



Game-theoretic Evolutionary Algorithm Based on Behavioral Expectation and its Performance Analysis

Guanci Yang, Weihua Sheng, Shaobo Li, Yang Wang & Fei Xu

To cite this article: Guanci Yang, Weihua Sheng, Shaobo Li, Yang Wang & Fei Xu (2017) Game-theoretic Evolutionary Algorithm Based on Behavioral Expectation and its Performance Analysis, Applied Artificial Intelligence, 31:5-6, 493-517, DOI: [10.1080/08839514.2017.1378205](https://doi.org/10.1080/08839514.2017.1378205)

To link to this article: <https://doi.org/10.1080/08839514.2017.1378205>



Published online: 25 Oct 2017.



Submit your article to this journal [↗](#)



Article views: 507




View related articles [↗](#)



View Crossmark data [↗](#)



Game-theoretic Evolutionary Algorithm Based on Behavioral Expectation and its Performance Analysis

Guanci Yang ^{a,b}, Weihua Sheng^b, Shaobo Li^a, Yang Wang^c, and Fei Xu^d



^aKey Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, Guiyang, China; ^bSchool of Electrical and Computer Engineering, Oklahoma State University, Stillwater, USA; ^cChengdu Institute of Computer Applications, University of the Chinese Academy of Sciences, Chengdu, China; ^dSchool of Management, Guizhou University, Guiyang, China

ABSTRACT

Game theory has been widely recognized as an important tool in many fields, which provides general mathematical techniques for analyzing situations in which two or more individuals make decisions that will influence one another's welfare. This paper presents a game-theoretic evolutionary algorithm based on behavioral expectation, which is a type of optimization approach based on game theory. A formulation to estimate the payoffs expectation is given, which is a mechanism of trying to master the player's information so as to facilitate the player becoming the rational decision maker. GameEA has one population (players set), and generates new offspring only by the imitation operator and the belief learning operator. The imitation operator is used to learn strategies and actions from other players to improve its competitiveness and applies it into the future game, namely that one player updates its chromosome by strategically copying some segments of gene sequences from the competitor. Belief learning refers to models in which a player adjusts its own strategies, behavior or chromosome by analyzing current history information with respect to an improvement of solution quality. The experimental results on various classes of problems using real-valued representation show that GameEA outperforms not only the standard genetic algorithm (GA) but also other GAs having additional mechanisms of accuracy enhancement. Finally, we compare the convergence of GameEA with different numbers of players to determine whether this parameter has a significant effect on convergence. The statistical results show that at the 0.05 significance level, the number of players has a crucial impact on GameEA's performance. The results suggest that 50 or 100 players will provide good results with unimodal functions, while 200 players will provide good results for multimodal functions.

Introduction

Current studies address the excessive sensitivity of Evolutionary Algorithms (EAs) to control parameters, premature convergence and slow computation

CONTACT Guanci Yang  guanci_yang@163.com  Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, Jixie Building 604, Guiyang 550025, P. R. China

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/UAAI.

speed, but capturing historical information and providing a predictive adaptive response to evolution requires further study. Recent research suggests that predictive adaptive responses can lead to differential development among initially similar individuals as well as provide an increase in evolutionary fitness, which is an adaptive change in long-term behavior or development due to a triggering environmental stimulus. EAs employ a set of nature-inspired computational methodologies and approaches to solving complex real-world problems that cannot be perfectly addressed using mathematical or deterministic algorithms. Several approaches have been proposed to model the specific intelligent behaviors, such as Ant Colony Algorithm (Dorigo and Gambardella 1997) based on the behavior of ants searching for food; Firefly Algorithm, inspired by the flashing behavior of fireflies (Yang 2010); Fish Swarm Algorithm, inspired by the collective movement of fish and their various social behaviors (Li, Shao, and Qian 2002; Neshat et al. 2014); Artificial Bee Colony (Karaboga and Basturk 2007); Cuckoo Search (Yang and Deb 2009); Multi-objective evolutionary algorithm based on decision space partition (Yang et al. 2017); and Artificial Immune System (Woldemariam and Yen 2010). Some other types of algorithms, such as neural computation (Maass, Natschläger, and Markram 2002) or brainstorming algorithm (Shi 2011), are inspired from the advanced activities of the human brain, which focus on the intelligence of cell behaviors. Although these types of algorithms do not guarantee the detection of the optimum value for a problem, they are sufficiently flexible to solve different types of problems due to their qualities of global exploration and local exploitation. However, the previous studies concentrated on the simulated biologic behavior rule or biological mechanism.

