ABSTRACT

This paper examines which elements are necessary for coevolutionary genetic algorithms to evolve cooperative strategies under pure egoistic considerations. Since competitions and cooperations coexist in an auction-based manpower allocation problem, the problem is adopted for further investigation. To alleviate analytical burden, the problem is abstracted to a resource-bidding game under the Nash game framework. A mathematical model for the resource-bidding game is defined and several special cases are illustrated. One of these special cases, named c-mNE, is further investigated due to the existence of cooperative modes. Various kinds of egoistic fitness functions and evolutionary mechanisms are experimented on c-mNE. Based on the experimental results, this paper suggests that coevolutionary mechanisms which properly eliminate aggressive strategies and preserve cooperative strategies can evolve cooperative modes under the pure egoistic assumption.

1. INTRODUCTION

In Mother Nature, resources, e.g., foods, shelters, and sexual mates are usually limited for species. Under the limited resource scenario, both competitions and cooperations occur simultaneously. For instance, Grant’s gazelles fight with each other for mating rights, while herded for averting predators [7]. Cooperative behaviors and coalitional games have been studied for decades in the game theory field [6]. Mathematicians and economists have tried to explain cooperations explicitly in mathematics [2, 11]. Besides, scientists in other fields try to explain cooperations form different views.

Ethologist Dawkins [5] suggested that “I shall argue that a predominant quality to be expected in a successful gene is ruthless selfishness. This gene selfishness will usually give rise to selfishness in individual behaviour.” He also stated that “what is a single selfish gene trying to do? It is trying
to get more numerous in the gene pool." Moreover, "a gene might be able to assist replicas of itself that are sitting in other bodies. If so, this would appear as individual altruism but it would be brought about by gene selfishness." Dawkins argued that behaviors are evolved by genes and even altruism is evolved under the egoistic consideration. His argument implies that cooperative behaviors are evolved under egoism. Is it true for evolutionary computations where the concept is borrowed from the nature? This is the radical question in our research.

To begin with, we need a suitable test problem where competitions and cooperations coexist. In addition, this problem should be common in the real life so that we may obtain meaningful insights. A manpower allocation problem is a proper choice, since the staff work cooperatively to complete the same task while compete for taking days off on important holidays. Computer scientists have tried to design algorithms to solve this NP-hard problem [14] more efficiently; however, unexpressive preferences of staffs are difficult for scientists to understand and model. Auction is one of the solutions to this problem [4, 9, 16].

De Grano et al. [9] used a two-stage approach to solve a nurse scheduling problem (one type of the manpower allocation problem). In the auction stage nurses bid for their preferred working shifts and days off using points. An optimization model allocated shifts to the nurse who bid the most points insofar as possible with constraint satisfied. Then, schedule completion stage allocates remaining unfilled shifts to nurses who have not yet met their minimum required working hours. Although De Grano et al. do not have actually take their method into practice, their experimental result suggested that the approach would work under various conditions.

This paper uses coevolutionary genetic algorithms to investigate cooperative behaviors in the auction-base manpower allocation problem. Using the coevolutionary concept to understand and analyze games is popular [3, 13, 15], whereas only few researches discussed necessary components of coevolution of strategies in games [12]. What kind of strategies can be evolved with a specific coevolutionary mechanism? The answer to this question is still vague.

In this research, we attempt to investigate which elements are necessary for coevolutionary genetic algorithms to evolve cooperative strategies under pure egoistic considerations. Egoism means that an individual’s utility function would depend only on its fitness (interests in game theory), whereas altruism means that utility function depends on all individuals’ fitness [1]. Our goal is to understand whether an element lead coevolutionary genetic algorithms to evolve cooperative strategies or not. Furthermore, we use the findings of this paper to explain cooperative behaviors in the nature.

The rest of the paper is composed of three parts: (1) the definition of the mathematical model and interesting cases of the resource-bidding game, (2) methods and experimental results of coevolutionary genetic algorithms, and (3) conclusions and the future work.

## 2. MATHEMATICAL MODEL

To directly analyze an auction-based problem or to simulate behaviors of bidders entirely in the computer system are difficult and complicated. Hence, we abstract the auction-based manpower allocation problem under the Nash game framework [6] with four following assumptions.
1. The seal bid, single run auction is adopted.
2. Players receive fixed and equal points at the beginning of each turn.
3. Players have to bid all points in each turn.
4. For equal bids, we use the seniority for tie break — the most senior player wins the bid.

