MSDM5003 (Fall 21/22) Final Project

Adaptive Minority Game

Cheng Wing Kit

1. Introduction

In real world, there are many occasions that being the minority is more favorable than being the majority. For example, in the stock market, when there are more buyers than sellers, the price will rise and the sellers take advantage. Conversely, the price will drop when there are more sellers than buyers which favors the buyers. Minority Game is a simple model used to mimic such characteristics. Different agents made their own decisions with the aim of being one in the minority group so that they could benefit from the system. In this project, we are going to study some "adaptive" strategies that may help one to perform better in the Minority Game.

2. Theory

The setting of Minority Game is described in Appendix 1. M. Sysi-Aho and others [1] presented an adaptive version of the Minority Game which allows an agent to "adapt" to the environment by periodically changing his strategies if he is among the worst performing agents. The period and the worst performing fraction are denoted by $\boldsymbol{\tau}$ and $\mathbf{n}$ respectively. They introduced the one-point genetic crossover scheme for making changes to the strategies, which is described in Appendix 2. In addition, we have proposed the mutation scheme and the combination of crossover and mutation scheme which are described in Appendix 3 and 4. Six crossover schemes, two mutation schemes and two crossover-mutation schemes summarized in the table below will be studied.

<table>
<tr><td colspan="3">Crossover</td><td colspan="4">Mutation</td></tr>
<tr><td>Scheme</td><td>Parents</td><td>Offspring</td><td>Scheme</td><td colspan="2">Target</td><td>Mode</td></tr>
<tr><td>A</td><td>Random 2</td><td>Same</td><td>G</td><td colspan="2">Worst 2</td><td>Tail</td></tr>
<tr><td>B</td><td>Random 2</td><td>Random 2</td><td>H</td><td colspan="2">Worst 2</td><td>Long</td></tr>
<tr><td>C</td><td>Random 2</td><td>Worst 2</td><td colspan="4">Crossover-mutation</td></tr>
<tr><td>D</td><td>Best 2</td><td>Same</td><td>Scheme</td><td>Parent</td><td>Target</td><td>Mode</td></tr>
<tr><td>E</td><td>Best 2</td><td>Random 2</td><td>I</td><td>Best</td><td>Worst 2</td><td>Long</td></tr>
<tr><td>F</td><td>Best 2</td><td>Worst 2</td><td>J</td><td>Best</td><td>Worst 2</td><td>Short</td></tr>
</table>

In this project, we will focus on (1) the collective behavior, (2) the revenging ability and (3) the relative performance of different adaptive schemes.

3. Results

3.1. Collective behavior

To study the collective behavior of the crossover scheme, M. Sysi-Aho and others introduced the measure of "Scaled utility of the system $U(x_t)$" given in Appendix 5. In the simulation for each of the adaptive schemes, we assume all agents use the same scheme and obtain $U(x_t)$ through game played with parameters **N**=501, **m**=4, **S**=10, **T**=10000, **τ**=40, **n**=0.3. Results are plotted in Fig. 1 which show that comparing with the basic Minority Game, the fluctuation of $U(x_t)$ in all adaptive schemes are reduced in long term. For Scheme **E** and **F**, the fluctuation has even disappeared. Fluctuation reduced by crossover schemes are higher than that of mutation schemes and crossover-mutation schemes. In other words, system in which all agents applying crossover schemes are more efficient than that applying mutation or crossover-mutation schemes.