Game theory applies to a wide range of behavioral relations, and is now an umbrella term for the science of logical decision making in humans, animals, and computers (Ram et al. 2014). Generally speaking, the game typically involves several players, strategies or actions, orders, and payoffs, which are similar to evolutionary algorithms' individuals, genetic operators (selection, crossover, and mutation), and fitness. Both of them are population-based evolutions. On the game, the strategy selection of players is based on the payoffs expectation, and there are several learning methods to improve the players' chances of obtaining gains, such as reinforcement learning (Madani and Hooshyar 2014), belief learning (Spiliopoulos 2012), imitation (Friedman et al. 2015), directional learning (Nax and Perc 2015; Simon et al. 1999), and rule learning (Stahl 2000). The player makes a decision based on the understanding of the other players' action, and the player will become increasingly smart naturally. If the player is a candidate solution, it is possible to find the global optimal solution as the game goes on. Evolution through natural selection is often understood to imply improvement and progress, and *for studying frequency-dependent selection, game-theoretic arguments are more*

appropriate than optimization algorithms (Nowak and Sigmund 2004). Replicator and adaptive dynamics describe short- and long-term evolution in phenotype space and have found applications ranging from animal behavior and ecology to speciation, macroevolution, and human language. Evolutionary game theory is an essential component of a mathematical and computational approach to biology. Additionally, recent research (Rosenstrom et al. 2015) has suggested that predictive adaptive response can lead to differential development among initially similar individuals, and increase evolutionary fitness, which is an adaptive change in the long-term behavior or development due to an environmental exposure that triggers it. Meanwhile, there is evidence that using a learning strategy can win against the top entries from the Othello League without the explicit use of human knowledge (Szubert, Jaskowski, and Krawiec 2013). With the motivation of the above-mentioned facts, we try to investigate a novel computational algorithm – the game evolutionary algorithm – which is an optimization approach based on behavioral expectation.

The remainder of this paper is organized as follows. Section II presents some related works. Section III describes the fundamentals of our proposed algorithms. Section IV details the proposed algorithms' procedures. Section V presents the comprehensive experiments and analysis. Additional experiments with different players' size are conducted and analyzed in Section VI. Finally, Section VII concludes this paper and threads some future research issues.

Related works

Evolutionary Game Theory (Kontogiannis and Spirakis 2005) is the study of strategic interactions among large populations of agents who base their decisions on simple, myopic rules, which is to determine broad classes of decision procedures, which provide plausible descriptions of selfish behavior and include appealing forms of aggregate behavior. It seems very attractive to employ the game theory to improve the performance of some popular evolutionary algorithms. Game-theoretic differential evolution (GTDE)(Ganesan, Elamvazuthi, and Vasant 2015) was developed from differential evolution (DE) by employing cooperative and defective strategies. The defective strategy was used to reduce the influence of the child vector on the principal parent at each generation and to modify the relation to increase the degree of mutation for the child vector, whereas the cooperative strategy was used to reduce or increase the mutation factor of the child. Although the computation time of GTDE is double that of DE, the performance of GTDE was enhanced using knowledge from evolutionary game theory. Particle swarm optimizer based on evolutionary game (EGPSO) (Liu and Wang 2008) was developed by connecting game theory with the classical particle swarm optimization (PSO). The choosing strategies in evolutionary games were used to map particles who hold

current optimal solution in PSO algorithm to players who aim to pursue maximum utility, and replicator dynamics were employed to model the behavior of particles, and a multi-start technique was used to surmount premature convergence. An evolutionary algorithm based on Nash dominance for equilibrium problems with equilibrium constraints (Koh 2012) was proposed by redefining selection criteria. The Nash dominance was used to find the Nash equilibrium, which is the largest label. The game-theoretic approach for designing evolutionary programming is characterized by a mixed strategy using Gaussian and Cauchy mutations (He and Yao 2005). Although the framework of the algorithm has no difference with the popular evolutionary programming, its mutation operation is very similar to the game strategy. With an ever-increasing complexity in design engineering problems, game strategies have been proposed as one of the key technologies to save CPU usage and produce high model quality due to their efficiency in design optimization (Lee et al. 2011; Periaux et al. 2001), in which game strategies are hybridized and coupled to multiobjective evolutionary algorithms (MOEA) to improve the quality of solutions.

Game theory is also increasingly used to solve complex problems. Considering the problem of spectrum trading, with multiple licensed users selling spectrum opportunities to multiple unlicensed users, a game-theoretic framework is proposed to investigate the network dynamics under different system parameter settings and under system perturbation (Niyato, Hossain, and Zhu 2009), which is designed for modeling the interactions among multiple primary users and multiple secondary users. The theory of evolutionary game is used to model the evolution and the dynamic behavior of secondary users, and a non-cooperative game is formulated to model the competition among the primary users. An evolutionary mechanism is designed by introducing the Nash equilibrium (Wei et al. 2010) based on a practical approximated solution with two steps for the quality of service constrained resource allocation problem. First, each player solves its optimal problem independently by using the proposed binary integer programming method. Second, a mechanism to change multiplexed strategies of the initial optimal solutions of different players is employed to minimize their efficiency losses. The author claims that the Nash equilibrium always exists if the resource allocation game has feasible solutions. Concentrating on the optimality of the wireless network, the evolutionary game framework allows an arbitrary number of mobiles that are involved in a local interaction to extend on the evolution of dynamics and the equilibrium (Jiang et al. 2015; Tembine et al. 2010). To explain the evolution of structured meaning-signal mappings on why the evolutionary dynamics is trapped in local maxima that do not reflect the structure of the meaning and signal spaces, a simple game theoretical model is used, which can show analytically that