Important holidays are the competitive resources in our case. In other applications, they can be anything. In addition, more senior players having the priority is also a common scene in the nature and societies. The Japanese idiom *NenKouJoRetsu* means that more years in the organization, more chances of promotions. The detailed model definition is described in the next section.

### 2.1 Model Definition

The resource-bidding game is defined as follows.

**Definition 1.** A resource-bidding game \( G = (R, P) \) is an ordered pair, where

\[
R = \{r_1, r_2, ..., r_m\} \text{ represents } m \text{ kinds of resources, where } r_j \in \mathbb{Z} \text{ for } j = \{1, ..., m\} \text{ is the amount of } r_j \text{ for players to bid,}
\]

\[
P = \{P_1, P_2, ..., P_n\} \text{ represents } n \text{ players. } P_i = \{A_i, f_i\}, \text{ for } i = \{1, ..., n\} \text{ is also an ordered pair, where}
\]

- \( A_i = \{\bar{a}_{i,1}, ..., \bar{a}_{i,z}\} \) is an action set of player \( P_i \), where \( z \) means the number of all types of action. An element \( \bar{a}_{i,j} = (b_1, ..., b_m) \) is a \( m \)-dimensional vector, where \( b_j \in \mathbb{Z} \text{ for } j = \{1, ..., m\}. \) The quantity \( b_k \) of vector \( \bar{a}_{i,j} \) indicates \( P_i \) bid \( b_k \) points for resource \( r_k \) for \( k = \{1, ..., m\} \) of \( j \)-th type action.

- \( f_i : R \to \mathbb{N} \) is an utility function mapping different kinds of resources to player \( i \)'s payoffs.

In each turn, only the first \( r_j \) highest bidding-players can win the bid of the resource \( r_j \). If the number of bidders is less than \( r_j \), the extra amount of resource \( r_i \) is wasted. If player \( i \) bid the resource \( r_j \), he can get \( f_i(r_j) \) payoff in that turn. This game holds \( t \) turns, where \( t \in \mathbb{N} \). Finally, players tend to maximize their own payoffs.

The definition of the resource-bidding game is simple, yet variational. The next section illustrates three interesting cases under the definition.

### 2.2 Interesting Cases

Three special cases \( c-\text{pNE} \), \( nc-\text{pNE} \) and \( c-\text{mNE} \) of the resource-bidding game are illustrated as follows.

\( c-\text{pNE} \) is a two-players, three-resources game. The more senior player is \( P_1 \). Each player has two points in each turn. The resource set is \( R = \{1, 1, 1\} \). Utility functions for players are \( f_1 = f_2 \) where \( f_1(r_1) = 5 \), \( f_1(r_2) = 4 \) and \( f_1(r_3) = 2 \). Pure Nash equilibrium strategy [6] exists in this game. A set of \( P_1 \)'s action vector \((0, 1, 1)\) and \( P_2 \)'s action vector \((2, 0, 0)\) is the Nash equilibrium strategy (Figure 1). In Figure 1, no other candidate action can unilaterally enable a player to take more payoff than the Nash equilibrium strategy. In other words, no player has an incentive to change their strategy unilaterally (the definition of the Nash equilibrium). \( c-\text{pNE} \) is the abbreviation of the resource-competitive game where pure Nash equilibrium strategy exists. By the word resource-competitive, we mean that two players have...
Figure 1: A Nash equilibrium strategy in c-pNE. Tabular structures in this figure are used to represent the $A_i$ for $i \in \{1, 2\}$. Numbers inside the structures indicate $f_i(r_j)$, for $i \in \{1, 2\}$ and for $j = \{1, 2, 3\}$. The candidate actions illustrate possible actions which enable players to take much payoff than the Nash strategy.

nc-pNE is almost as same as c-pNE, except for the utility functions, where $f_1(r_1) = 5$, $f_1(r_2) = 4$, $f_1(r_3) = 2$, $f_2(r_1) = 2$, $f_2(r_2) = 4$ and $f_2(r_3) = 5$. A pure Nash equilibrium strategy $((1,1,0),(0,0,2))$ exists (Figure 2).