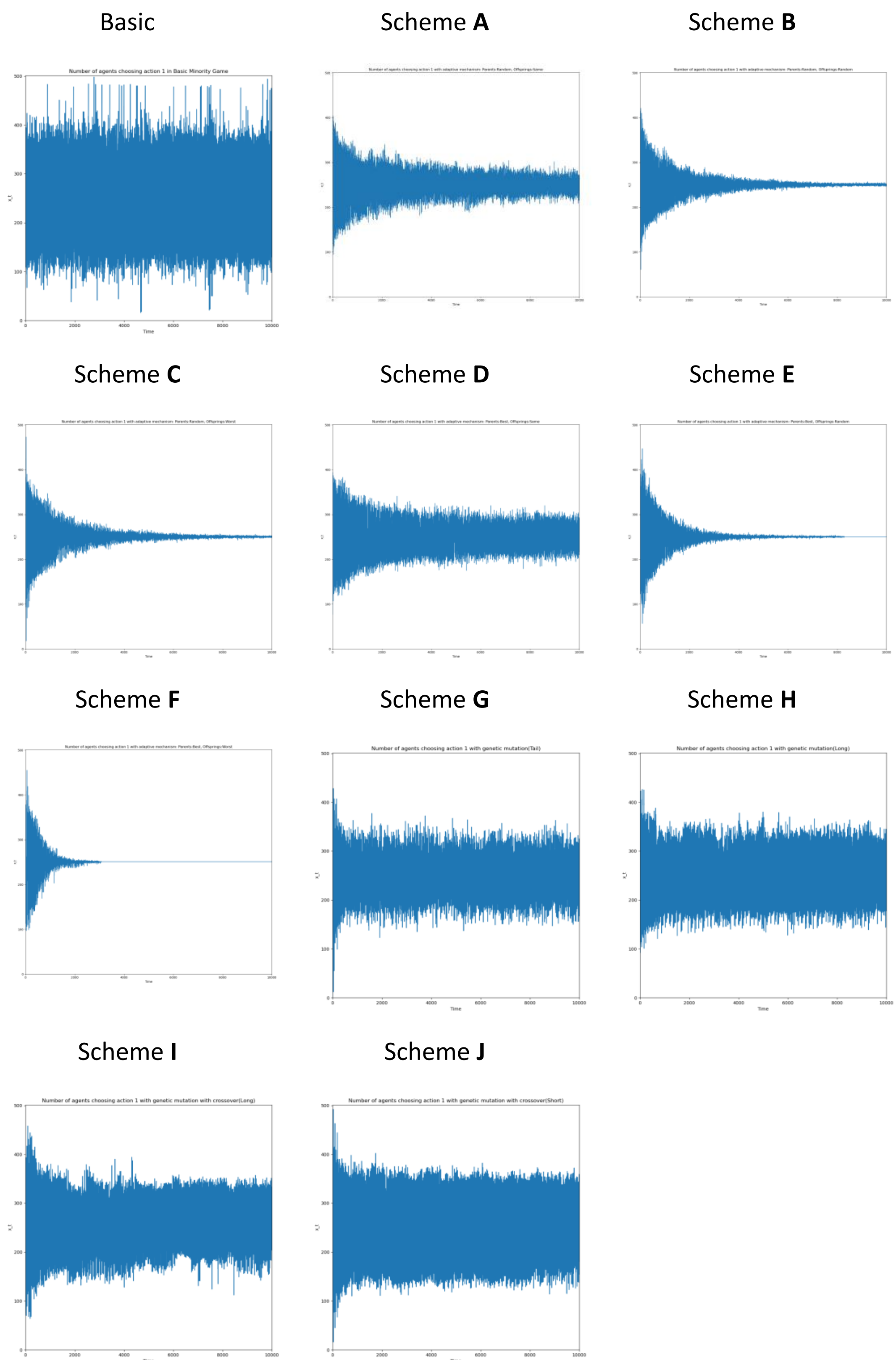


Fig. 1 – Plots of x_t over time of different scheme

In Fig.2, we have plotted the scaled utility of the average over 30 runs and each point in the curves represents a time average taken over a bin of length 50 time-steps. From the curves, scheme **F** is the most efficient among all schemes proposed.

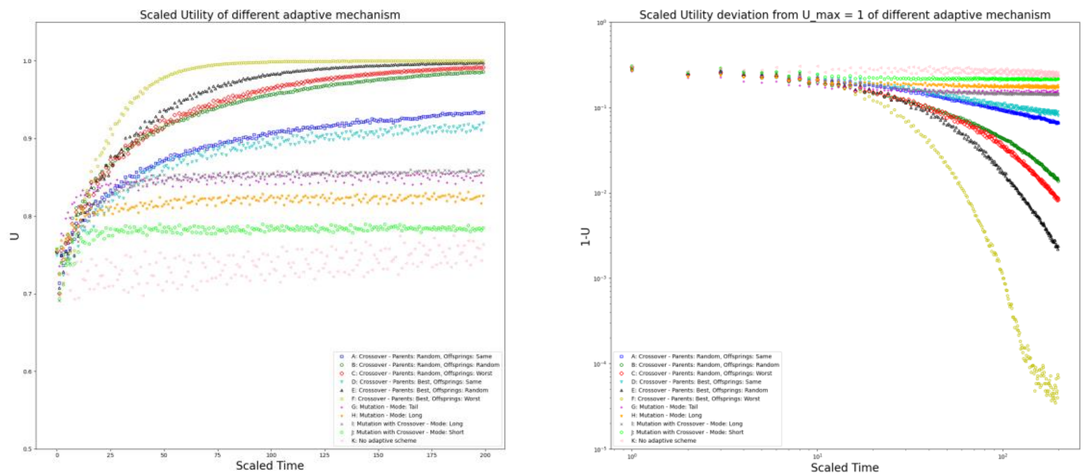
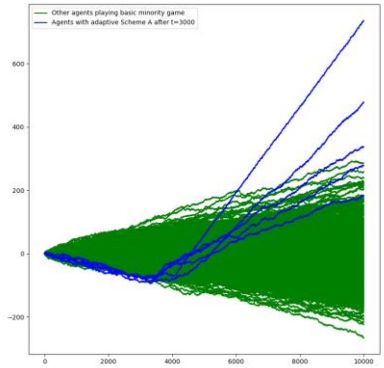


Fig. 2 – (left) $\mathrm{U}(\mathrm{x}_t)$ vs scaled time, (right) $\log(1 - \mathrm{U}(\mathrm{x}_t))$ vs log(scaled time)

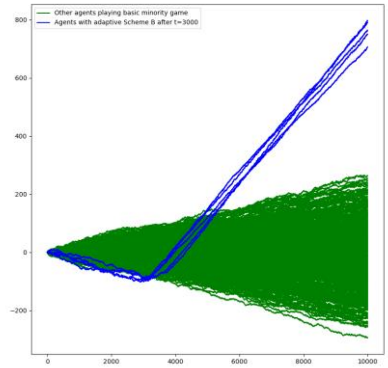
3.2. Revenge ability

In this section, we will look at the revenge ability of different schemes. Revenge means turning from lose to win. Separate simulations were performed with **N**=501, **m**=4, **S**=10, **T**=10000, **τ**=40 and **n**=0.3 for each scheme in a way that the adaption scheme only applied to the worst five agents at **t**=3000 while other agents remain playing the basic minority game. Results are given in Fig. 3 which shows that all schemes are able to turn an agent from losing to winning as the slopes of the performance become positive after applying the adaptive schemes. However, the revenge power are different for different schemes.