when individuals adopting the same communication code meet more frequently than individuals using different codes – a result of the spatial organization of the population – then advantageous linguistic innovations can spread and take over the population (Fontanari and Perlovsky 2008). The game theoretic trust model (Mejia et al. 2011) for the on-line distributed evolution of cooperation adapts effectively to environmental changes, but it relies on a bacterial-like algorithm to let the nodes quickly learn the appropriate cooperation behavior. A game-theoretic approach to partial clique enumeration (Bulo, Torsello, and Pelillo 2009) has proved its effectiveness, which can avoid extracting the same clique multiple times by casting the problem into a game-theoretic framework. Recently, a constant model hawk-dove game (Misra and Sarkar 2015) was designed to ensure prioritizing the local data-processing units in Wireless Body Area Networks during medical emergency situations. Aimed at obtaining the mechanism of behavior game, the game evolutionary model (Yang et al. 2016) employed to evaluate the payoffs expectation was established. Otherwise, the game theory has been used in many other fields like data mining (Qin et al. 2015) and knowledge discovery (Hausknecht et al. 2014).

Fundamental of GameEA

Origin of thought

Consider a serious study of the game theory. First, the players make a decision based on the information possessed and use this decision to gain optimal payoffs. Under non-cooperative gaming, John Nash proved that one player can gain worse payoffs except for employing the best strategy among the strategies set if the other participants do not change their decisions. It means a rational player will reach the consistency prediction by analyzing the information. Second, a repeated game is a massive form of a game, which is composed of some numbers of repetitions of some stage of the game. It reveals the idea that one person will have to take into account the impact of his current action on the future actions of other players. It can be proved that every strategy that has a payoff greater than the minimal payoff can be a Nash equilibrium, which is a very big set of strategies. The payoff is the evaluative criteria when the players chose which kind of strategy is to be used, and the payoff of a repeated game is a distinguished form of the one-shot game. Each stage of the repeated game gains payoffs, and if it is assigned to each turn of the game, it leads to a number of one-shot games. Thus, the repeated game's payoffs are related to the used strategies and the current total payoffs. If there is a base game G (static game or dynamic game), T is the times of repeating game, and the game results can be observed before starting a new G , such kind of event is called a

T -repeated game of G , marked as $G(T)$. Usually, there is a presupposition that players act rationally and make decisions intelligently, but human behavior often deviates from absolute rationality and absolute intelligence, and this conception of learning leads to the game theory.

In short, rationality and intelligent players can have an insight into the optimal strategies by grabbing and analyzing perfect information so as to gain the highest payoffs. Moreover, all participants' final payoffs are related to stage gains in a repeated game. Participants can scan historical stage information, and choose a suitable decision strategy, which can be varying equilibrium strategies for long-term interests. Additionally, in practice, learning is an important approach to gain a competitive edge when the player is lacking in rationality and intelligence. Based on these facts, we propose a new computational intelligent algorithm based on the game theory.

Model of game evolution

In an extensive form, a repeated game can be presented as a game with complete but imperfect information using the so-called Harsanyi Transformation (Hu and Stuart 2002). This transformation introduces to the game the notion of nature's choice. Hence, when a challenger competes with opponents, the opponents are considered a part of nature; therefore, the challenger is subject to nature's choice. Actually, the challenger doesn't know what nature's move was; he/she only knows the probability distribution under the various options and the winner's payoffs are known to every player. In traditional consideration, the player's action or strategy is the most important factor, but with nature's choice we can find some difference. First, the player's stable payoffs constitute winning the opponent and the other challenger. Second, if the player accepts the game, irrespective of whether it is lost or won, he/she can learn something from the opponent, which will indirectly influence future competition. Third, if the player gives up competing, he/she can improve by self-training. Let w_1 and w_2 stand for the payoffs from fighting with nature's choice and the gains from self-training, respectively; then nature has a probability of p to choose a weaker rival, thereby providing a stronger opponent with a probability of $1-p$. That nature's choice has the probability p of selecting a weaker opponent results in different performances for different challengers and even for a homogenous challenger at different stages; thus, the payoffs w_1 and w_2 are indeterminate and are not calculated in the simplified matrix of payoffs.

When the challenger is not sure about the competitiveness of the competitor, the best way is using the mathematical expectation to estimate the payoff for a risk preference player. If the challenger decides to fight with nature's choice, its expectation payoffs can be presented as $(1-p) \times w_1 + p \times (w_1 + 1)$; otherwise, if the challenger gives up the competition, the

expectation payoffs can be calculated by using $(1-p) \times w_2 + p \times w_2$. Thus, the total expectation payoffs E can be calculated using (1), and by substituting $w = w_2 - w_1$ into Equation (1), we can get Equation (2).

$$\begin{aligned} E &= (1-p)w_1 + p(w_1 + 1) - (1-p)w_2 - pw_2 \\ &= p(1 + w_1 - w_2) - (1-p)(w_2 - w_1) \end{aligned} \quad (1)$$

$$E = p(1 - w) - (1 - p)w \quad (2)$$

We use Equation (2) as the criterion of one challenger, namely that if $E \leq 0$, then the challenger gives up the competition; otherwise, if $E > 0$, then the challenger has to fight with a random selected sample (nature's choice).