Figure 6 shows that no pure Nash equilibrium strategy exist in c-mNE. The iterated elimination dominated strategy (IEDS) [6] technique is adopted to reduce the size of the game. Consider the mixed Nash equilibrium strategy [6] $((2/5,1/5,2/5),(1/5,2/3,2/15))$, where $\left(2/5,1/5,2/5\right)$ and $\left(1/5,2/3,2/15\right)$ are the probabilistic vectors of $P_1$ and $P_2$ respectively. Each vector represents a probabilistic distribution for player to chooses actions of the IEDS-performing game. For the mixed Nash equilibrium strategy, we evaluate the expected payoff is 5 for $P_1$ and 3 for $P_2$.

To examine the necessary components of coevolution of cooperative strategies in c-mNE, the cooperation needs to be clearly defined. For objective investigations, the excepted payoff of the Nash equilibrium strategy is used as a metric. The Nash equilibrium is a strategy set which rational players compete among each other. Hence, if all players earn more payoff than the excepted payoff of the Nash equilibrium, we believe that players do not only compete with each other, but try to cooperate with each other. We define two modes $p_{co}$ and $c_{co}$ of partial cooperation and complete cooperation in c-mNE. The word complete means no resource-wasting occur (players can take all resources in the system), whereas the word partial means players cannot take all resources in the system. The definitions of $p_{co}$ and $c_{co}$ are defined as follows.

\begin{definition}
\[p_{co} = \begin{cases}
\text{true}, & \text{if } \rho_1 > 5, \rho_2 > 3, \rho_1 + \rho_2 < 9, \\
\text{false}, & \text{otherwise},
\end{cases}\]
\end{definition}

\begin{definition}
\[c_{co} = \begin{cases}
\text{true}, & \text{if } \rho_1 > 5, \rho_2 > 3, \rho_1 + \rho_2 = 9, \\
\text{false}, & \text{otherwise},
\end{cases}\]
\end{definition}

We further analyse c-mNE in Section 2.3.

2.3 Analysis of c-mNE

The Nash equilibrium strategy set for $i = \{1, 2\}$. Numbers inside the structures indicate $f_i(r_j)$, for $i = \{1, 2\}$ and for $j = \{1, 2, 3\}$. The candidate actions illustrate possible actions which enable players to take much payoff than the Nash strategy.
where $\rho_1, \rho_2$ refer to $P_1$ and $P_2$'s average payoff of each turn respectively. The maximum payoff of the whole system is 9 in one turn. $(f_i(r_1) + f_i(r_2) + f_i(r_3) = 9, \text{for } i = 1 \text{ or } 2)$.

To further analyze c-mNE, we categorize (2, 0, 0) and (0, 2, 0) of $P_1$ and $P_2$ as conservative strategies. (1, 0, 1) and (0, 1, 1) are extra-taken strategies (or cooperative strategies). (1, 1, 0) and (2, 0, 0) are aggressive strategies. Figure 4 shows a competition among these strategies in c-mNE.

Moreover, many cooperative modes would exist for the players to earn more payoff than the excepted payoff of the mixed Nash equilibrium strategy. For example, they can get $r_1$ and $r_2$ receptively and share $r_3$ in turn as Figure 5 shows. The investigation for the necessary components of the coevolution of cooperative modes is described in the next section.

3. METHOD AND EXPERIMENTAL RESULT

In this paper, we use three metrics to measure the degree of cooperation. The meanings of the metrics are described as follows.

- $p_{co\_number}$ represents the number of how many experiments where partial cooperation occurred.
- $p_{co\_mean}$ is the mean times of games where partial cooperation occurred.
- $p_{co\_div}$ represents the standard derivation of $p_{co\_mean}$.

In the remainder of this section, we describe methods of coevolutionary genetic algorithms used in this paper and demonstrate experiment results for each method. Section 3.1 uses the simple genetic algorithm (SGA) [8] as coevolutionary algorithm for experiment. Section 3.2 replaces replacement method of SGA with the restricted tournament
selection (RTS) [10]. Section 3.3 incorporates a new method, named switching fitness function (SFF), during evolution with SGA+RTS. Section 3.4 introduces modified RTS (mRTS).

3.1 SGA

We use SGA as our coevolutionary method in this section. Parameters, evolutionary mechanisms and operators are described specifically as follows (Figure 6).

1. The experiment are executed 1000-runs for stable results.

2. The number of turns of the game, \( t \), is set at 6.

3. The population is divided into two sub-populations, \( \mathbb{P}_1 \) and \( \mathbb{P}_2 \) which represent \( P_1 \)'s chromosomes and \( P_2 \)'s chromosomes, respectively. These two sub-populations evolve with operators of SGA. The size of each sub-population is 100.