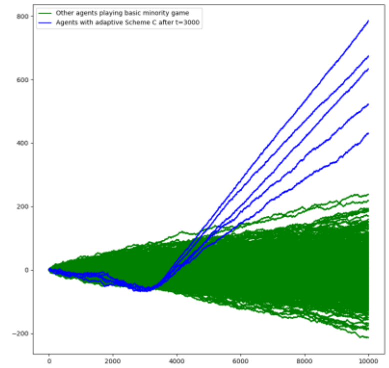
Scheme **A**



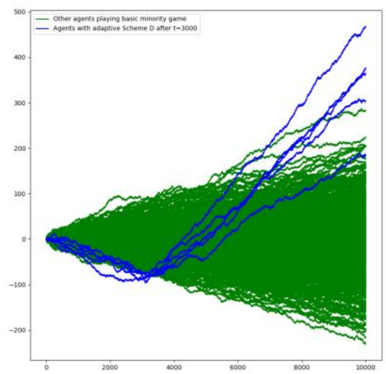
Scheme **B**



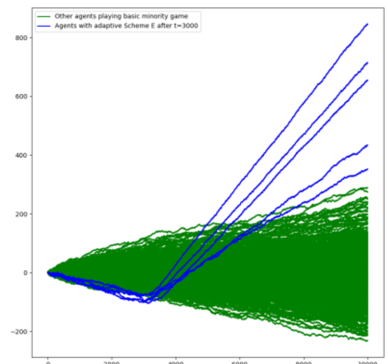
Scheme **C**



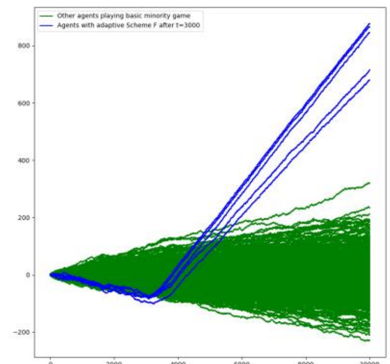
Scheme **D**



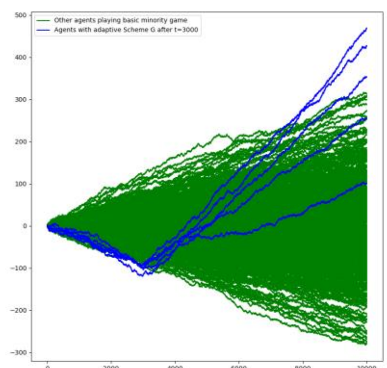
Scheme **E**



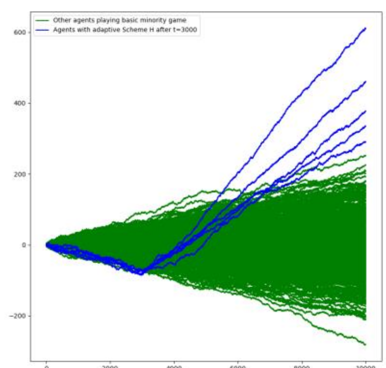
Scheme **F**



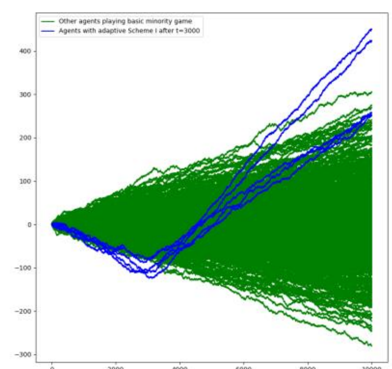
Scheme **G**



Scheme **H**



Scheme **I**



Scheme **J**

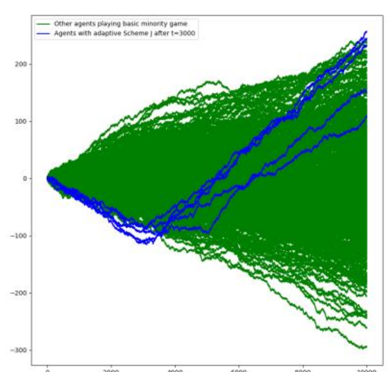


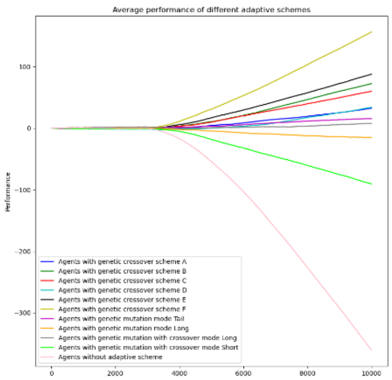
Fig. 3 – Revenge ability plot

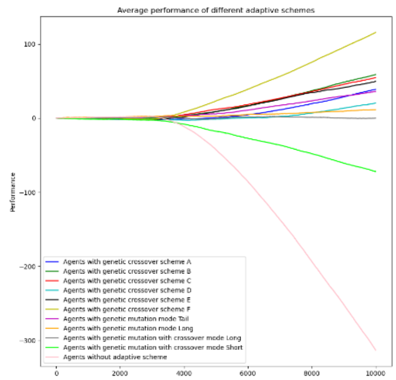
3.3. Relative performance

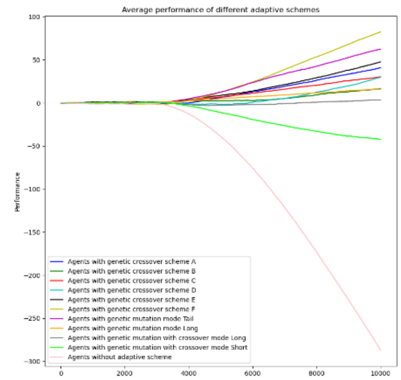
Now, we have 10 adaptive schemes that have revenge ability. But which one is best for an individual agent playing the game? In 3.1, we see that Scheme **F** provides the highest utility for the whole system. In 3.2, the revenge power of Scheme **F** is good in the sense that even the worst of the five revenging agents still performs better than the best of others didn't use any adaptive scheme. It seems Scheme **F** is the answer. However, we have to be careful that all simulations in 3.1 and 3.2 are based on the same parametric setting: **N**=501, **m**=4, **S**=10, **T**=10000, **τ**=40 and **n**=0.3. The result will probably be different if different set of parameters is used. Analyzing all the combinations are outside the scope of this project. Instead, we are going to determine which scheme is the best under the game that (1) all agents have the same memory size **m**=4, (2) all agents have **S**=10 strategies on hand, and (3) the game last for **T**=10000 iterations. In our game setting, we use $\mathbf{N} = 49 \times 11 = 539$ as the total number of agents so that each group (10 adaptive groups + 1 basic group) would have 49 agents. Since the performance of different schemes may depend on the choice of **n** and **τ**, we will simulate the game with different combinations of $\mathbf{n} \in \{0.1,0.2,0.3,0.4,0.5,0.6\}$ and $\boldsymbol{\tau} \in \{10,20,30,40,50,60\}$. All agents will play the basic minority game before they could apply the respective adaptive scheme after **t**=3000. Each **n- τ** combination of game will be played for 30 times and the average scaled performance of each group will be plotted for comparison. Results are shown in Fig.4 which indicated that most of the time Scheme **F** perform best. Scheme **G** performs best when **n**=0.1 and **τ**=40 or 60.