Learning strategies

The traditional game theory does not discuss how to achieve equilibrium. In equilibrium reasoning, it is assumed that participants either are rational enough or can approximate equilibrium by learning (Colman 2003). Considering that a perfectly rational player exists only in ideal state, learning is very important for the player in the wrestling model. In our research, we design some learning strategies for the wrestling model, which includes *imitation* and *belief learning*. *Imitation* means learning strategies and actions from other players to improve one's competitiveness and then apply it to the next game, namely that one player updates his/her own chromosome by copying some segments of the gene sequences from the opponents, which are characterized by positive feedback. In the context of learning in games, *belief learning* refers to models in which players are engaged in a repeated game and each player adjusts his/her own strategies, behavior, or chromosome by analyzing current history information with respect to an improvement of payoffs and competitiveness against the next-period opponent behavior.

Proposed algorithm: GameEA

Framework of proposed algorithm

For the convenience of description, we define some symbols shown in Table 1. Figure 1 presents the general framework of GameEA. At the beginning, the initialization procedure generates N initial players and initializes each player's active and passive payoff of game. Within the main loop, in case the given evolution generation is not met, for each player as a challenger, choose an opponent for the challenger, and the challenger makes a decision by carefully checking the selected player, then the imitation or the belief learning procedure is employed for offspring

Table 1. Definition of symbols.

Symbol	Meaning	Symbol	Meaning
T_{\max}	Maximum of game generation	N	Size of players/population
W_1	Payoffs weight	W_2	Losses weight
P_1	Imitation probability	P_2	Learning probability
P_3	Speculative probability	n	Dimension of problem
t	t th game generation	H_a	Total number of speculations
H_s	Total number of successful speculations	I_i	i th player/individual
I_i^a	Active payoff of game of I_i	I_i^p	Passive payoff of game of I_i
I_i^{obj}	Objectives of I_i	I_i^v	Total payoffs of I_i

```

Procedure GameEA
Begin
1   $t:=0$ 
2  Initialize players set  $I^t$ ,  $I_i^a:=0$  and  $I_i^p:=0$  ( $|I^t|=N$ ,  $0<i<N$ )
3  for  $I_i \in I^t$ , evaluate  $I_i^{\text{obj}}$ 
4  repeat
5    for each  $I_i \in I^t$ 
6       $j:=\text{random}(1,N) \ \&\& \ j \neq i$ 
7      update  $I_i^v=I_i^a+I_i^p$  and  $I_i^v=I_j^a+I_j^p$ 
8      if ( $I_i^v=0$  &&  $\text{random}() < P_1$ ) then  $I_i=\text{imitation}(I_i,I_j)$ ,continue.
9      calculate  $E(I_i)$  using (3)
10     if ( $E(I_i)>0$ ) then  $I_i=\text{imitation}(I_i,I_j)$ 
11     else if ( $\text{random}()<P_2$ ) then  $I_i=\text{beliefLearning}(I_i)$ 
12    end
13    update the so-best-solution  $I_{\text{best}}$ 
14     $t:=t+1$ 
15 until  $t>T_{\max}$ 
end

```

Figure 1. General framework of GameEA.

generation. Then the offspring is used to update the so-called best solution. GameEA has only one population (players set), and generates new offspring only by the imitation operator between the challenger and the opponent and the belief learning operator by self-training strategies. In the following paragraphs, the implementation details of each component in GameEA will be detailed.

Initialization

The initialization procedure of GameEA includes the following three aspects: the decision space and the objective function; the initialization of players of set I ; and the passive and active payoffs of each player that are assigned to zero. For the function optimization problems, the initial players in set I are

randomly sampled from the decision space \mathbf{R}^n via a uniform distribution using real-valued representation, and each player's objective value is calculated by using the objective function.

Imitation operator

According to the Harsanyi Transformation, nature should perform its duty – provide a choice before applying the imitation procedure – which relates to line 6 of [Figure 1](#). Equations (2) and (3) are used to calculate the expectation payoffs of I_i :

$$E(I_i) = \mu W_1 \left(\frac{I_i^a + I_i^p}{I_i^a + I_i^p + I_j^a + I_j^p} \right) - (1 - \mu) W_2 \left(\frac{I_j^a + I_j^p}{I_i^a + I_i^p + I_j^a + I_j^p} \right) \quad (3)$$

where $\mu = \text{random}()$. Considering that the players are not intelligent enough with zero total payoffs at the beginning like the human origins stage, we let the player imitate the selected player with probability P_1 , which is a very big number just like 0.9 and allowed the chromosome to remain unchanged, namely that the weak challenger I_i has to compete with the selected I_j , and then imitate some useful information from others by using operator *imitation* (I_i, I_j), but it has the probability of doing nothing at the current turn of the game to survive some schema. This is special strategy whereby various genes can either change or remain unchanged when the player does not know much about others.