4. A strategy is encoded into a chromosome of 6-ary and length 7. Each chromosome only matches against chromosomes in the opponent sub-population. Each gene represents the player’s responsive action for response to rival’s action in the last turn. An additional bit represents the first-turn action of the player. Each 6-ary represents one kind of action (\( \vec{a}_{i,j} \)).

5. Binary tournament selection, uniform crossover (crossover rate:0.8) are adopted as evolutionary operators of SGA. The maximum number of generations is 2000. To diminish uncertainty, we focus on effects of selection+crossover and exclude mutation mechanisms from this experiment.

6. Ten kinds of fitness functions are adopted for experiment (Table 1). In each generation, a chromosome is evaluated by matching with chromosomes in the sub-population of opponents. Its fitness is then calculated according to its payoffs and the rival-choosing scheme.

By rival-choosing, we mean that players can choose their rival to match against. Therefore, different ways of choosing rival indicate different kinds of fitness functions. For instance,

\( \text{avg} \) refers to the case where players choose their rival by uniformly random,

\( \text{med} \) refers to the case where players choose their rival with the median payoffs, the most probable one,

\( \text{max} \) refers to the case where players choose their rival who is the most advantageous to them, and

\( \text{min} \) refers to the case where players choose their rival by planning for the worst.

Rival choosing concept is common in the real-life. For example, an experienced pickpocket will choose an easy target as a victim, and a boxer have to challenge a champion for the title.

The results of SGA are shown in Figures 7 and 8. In addition, we also examine the result under altruistic consideration for comparison (Table 2). The results show that

1. SGA converges to one fixed point for \( \text{mean} \) are 1000 times of \( \text{std} \), and that

2. SGA hardly evolves cooperative modes whichever the fitness function is adopted, since small \( \text{no} \) and large \( \text{std} \).

The major reasons for SGA to hardly evolve cooperative modes are that
Table 1: Fitness Functions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Use of all matches as fitness, where x is</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>the average.</td>
<td></td>
</tr>
<tr>
<td>+d</td>
<td>avg + 1 standard deviation.</td>
<td></td>
</tr>
<tr>
<td>+2d</td>
<td>avg + 2 standard deviation.</td>
<td></td>
</tr>
<tr>
<td>-d</td>
<td>avg - 1 standard deviation.</td>
<td></td>
</tr>
<tr>
<td>-2d</td>
<td>avg - 2 standard deviation.</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>the maximum.</td>
<td></td>
</tr>
<tr>
<td>3/4</td>
<td>the third-quartile.</td>
<td></td>
</tr>
<tr>
<td>med</td>
<td>the median.</td>
<td></td>
</tr>
<tr>
<td>1/4</td>
<td>the first-quartile.</td>
<td></td>
</tr>
<tr>
<td>min</td>
<td>the minimum.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: The implementation schemes. Each strategy (chromosome) matches against all chromosomes among opposite sub-population, like a bipartite graph. The responsive actions refer to response to rival’s last turn action.

1. cooperative strategies are eliminated rapidly for existence of aggressive strategies, and that

2. convergence times are unequal of two sub-populations.

Although SGA hardly evolves cooperative modes under the egoistic consideration, it eliminates the dominated strategies precisely. In other words, SGA performs the IEDS in c-mNE.

By observing the sub-populations of the final results, SGA evolves conservative strategies for two players or evolves extra-taken strategies for one player and conservative strategies for the other. Players win payoffs which are equal to the expected payoffs of mixed Nash equilibrium strategies in the former kind of results, and only one player win payoffs which more than the expected payoffs of mixed Nash equilibrium strategies in the latter kind of results.

Table 2: Experimental Result under Altruism of SGA with avg fitness

<table>
<thead>
<tr>
<th></th>
<th>The total payoff of two players is 9 in each turn</th>
<th>Average payoffs of P₁ and P₂ are much than payoffs of the mixed strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>mean</td>
<td>10000</td>
<td>0</td>
</tr>
<tr>
<td>div</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 7: Experimental Result in p_co of SGA. ran means the strategy randomly generated without evolution.
3.2 SGA + RTS

Niching techniques are often applied to SGA to preserve the population diversity for multimodal problems. They are also important for coevolutionary genetic algorithms. In Lin and Yu’s research [12], they found that niching techniques are helpful to coevolve pure strategies. In this subsection, we adopt RTS as our niching mechanism for SGA. Detailed procedures of SGA+RTS are showed in Algorithm 1.