$\tau = 10$ | $\tau = 20$ | $\tau = 30$

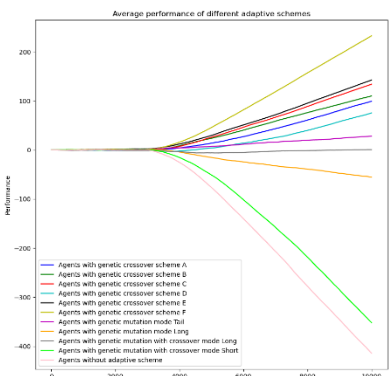
$n = 0.1$

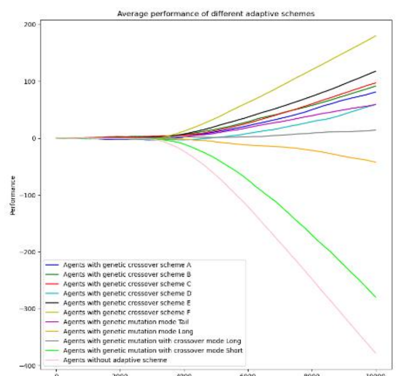


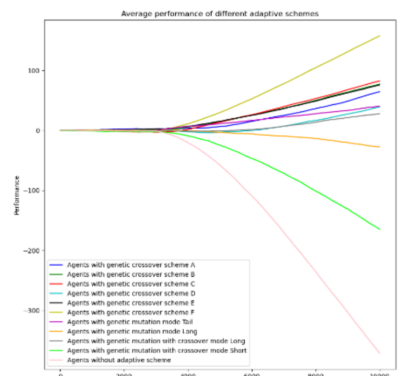




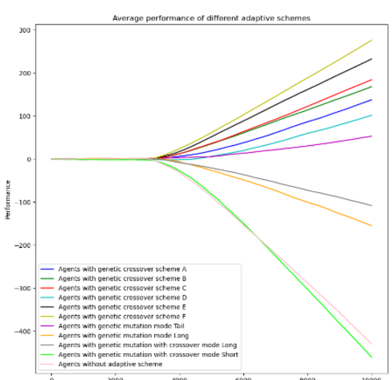
$n = 0.2$

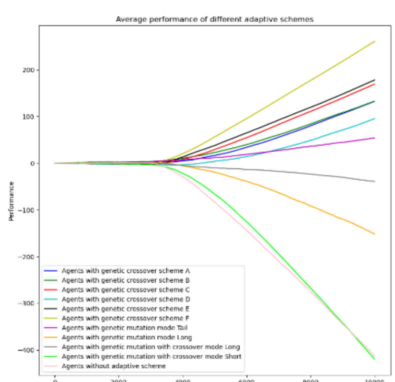


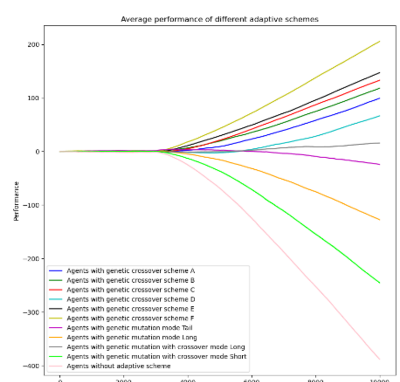




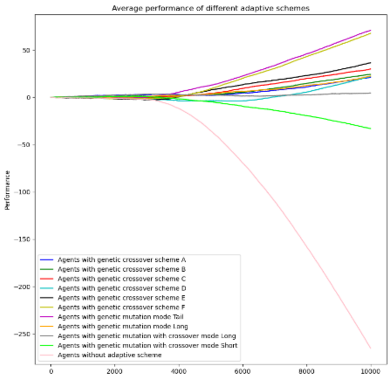
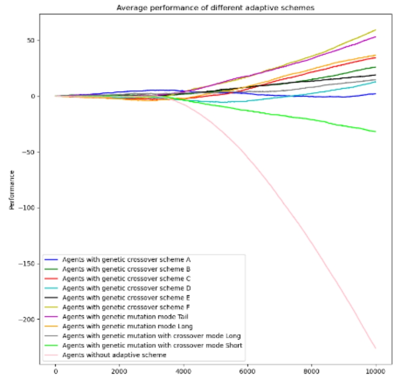
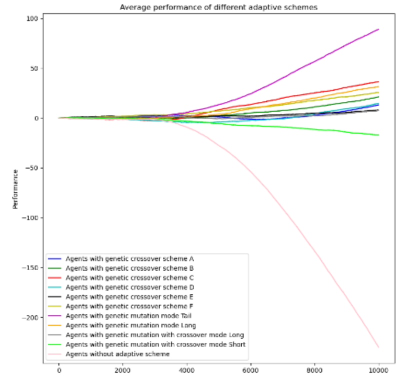
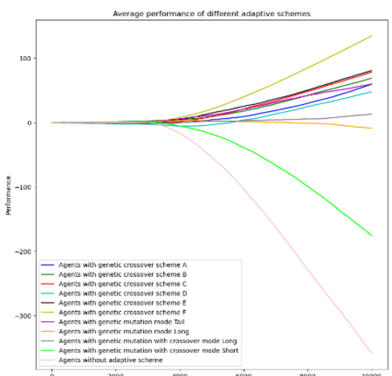
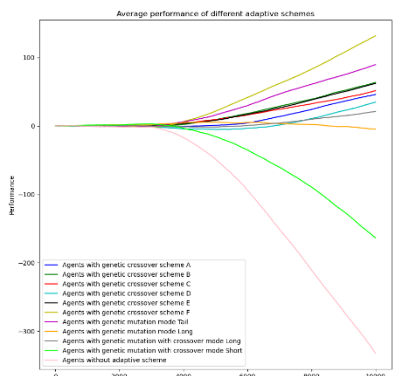
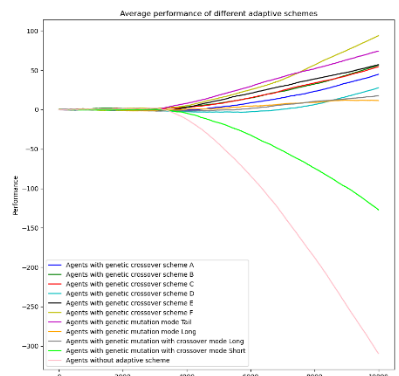
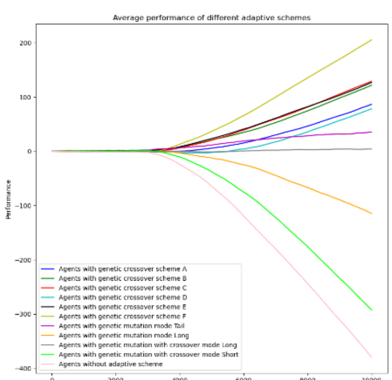
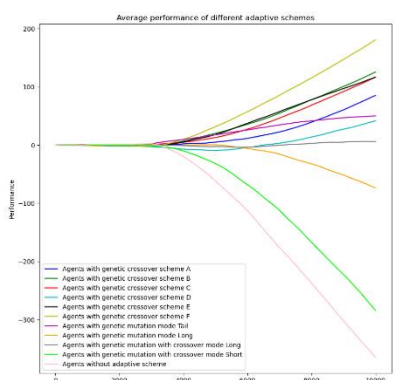
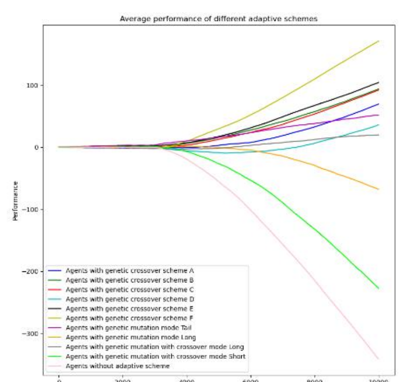
$n = 0.3$