The calculation of payoffs expectation (line 9 of [Figure 1](#)) is very significant, which is a mechanism of trying to master the other's information because the total payoffs indicate one's historic strategies. Such a mechanism allows the player become more rational, and helps the player make a rational decision. It seems that players often form an aspiration based on their experiences with different actions and opponents, and payoff rewards are reinforcing or deterring depending on who they are compared to, and that decision makers adjust their aspirations as they gain experience (Borgers and Sarin 1997).

[Figure 2](#) shows the general procedure of imitation. The objective value comparison between the pair of players (line 1 of [Figure 2](#)) is the basis of the historic performance, but the comparison result impacts both the active and passive payoffs. We use a temporary variable to breed new individuals based on some imitation strategies according to the historic action, and the offspring replaces its parent only when the offspring is better than its parent so as to facilitate global convergence.

In the real world, if one person feels that some others have competitive skills, perhaps he/she will try to learn such skills, but if they do not have that, the final decision is effected by the attraction of the skills and the subjective initiative.

```

Procedure imitation( $I_i, I_j$ )
Begin
1  if ( $I_i^{obj} < I_j^{obj}$ ) then  $I_i^a := I_i^a + 1$ 
2  else  $I_i^p = I_i^p + 1$ 
3  initialize temporary variable  $I_p = I_i$  and  $B := 0$ 
4  if ( $(\text{random}() * (H_a + 1) / (2H_s + 1)) < P_3$ ) then
5      change genes of  $I_p$  by speculatively learning from  $I_j$ 
6       $B = 1$  and  $H_a = H_a + 1$ 
7  else change genes of  $I_i$  by strategically copying a segment of genes from  $I_j$ 
8  update the objectives value of  $I_p^{obj}$ 
9  if ( $I_p^{obj} < I_i^{obj}$ ) then  $I_i = I_p$ 
10     if ( $B = 1$ ) then  $H_s = H_s + 1$ 
11 return  $I_i$ 
end

```

Figure 2. General framework of the imitation operator.

Inspired by that, we use a random value, i.e. $\text{random}()$, to present the degree of subjective initiative and the ratio of succeed imitation number ($H_a + 1$) and the total number of imitations ($2H_s + 1$). If $\text{random}() * (H_a + 1) / (2H_s + 1) < P_3$, it means that all conditions indicate that the players need to improve themselves by speculatively leaning from others.

In practice, different imitation strategies can be implemented according to the characteristics of the problem. For the function optimization problems, if $r_1 = \text{rand}(0, n-1)$, $r_2 = \text{rand}(0, n-1)$, and $\beta = \text{random}()$, then if $\beta < 0.5$ then $\tau = (2\beta)^{1/16}$, otherwise $\tau = (2 - 2\beta)^{1/16}$, so we use the following strategy to implement I_i speculatively learning from I_j (line 5 of Figure 2): $I_p.\text{gen}[r_2] = 0.5(1 - \tau) I_i.\text{gen}[r_2] + (1 + \tau) I_j.\text{gen}[r_2]$, then if the value of $I_p.\text{gen}[r_2]$ is out of range, assign the required random value to $I_p.\text{gen}[r_2]$. We then implement that I_i strategically copies a segment of genes from I_j by $I_p.\text{gen}[r_1] = 0.5(1 - \tau) I_i.\text{gen}[r_1] + (1 + \tau) I_j.\text{gen}[r_1]$, and if the value of $I_p.\text{gen}[r_1]$ is out of the decision space, assign the required random value to $I_p.\text{gen}[r_1]$.

Belief learning operator

As mentioned above, belief learning refers to models in which each player adjusts its own strategies with respect to an improvement of payoffs and competitiveness against the next-period opponent behavior. That means the player convinces that one strategy will facilitate positive feedback, and insists on employing the strategy to train itself with the expectation of improving the future competitiveness. Generally, this kind of self-training is associated with training methods and duration, and decision makers are usually not completely committed to just one set of ideas, or to just one way of behaving, and the real story is that several systems of ideas, or several possible ways of

behaving are present in their minds simultaneously. Which of these predominate, and which are given less attention, depends on the experiences of the individual (Borgers and Sarin 1997). Thus, we let the operator perform under a given probability P_2 . If the belief learning is expected to run many times, assign P_2 a bigger value, otherwise P_2 is set to a small one. Different belief learning strategies can be designed for difference problems. If some characteristic of the solution need to be emphasized, the preferential knowledge can be used to specify a belief learning algorithm. Some local optimization methods can also be added into this section. It is very convenient for people to implement what they think of. Figure 3 is a kind of belief learning procedure for real-valued presentation problems. The offspring replaces its parent only when the offspring is better than its parent so as to implement the elitist conservation strategy.

In particular, the belief learning operator is different from the belief space of a cultural algorithm (Reynolds 1999), which is divided into distinct categories representing different domains of knowledge that the population has of the search space. The belief space is updated after each iteration by the best individuals of the population.