**Algorithm 1** Coevolutionary GA with RTS

1. Randomly initialize two sub-populations $sP_1$ and $sP_2$.
2. Evaluate the fitness of each chromosome in $P_1$ and $P_2$ by playing games with all chromosomes in the sub-population of opponents.
3. $sP_1$ and $sP_2$ generate a new sub-populations $sP'_1$ and $sP'_2$ by applying a crossover operator respectively.
4. Evaluate the fitness of each chromosome in $sP'_1$ and $sP'_2$ by playing games with all individual in the new sub-population of opponents.
5. For each offspring $x$ in $sP'_1$ and $sP'_2$, perform the restricted tournament selection with its own old sub-population $sP_1$ or $sP_2$.
The experimental results of Figures 9 and 10 show that

1. RTS improves the performance by keeping the sub-populations diversity. The results are consistent with Lin and Yu’s finding.

2. \( \min \) outperforms \( \max \). The reason is detailed in Section 3.3.

3. \( \text{avg}, \text{med} \) and \( 1/4 \) fitness functions outperform the others.

4. Large \( p_{\text{co div}} \) and \( c_{\text{co div}} \) indicate that the experiment results are not stable.

### 3.3 SGA + RTS + SFF

Section 3.2 shows that \( \min \) outperforms \( \max \). The reason is clear if we observe the chromosomes of the final results in the experiment. \( \min \) leads to conservative strategies, whereas \( \max \) leads to aggressive strategies which eliminate cooperative strategies. An intuitive thinking is starting with \( \min \) and switching to \( \max \) in the halfway to improve the performance. Specifically, we adopt \( \min \) to preserve cooperative strategies from aggressive strategies and eliminate aggressive strategies. Then, we switch to \( \max \) to enable cooperative strategies to outperform conservative strategies.

The Experimental results (Figures 11 and 12) show that

1. SFF improves the performance, although the performance is worse than \( \text{avg}, \text{med} \) and \( 1/4 \) of SGA+RTR, and that

2. \( c_{\text{co mean}} \) peaked in the 1700th generation. It is reasonable that chromosomes with different fitness functions need time for evolving.

The experimental results prove that switching the fitness function (rival-choosing criterion) in proper time improves the performance. However, we need a self-adaptive method for finding the proper switching timing. We leave it as our future work.

### 3.4 SGA + mRTS
The results of Section 3.2 show that RTS keeps the diversity of sub-populations and improves the performance. However, RTS may also keep too many types of strategy and hinder cooperation from being evolved. In addition, RTS in coevolutionary algorithms has a controversial procedure. RTS evaluate the fitness of chromosomes in new sub-population by comparing with rival’s chromosomes in new sub-population. Then, RTS checks whether chromosome in new sub-population can replace chromosome in old sub-population by compare their fitness. This procedure costs much computational time for comparison with new sub-populations. Moreover, it is unreasonable for comparing fitnesses which evaluated by different rivals. Hence, we modify the procedure to a less computational burden and more reasonable procedure.

We introduce the most competitive rival evaluation (a modified RTS) as our the new operator for replacement. The only difference between RTS and mRTS is that mRTS evaluates fitness of the new sub-population by match against the highest payoff chromosome of rival’s old sub-population. This modification leads RTS to consume less computational power, since every chromosome in the new sub-population need only one time of match. In addition, adaption to rival’s best player is also reasonable, since the best player has the least chance been replaced (for his highest payoff). Moreover, mRTS can be viewed as another way of rival choosing. Algorithm 2 shows detail procedures of mRTS.

The experiment results (Figures 13 and 14) are satisfactory by two reasons.

1. \textit{p\_co mean} and \textit{c\_co mean} improved obviously in all fitness functions, especially \(\frac{1}{4}\). The results show that \(p\_co\) mean plus \(c\_co\) mean is about 2200 times. Cooperative behaviors happened beyond \(\frac{1}{5}\) matches.