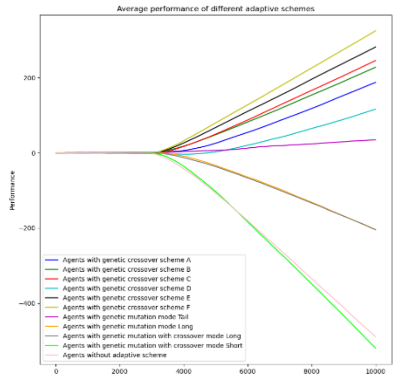
MSDM5003 (Fall 21/22) Final Project

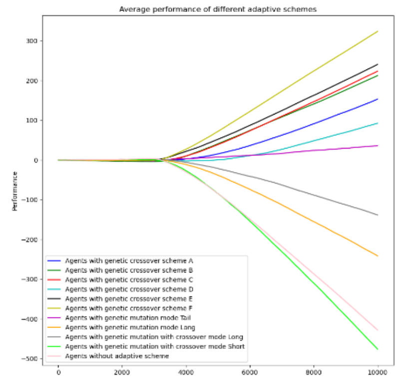
	$\tau = 40$	$\tau = 50$	$\tau = 60$
$n = 0.1$			
$n = 0.2$			
$n = 0.3$			

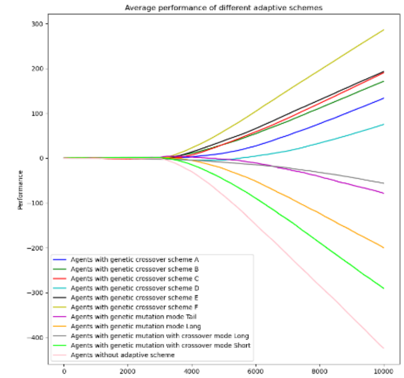
MSDM5003 (Fall 21/22) Final Project

$\tau = 10$ | $\tau = 20$ | $\tau = 30$

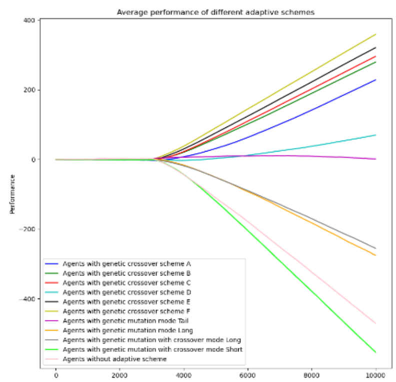
$n = 0.4$

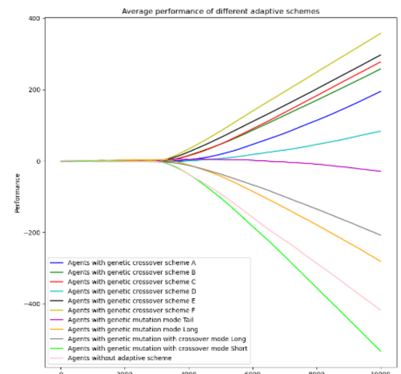


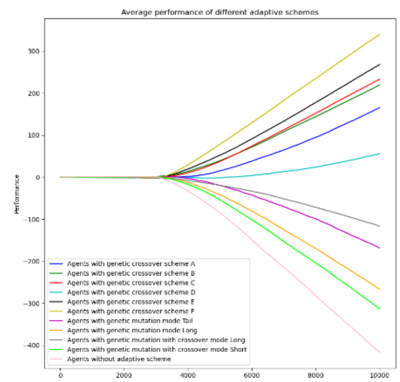




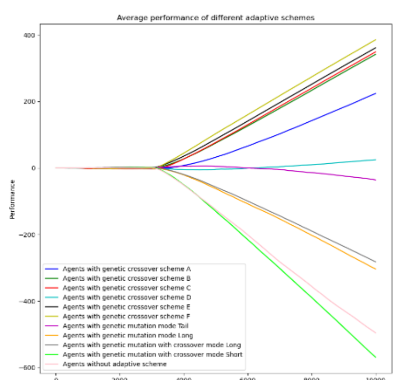
$n = 0.5$

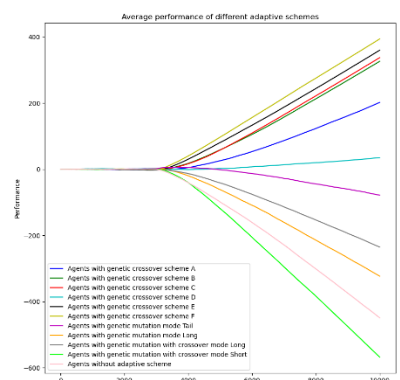


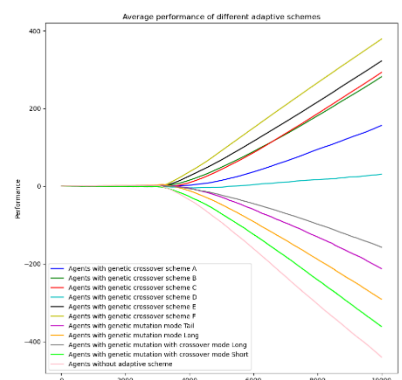




$n = 0.6$







$\tau = 40$ $\quad$ $\tau = 50$ $\quad$ $\tau = 60$

$n = 0.4$

$n = 0.5$

$n = 0.6$

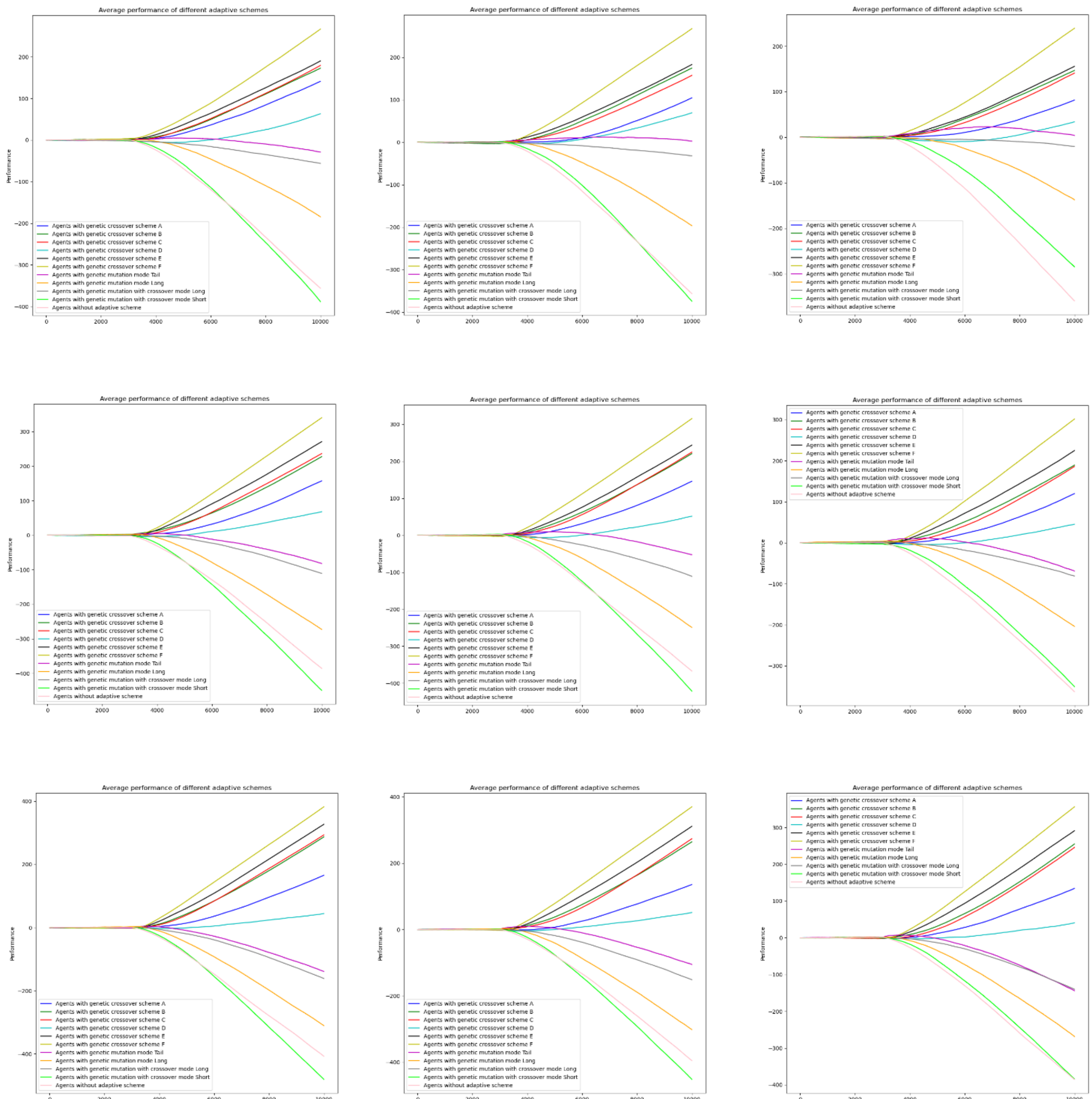


Fig. 4 – Average performance of different schemes in mixed game

For ease of comparison, the performance of different schemes are ranked in Fig. 5 (with 1 being the best and 11 being the worst). In general, crossover schemes outperform mutation and crossover-mutation schemes with some exemptions that mutation perform better when n is small and τ is large.

		Scheme										