Update players set

We believe the existence of the parent is justified and competitive; we infer the offspring has the capability to survive if the offspring is stronger or better than its parents. Based on this, the GameEA uses a simple strategy to update the players' set, which is that the offspring replaces its parent and inherits the parent's fortune like the payoffs and behavior only when the offspring is better than its parent so as to implement the elitist conservation strategy and to facilitate global convergence. For each repeat loop (lines 5–12 of Figure 1), the new player does not shy away from other players' challenges; the game is

```

Procedure beliefLearning( $I_i$ ) // for real-valued presentation
Begin
1   $r_1 := \text{rand}(0, n-1)$ ,  $\beta := \text{random}()$ 
2   $\Delta :=$  difference of maximum and minimum value of  $r_1$ th dimension
3  if ( $\beta < 0.5$ ) then  $\tau = (2\beta)^{1/21} - 1$ 
4  else  $\tau = 1 - (2-2\beta)^{1/21}$ 
5   $I_{i,\text{gen}}[r_1] = I_{i,\text{gen}}[r_1] + \tau * \Delta$ 
6  if the value of  $I_{p,\text{gen}}[r_1]$  out of the given ranges
7  then assign a required random value to  $I_{p,\text{gen}}[r_1]$ 
end

```

Figure 3. Pseudo-code for the belief learning operator for the real-valued presentation.

all around for all players regardless of whether the individual is experienced or inexperienced.

Performance comparison and experimental results

Test problems

In order to check the performance, C++ language was employed to implement GameEA. Our programming platform is the.net, and the PC is configured to AMD Phenom™ II X4 810 CPU (2.59 GHz), 2 GB of RAM. We also have chosen four kinds of algorithms: Standard Genetic Algorithm (StGA) (Holland 1975), Island-model Genetic Algorithms (IMGA) (Alba and Tomassini 2002), finding multimodal solutions using restricted tournament selection (RTS)(Harik 1995), and Dual-Population Genetic Algorithm for adaptive diversity control (Park and Ryu 2010), which are famous for their distinguished performances with different characteristics and were used for comparing the performances. IMGA is a typical example of multi-population GAs, evolves two or more subpopulations, and uses periodic migration for the exchange of information between the subpopulations. RTS employs a crowding method, which is somewhat different from the standard crowding. DPGA employs two populations, in which the main population evolves to find a good solution to the given problem and the reserve population evolves to provide controlled diversity to the main population.

To focus on the searching accuracy and stability, 13 benchmark functions f_1 – f_{13} shown in Table 2 have been considered in our experiments on real-valued representation, which are widely adopted for comparing the capabilities of evolutionary algorithms. The functions f_1 – f_7 are uni-modal distribution function with one peak within the entire given domain, and the functions f_8 – f_{13} are multimodal functions with a number of local optima in the searching space. The column n in Table 2 indicates the dimensions used.

Experimental setup

Specifically, we continue to use the parameters and strategies that StGA, IMGA, RTS, and DPGA have used. IMGA has multiple subpopulations, each of which evolves separately. RTS uses a crowding method, which is somewhat different from the standard crowding, which selects parents randomly and lets the latest offspring replace the most similar one by using a specific strategy. DPGA also uses an additional population as a reservoir of diversity. Additional details of these algorithms can be found in other studies (Alba and Tomassini 2002; Harik 1995; Holland 1975; Park and Ryu 2010).

Table 2. Test functions used for the experiments.

No.	Name	n	Function	Range
f_1	Sphere	30	$f(x) = \sum_{i=1}^n x_i^2$	$x_i \in [-100, 100]$
f_2	Schwefel 2.22	30	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$x_i \in [-10, 10]$
f_3	Schwefel 2.21	30	$f(x) = \max\{ x_i , 1 \leq i \leq n\}$	$x_i \in [-100, 100]$
f_4	Rosenbrock	30	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$x_i \in [-30, 30]$
f_5	Step	30	$f(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	$x_i \in [-100, 100]$
f_6	Noisy Quartic	30	$f(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	$x_i \in [-1.28, 1.28]$
F_7	Goldstein-price	2	$f(x) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))g(x)$ $g(x) = 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$	$x_i \in [-2, 2]$
F_8	Branin	2	$f(x) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	$x_1 \in [-5, 10]$ $x_2 \in [0, 15]$
F_9	Six-hump camelback	2	$f(x) = 4x_1^2 - 2.1x_1^4 + x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$x_i \in [-5, 5]^n$
f_{10}	Rastrigin	30	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$x_i \in [-5.12, 5.12]$
f_{11}	Griewank	30	$f(x) = 1 + \frac{\sum_{i=1}^n (x_i^2)}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$x_i \in [-600, 600]$
f_{12}	Schwefel 2.26	30	$f(x) = -\sum_{i=1}^n (x_i \sin \sqrt{ x_i })$	$x_i \in [-500, 500]$
f_{13}	Ackley	30	$f(x) = 20 + e - 20 \exp\left(-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^2)}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$	$x_i \in [-32, 32]$

The players size N is set to 50 for GameEA, the payoffs weight W_1 , losses weight W_2 , learning probability P_1 , mutation probability P_2 , and the speculative probability P_3 are set to 0.9, 0.01, 0.9, 0.1, and 0.1, respectively. For each function, the maximum iteration of GameEA is a set of homogeneous iterations with the compared algorithm. For example, the parent size and the offspring size for DPGA are, respectively, set to 50 and 100 for function f_1 , and the iterations are set to 1500, then its maximum iteration is 225000, so the maximum generation is set to 4500 with the players' size of 50. Table 3 summarizes the generations used for each function, which are indicated in the *Maximum generation* column of the table. The following sections present the statistical experimental results obtained for the functions mentioned above.

