	30332->1	60272->2
01462->1	11162->1	20026->0	20572->5	31212->3	61222->0
01722->1	11212->1	20027->2	20612->5	31242->3	62224->0
01725->1	11213->1	20032->4	20621->2	31252->1	62227->0
01756->1	11215->1	20042->3	20642->5	31272->3	70102->0
01762->1	11222->1	20045->2	20672->5	32424->3	70112->0
01772->1	11224->4	20054->5	20712->2	32425->1	70122->0
10001->1	11227->7	20057->5	20722->2	32427->3	70125->0
10012->1	11232->1	20062->0	20772->2	32527->1	70162->0
10021->1	11242->4	20072->2	21122->2	32727->3	70212->0
10024->4	11243->4	20075->2	21222->2	40000->1	70215->0
10027->7	11252->7	20102->2	21223->2	40002->1	70222->1
10121->1	11254->3	20112->2	21224->2	40102->0	70232->0
10124->4	11257->7	20122->2	21227->2	40112->0	70262->6
10127->7	11262->6	20142->2	21232->3	40122->0	70312->0

Table 3. Additional transition rules common to the proposed three SP evoloops.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
01243->1	10031->1	11113->1	11372->7	20237->2	40113->0
01273->1	10034->4	11134->4	20132->2	20432->2	40213->0
01443->1	10312->1	11137->7	20231->2	20732->2	70113->0
01773->1	10342->4	11342->4	20234->2	30202->3	70213->0

Table 4. Additional transition rules specific to the shielded SP evoloop.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00208->0	10094->4	11219->1	20492->2	32429->9	90272->9
01249->1	10097->7	11942->4	20792->2	32729->9	90282->2
01279->1	10229->7	11972->7	20922->2	40119->0	91212->9
01292->1	10292->7	20029->2	21229->2	40219->0	91242->9
01449->1	10372->9	20092->2	21922->2	70119->0	91272->9
01779->1	10912->1	20192->2	22249->2	70219->0	91282->2
01792->1	10942->4	20229->2	22279->2	90113->0	92424->9
01922->1	10972->7	20291->2	22294->2	90202->9	92427->9
10029->7	11119->1	20292->2	22297->2	90212->9	92428->2
10037->9	11194->4	20294->2	30292->9	90213->0	92727->9
10091->1	11197->7	20297->2	31292->9	90242->9	92728->2

Table 5. Additional transition rules specific to the deflecting SP evoloop.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
01246->1	10061->1	11116->1	20264->2	22264->2	70116->0
01276->1	10064->4	11164->4	20267->2	22267->2	70216->0
01292->1	10067->7	11167->7	20292->2	22276->2	90113->0
01446->1	10229->7	11216->1	20462->2	30292->6	90213->0
01776->1	10292->7	11642->4	20762->2	31292->6	
01792->1	10372->9	11672->7	20922->2	32429->6	
01922->1	10612->1	20162->2	21229->2	32729->6	
10029->7	10642->4	20226->2	21262->2	40116->0	
10037->9	10672->7	20261->2	22246->2	40216->0	

Table 6. Additional transition rules specific to the poisoning SP evoloop.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
01279->1	01822->1	10372->8	20822->2	30282->9	90212->2
01282->1	10028->7	20192->2	21228->2	31282->9	90242->2
01292->9	10037->8	20282->2	21292->2	32428->9	90272->2
01779->1	10228->7	20292->0	21922->2	32728->9	
01782->1	10282->7	20297->2	22297->2	90202->2	

Table 7. Additional transition rules used to check the robustness of results against model variations. These rules enable state 4 genes to trigger arm bonding as well as state 7 genes.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00233->3	20133->2	20332->2	20733->2	22433->2	30213->2
20033->2	20204->3	20433->2	21332->2	22733->2	30312->2

3. Let `9`→`0` for all undefined situations whose `C` is `9`.
4. To all the undefined situations in whose four neighbors (`TRBL`) there is at least one site in state `9`, apply the following:
   (a) Let `0`→`0`.
   (b) Let `1`,`2`,. . . ,`8`→`9`.
5. To all the situations (still remaining undefined after the above) in whose four neighbors (`TRBL`) there is at least one site in state `8`, apply the following:
   (a) Let `0`,`1`→`8` if there is at least one site in state `2`,`3`,. . . ,`7` in its four neighbors (`TRBL`); otherwise let `0`→`0` and `1`→`1`.
   (b) Let `2`,`3`,`5`→`0`.
   (c) Let `4`,`6`,`7`→`1`.
6. Clear up all the undefined situations by letting `0`→`0` and `1`,`2`,. . . ,`7`→`8`.