n	**τ**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**
0.1	10	6	3	4	5	2	1	7	9	8	10	11
0.1	20	5	2	3	7	4	1	6	8	9	10	11
0.1	30	4	7	5	6	3	1	2	8	9	10	11
0.1	40	8	5	4	7	3	2	1	6	9	10	11
0.1	50	9	5	4	8	6	1	2	3	7	10	11
0.1	60	7	5	2	6	8	4	1	3	9	10	11
0.2	10	5	4	3	6	2	1	7	9	8	10	11
0.2	20	5	4	3	6	2	1	7	9	8	10	11
0.2	30	5	3	2	6	4	1	6	9	8	10	11
0.2	40	6	4	3	7	2	1	5	9	8	10	11
0.2	50	6	3	5	7	4	1	2	9	8	10	11
0.2	60	6	4	5	7	3	1	2	9	8	10	11
0.3	10	5	4	3	6	2	1	7	9	8	11	10
0.3	20	5	4	3	6	2	1	7	9	8	11	10
0.3	30	5	4	3	6	2	1	8	9	7	10	11
0.3	40	5	4	2	6	3	1	7	9	8	10	11
0.3	50	5	2	4	7	3	1	6	9	8	10	11
0.3	60	5	3	4	7	2	1	6	9	8	10	11
0.4	10	5	4	3	6	2	1	7	9	8	11	10
0.4	20	5	3	4	6	2	1	7	9	8	11	10
0.4	30	5	4	3	6	2	1	8	9	7	10	11
0.4	40	5	4	3	6	2	1	7	9	8	11	10
0.4	50	5	3	4	6	2	1	7	9	8	11	10
0.4	60	5	3	4	6	2	1	7	9	8	10	11
0.5	10	5	4	3	6	2	1	7	9	8	11	10
0.5	20	5	4	3	6	2	1	7	9	8	11	10
0.5	30	5	4	3	6	2	1	8	9	7	10	11
0.5	40	5	4	3	6	2	1	7	9	8	11	10
0.5	50	5	4	3	6	2	1	7	9	8	11	10
0.5	60	5	3	4	6	2	1	7	9	8	10	11
0.6	10	5	4	3	6	2	1	7	9	8	11	10
0.6	20	5	4	3	6	2	1	7	9	8	11	10
0.6	30	5	4	3	6	2	1	8	9	7	10	11
0.6	40	5	4	3	6	2	1	7	9	8	11	10
0.6	50	5	4	3	6	2	1	7	9	8	11	10
0.6	60	5	3	4	6	2	1	8	9	7	10	11