Experimental results analysis

For each test problem, 50 independent experiments were executed and the averaged results are shown in Table 3. The *avg* and *stdev* columns stand for the average and standard deviation of the results, respectively. Table 3 also provides the results of the *t*-test at the confidence level of 5% between GameEA and each of the other algorithms. To facilitate this description, we use “+” to indicate that GameEA is significantly better than the compared algorithm, “-” represents that GameEA is obviously worse than the other four algorithms, and “ \approx ” indicates that there are no statistically significant differences between GameEA and the compared algorithms.

For the uni-modal functions f_1 , f_2 , and f_4 , GameEA obtains the best results with an average result of 4.33E-96, 1.52E-66, and 0.0374, respectively. The DPGA shows the second best results, with an average result of 1.47E-52 for function f_1 , and the RTS is the third on the list. The StGA convergence accuracy is slightly worse than IMGA for functions f_1 and f_2 , but IMGA is worse than StGA for function f_4 . For the function f_4 , RTS with an average of 1.391 comes second, but GameEA's average is less than 3% of RTS. For the function f_6 , GameEA is significantly better than StGA and DPGA. GameEA is similar to IMGA and RTS, but it holds a smaller standard deviation. For the functions f_5 and f_7 , all the above-mentioned algorithms converge to the global minimum.

To make a careful observation of the results on the highly multimodal functions f_8 - f_{13} , which have a number of local optima, all the algorithms converge to the global minimum on functions f_8 and f_9 . RTS, DPGA, and GameEA show no significant differences with the convergence of global optima in each independent trial on function f_{10} , which means all the algorithms are capable of solving such problems.

GameEA is the best, with the best stability convergence in 50 independent trials for functions f_{11} - f_{13} , which shows good search-ability, stability, and

Table 3. Fifty independent experimental statistics results based on StGA, IMGA, RTS, DPGA, and GameEA using real-valued representation on functions f_1-f_{13} .

	Maximum generation	Optimum solution	StGA		IMGA		RTS		DPGA		GameEA		t-Test(GameEA Versus)			
			avg	stdev	avg	stdev	avg	stdev	avg	stdev	avg	stdev	StGA	IMGA	RTS	DPGA
f_1	1500	0	6.12E-34	1.27E-38	2.04E-34	5.23E-34	7.90E-43	1.74E-42	1.47E-52	4.17E-52	4.33E-96	2.79E-95	+	+	+	+
f_2	2000	0	3.32E-29	1.78E-28	6.40E-32	1.36E-31	7.18E-37	6.19E-37	5.19E-45	7.90E-45	1.52E-66	4.61E-66	+	+	+	+
f_3	5000	0	7.00E-15	1.37E-14	4.28E-06	3.64E-06	1.54E-05	1.91E-05	3.12E-09	1.18E-08	7.85E-04	2.26E-03	-	-	-	-
f_4	20000	0	5.454	3.662	5.554	4.522	1.391	1.211	3.047	3.558	0.0374	0.264	+	+	+	+
f_5	1500	0	0	0	0	0	0	0	0	0	0	0	≈	≈	≈	≈
f_6	3000	0	1.37E-02	3.24E-03	7.52E-03	2.24E-03	1.82E-03	4.56E-04	1.46E-02	3.93E-03	7.66E-03	1.82E-03	+	+	+	+
f_7	100	3	3	0	3	0	3	0	3	0	3	0	≈	≈	≈	≈
f_8	100	0.398	0.398	0	0.398	0	0.398	0	0.398	0	0.398	0	≈	≈	≈	≈
f_9	100	-1.032	-1.032	0	-1.032	0	-1.032	0	-1.032	0	-1.032	0	≈	≈	≈	≈
f_{10}	5000	0	11.809	2.369	0.358	0.746	0	0	0	0	0	0	+	+	+	+
f_{11}	2000	0	1.63E-03	3.91E-03	3.54E-03	7.73E-03	2.07E-03	5.31E-03	1.28E-03	3.31E-03	0	0	+	+	+	+
f_{12}	9000	-12569.4866	-11195.1	284.5	-12008.1	284.9	-12443.9	142.4	-12550.5	43.9	-12569.4866	0	+	+	+	+
f_{13}	1500	0	3.55E-15	0	4.69E-15	1.67E-15	5.26E-15	1.79E-15	3.55E-15	0	6.84E-16	1.30E-15	+	+	+	+

robustness. It is a remarkable fact that StGA and DPGA did not converge to global optimal solution with the same standard deviation of zero in all the trials for function f_{13} , which indicates that those algorithms have been stalled at the local optimal points, whereas GameEA escapes and continually evolves to the global minimum. We also determined the final solutions of GameEA about f_{13} ; although its averages of optimum are not perfect, we found that there were 39 parts in 50 to obtain the global optimum.