2. Decreases of \(p\_co\) div and \(c\_co\) div imply more stable experiment results.

We speculate that the most competitive rival evaluation reduces much noise and hence yields a better performance. However, we do not have any convincible evidences and connect the result with cooperative behaviors in the nature for now. We leave it as our future work.
Algorithm 2 Coevolutionary GA with mRTS

1. Randomly initialize two sub-populations \( sP_1 \) and \( sP_2 \).

2. Evaluate the fitness of each chromosome in \( sP_1 \) and \( sP_2 \) by playing games with all chromosomes in the sub-population of opponents.

3. \( sP_1 \) and \( sP_2 \) generate a new sub-populations \( sP'_1 \) and \( sP'_2 \) by applying a crossover operator respectively.

4. Evaluate the fitness of each chromosome in \( sP'_1 \) and \( sP'_2 \) by playing games with the individual having the highest payoff in the opposite old sub-population.

5. For each offspring \( x \) in \( sP'_1 \) and \( sP'_2 \), perform the restricted tournament selection with its own old sub-population \( sP_1 \) or \( sP_2 \):
   (a) Find \( y \in W \) such that minimizes the distance between \( x \) and \( y \), where \( W \subseteq sP_1 \) or \( W \subseteq sP_2 \) and \( |W| = w \) is a random sample set with window size \( w \).
   (b) Replace \( y \) with offspring \( x \) if \( x \) has a higher fitness, otherwise discard \( x \)

6. Stop if the terminating condition is satisfied, otherwise go to step 2.

To summarize, we experimented four different mechanisms with ten different kinds of fitness functions in c-mNE. SGA yielded poor performance on each fitness function, even worse than the result generated randomly. Results of SGA+RTS showed that RTS slightly advance performance, but large standard deviation still indicated the results are unsteady. Results of SGA+RTS+SFF implied the possibility of improving performance by switching fitness function in proper time. Finally, SGA+mRTS yielded the best performance of all metrics among all mechanisms so far.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated necessary components of the coevolution of cooperative strategies under the egoistic assumption. We build an abstract mathematical model of an auction-based manpower allocation problem for easy understanding and analysis. We analyzed and illustrated this model by providing three cases, including a case named c-mNE. Then, we defined what a cooperative mode is precisely, and notice that cooperative modes exist in c-mNE. We compared results of ten types of egoistic fitness functions with various kinds of evolutionary mechanisms.

Based on the experimental result, we conclude this paper as follows:

1. Coevolutionary genetic algorithms can evolve cooperative strategies under the egoistic consideration. However, the adopted fitness function and the mechanism should preserve cooperative-oriented chromosomes and eliminate aggressive-oriented chromosomes accurately.

2. Necessary elements for evolving cooperative modes are rival’s last turn action and self’s payoff in c-mNE, since our encoding method implies rival’s last turn action and fitness functions imply self’s payoff. It is interesting that players of the game do not need to know selves’ action history for cooperations. Yao and Darwen [17] compressed self’s action history in a chromo-
some for cooperations in the iterated prison dilemma game whereas our experimental results suggested that self’s action in c-mNE is not a necessity.

3. Altruistic considerations are not a necessary component for evolving cooperative strategies. Specifically, coevolutionary genetic algorithms with a altruism-oriented fitness function easily lead players to take whole payoffs in the system, but cannot guarantee all players take payoffs which are at least equal to the excepted payoff of the mixed Nash equilibrium strategy. On the other hand, coevolutionary genetic algorithms with the egoistic fitness function easily lead players to take payoffs which are at least equal to the excepted payoff of the mixed Nash equilibrium strategy, but lead players to cooperate with each other to take whole payoffs in the system under some specific conditions.

This paper investigated the necessary components of co-evolution of cooperations, and this work still leaves many unanswered questions and plenty of room to research. For example, we would like to develop a more effective, efficient and self-adaptive evolutionary mechanism for c-mNE, because our experiment results imply possibilities of advancing performance. In addition, we want to verify whether these mechanisms are effective similarly in other games, and analyze the results theoretically.

This paper describes what a resource-bidding game is and how we tune coevolutionary genetic algorithms to evolve cooperative modes of c-mNE. This work is only our first attempt, and the ultimate goal of this research is to understand why, when and how agents cooperate with each other in the evolutionary computation or even in the real world. This paper preliminarily implies that egoism is one of the elements that encourages cooperation under the resource-competitive scenario. In addition, diversity and rival-choosing greatly affect the coevolution of cooperation. We believe that these results are general, and that further investigations may reveal more clues about the origin of cooperative behaviors in the nature.

5. REFERENCES