Fig.5 – Rank of different schemes

4. Conclusion

From the simulated results, we have shown that systems become more efficient when the agents periodically change their strategies with any of the adaptive schemes described. Among different kinds of adaptive scheme, crossover schemes outperform mutation schemes and crossover-mutation schemes collectively. As an individual agent, he can revert the losing situation by applying any of the adaptive schemes described provided other agents do not react. In a mixed game where agents applied different adaptive schemes under same game setting, crossover schemes usually perform better. Among different choices of parent and target strategies to be replaced in crossover schemes, scheme **A** and **D** perform worst. Since both schemes replaced the parent strategies after crossover, we can conclude that replacing the parents is not a good choice for the crossover scheme. It is also interesting to observe that when **n** is small (0.1) and $\boldsymbol{\tau}$ is large (>=40), mutation scheme **G** performs well. One reason is that when **n** is small and $\boldsymbol{\tau}$ is large, the number of times that crossover schemes' agents can change their strategies is low as the worst 10% agents will be dominated by those without adaptive strategies. They are not able to develop very good strategies before the domination which gives mutation scheme a chance to beat them. Besides, despite crossover-mutation schemes have revenge ability, they are not good enough when comparing with others. Under certain combination of **n** and $\boldsymbol{\tau}$, the "Short" mode even perform worse than those without adaptive scheme. The results somehow align with the natural world that humans are reproduced through crossover of chromosomes.

Mutation may sometimes give better outcome but most of the time the opposite, making it not sustainable. The same also applied to crossover with mutation.

5. Reference

1. M. Sysi-Aho, A. Chakraborti, and K. Kaski, Physical Review E 69, 036125, Searching for good strategies in adaptive minority games, (2004).

6. Appendix

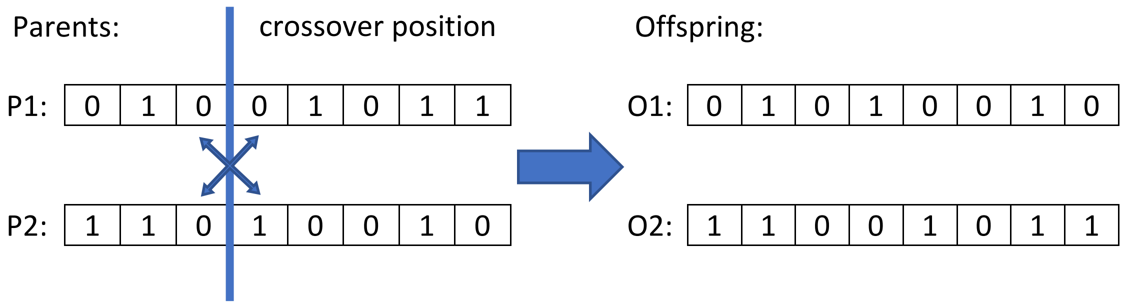
1. Minority Game

The Minority Game is played by an odd number, denoted by **N**, agents who have to choose to join either one of the two groups, denoted by 0 and 1, at each time step. At time **t**, the group having fewer number of agents will be the winning group and all agents belong to the group will be awarded a point while others do not. All agents are assumed to be able to memorize up to **m** latest winning groups. The game state is a bit string consisting of the latest **m** winning groups and will be updated dynamically. Each agent owns **S** strategies, each of size $2^{\mathbf{m}}$ storing the group choice for each possible game state. Each strategy has a virtual score which will be increased by 1 when it can correctly predict the winning group. Strategy among an agent with the highest virtual score at each time step will be the agent's best strategy and he should choose the group to join according to it. The game will be played repeatedly **T** times.

2. One-point genetic crossover scheme

Inspired by genetic evolution in biology, new strategies are generated from parents' strategies through crossover. Under the one-point genetic crossover scheme, two existing strategies are selected as parents. Then, one point is randomly selected as the crossover position. The parent strategies will then be separated into two parts and crossover with each other to form two offspring strategies.

Illustration:



Finally, two existing strategies are selected to be replaced by the offspring strategies and the corresponding virtual scores are updated as that of the parent strategies.

The six crossover schemes of interest (Scheme **A** to **F**) correspond to six different combinations of parent strategies and strategies to be replaced as follows:

(A) Two strategies are selected randomly as parents and the same two will be replaced by the offspring generated.

(B) Two strategies are selected randomly as parents and another two randomly selected strategies will be replaced by the offspring generated.

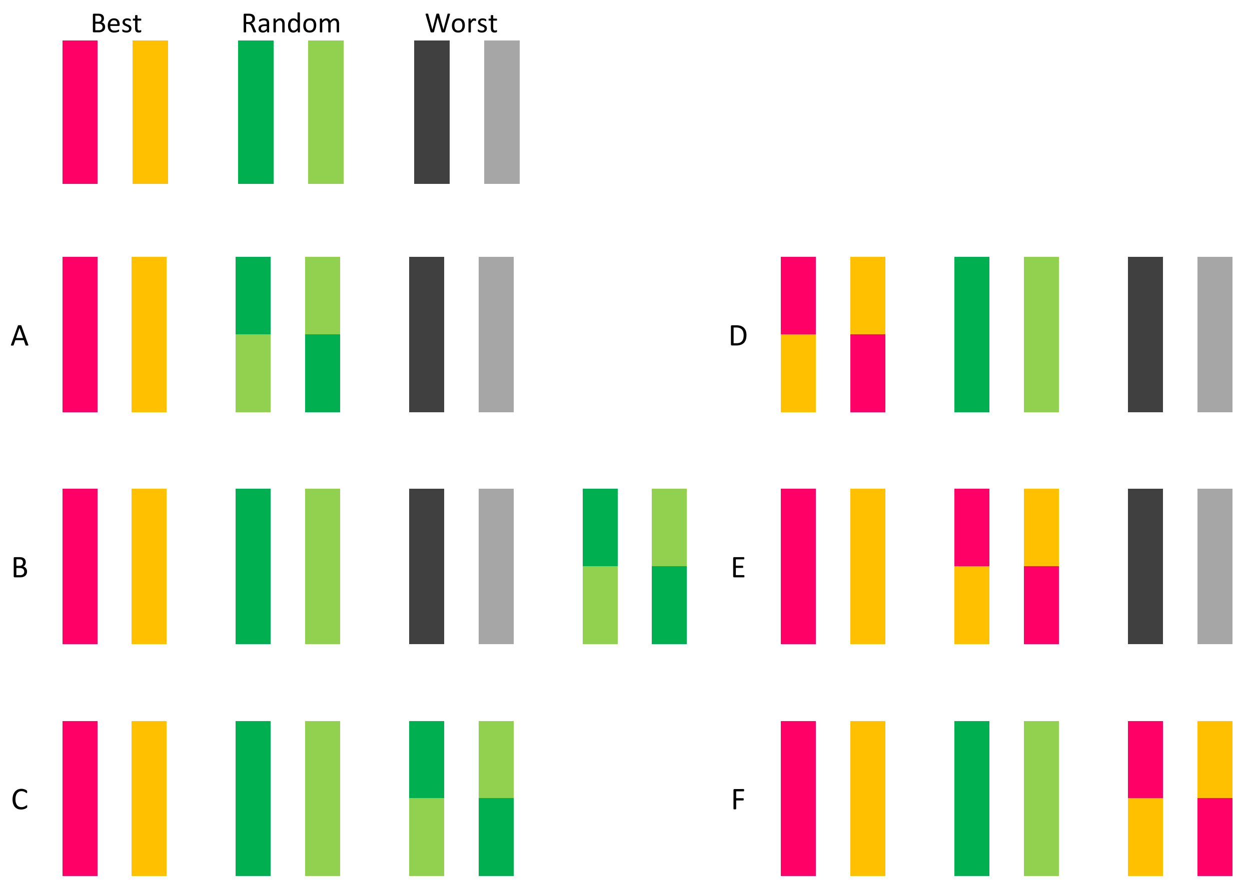
(C) Two strategies are selected randomly as parents and two worst strategies will be replaced by the offspring generated.

(D) Two best strategies are selected as parents and the same two will be replaced by the offspring generated.

(E) Two best strategies are selected as parents and two randomly selected strategies will be replaced by the offspring generated.

(F) Two best strategies are selected as parents and two worst strategies will be replaced by the offspring generated.

Illustration:

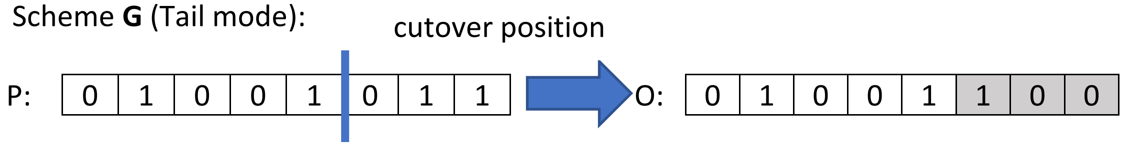


3. One-point genetic mutation scheme

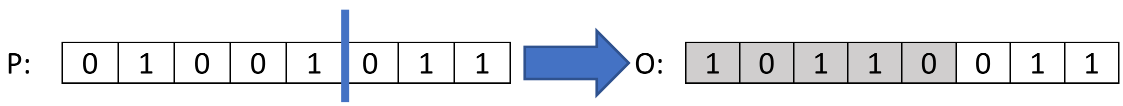
Similar to the crossover scheme, a random position is selected as the cutover point and the target strategy string will be separated into two parts. Two modes of mutation scheme are of interest, namely “Tail” and “Long”. For the “Tail” mode, all bits after the cutover point will be flipped (0→1 and 1→0). For the “Long” mode, bits in the longer chain after separation will be flipped. After the mutation, the virtual score of the target strategy will be updated as that of the best strategy. For adaptive scheme **G** and **H**, the worst two strategies are selected as the target.

Illustration:

Scheme **G** (Tail mode):



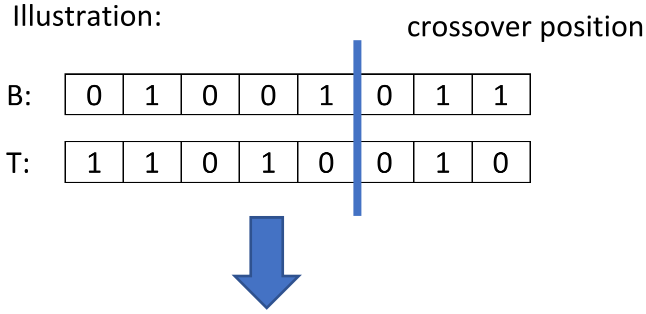
Scheme **H** (Long mode):



4. One-point genetic crossover with mutation scheme

   A random position is selected as the crossover point and the best strategy string is separated into two parts. A target strategy is also separated into two parts according to the same crossover point. Two modes of crossover-mutation scheme are of interest, namely “Long” and “Short”. For the “Long” mode, the longer part of the separated best strategy will crossover with the flipped string of the shorter part of the separated target strategy. For the “Short” mode, the shorter part of the separated best strategy will crossover with the flipped string of the longer part of the separated target strategy. After the crossover mutation, the virtual score of the target strategy will be updated as that of the best strategy. For adaptive scheme **I** and **J**, two worst strategies are selected as the target strategies.

   Illustration:



Scheme **I** (Long mode):

O:	0	1	0	0	1	1	0	1

Scheme **J** (Short mode):

O:	0	0	1	0	1	0	1	1

5. Scaled utility of the system

The total utility of the system is just the number of winning agents at each iteration and the scaled utility is given by:

$U(x_t) = \left[\left(1 - \theta(x_t - x_M)\right)x_t + \theta(x_t - x_M)(N - x_t)\right]/x_M,$

where $x_M = \frac{N-1}{2}$, $x_t \in \{0,1,2,\dots,N\}$ is the number of agents choosing 0 or 1 at time t and $\theta(x) = \begin{cases} 0 \text{ when } x \leq 0 \\ 1 \text{ when } x > 0 \end{cases}$ is the Heaviside unit step function.

The scaled utility will be maximum, denoted by $U_{max} = 1$, when x_t equals either x_M or x_{M+1}. The system is said to be more efficient if deviation from U_{max}, i.e. $1 - U(x_t)$, is small.