For function f_3 , GameEA is obviously worse than the other four algorithms. When we studied the 50 experimental outputs, it was found that the best solution is set to 4.43E-12 and the worst optimum is set to 0.0122, which means that the final results share a large span, which leads to a poor average and standard deviation. We expand the generations up to 20000 in order to check the sustainability of GameEA, and GameEA reported an average of 1.008E-15 with a standard deviation of 2.5E-12. The statistics results based on 50 independent trials obtained by GameEA for function f_3 are shown in Figure 4. According to the average best-so-far curve shown in Figure 4(a), GameEA can escape from the local optima and keep improving the precision, namely that the best-to-so-far is improved continually along with the increase of the evolution, which partially indicates that the GameEA has a predominance sustainability of evolution.

To observe the evolution of the best-so-far optimal solution, we have checked the tendency of averages of uni-modal function f_4 and multimodal function f_{12} , and its average best-so-far curves for those functions are shown in Figure 5 (the data of compared algorithms is from Park and Ryu (2010)). For uni-modal function f_4 , GameEA shows the fastest convergence and provides the best results at the very early evolution, followed by RTS and then DPGA; however, it also indicates the risk of getting stuck in local optima and cannot escape from them. For uni-modal function f_{12} , GameEA is the champion and DPGA comes next, although the convergence curves of all the algorithms are not that different at the last stage.

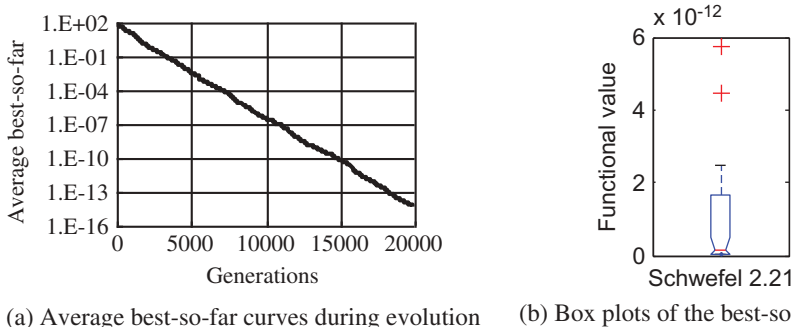


Figure 4. Statistics results based on 50 independent runs obtained by GameEA for function f_3 .

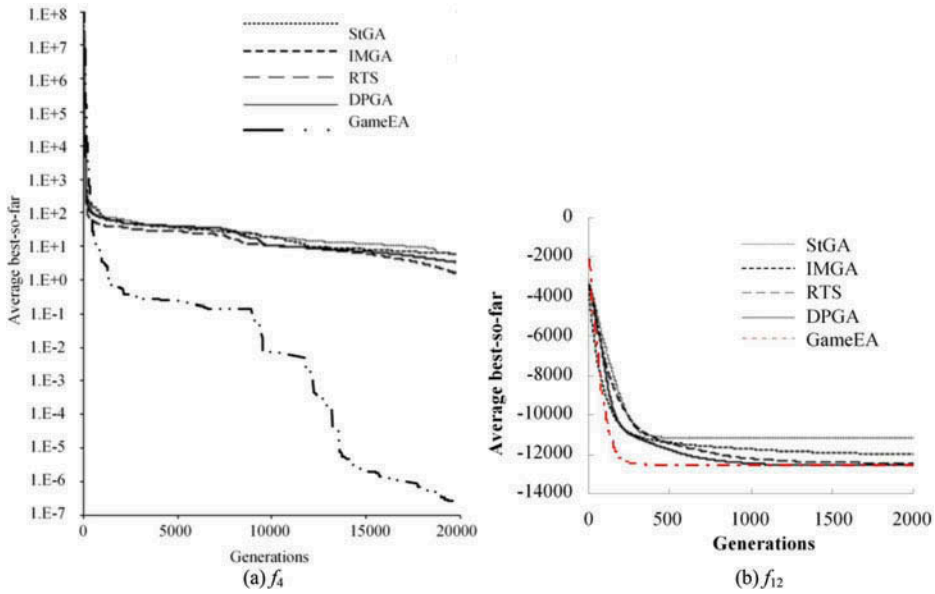


Figure 5. Average best-so-far curves for functions during evolution.

In conclusion, the summarized results of statistical significance testing to compare GameEA with each of the other four algorithms are shown in Table 4. GameEA performs better than the other algorithms for all of the multimodal problems. For the uni-modal functions, GameEA loses only one time against StGA, IMGAs, RTS, and DPGA. It can be concluded that GameEA has better performance, such as the stability, robustness, and accuracy of solution, compared with algorithms on function optimization problems.

Additional experiments with different players' size

This section compares the convergence of GameEA with different players' size to understand whether this parameter has significant effect on the convergence. The experimental settings are the same as the previous section, except the players size N and the maximum generation. The generation is set to $3E+6$, and the N is set to 20, 50, 100, 200, 300, and 500, respectively. In this experiment, we focus only on the players' size as the factor relating to the

Table 4. Summary of statistically significance testing.

problems	StGA		IMGAs		RTS		DPGA	
	+	-	+	-	+	-	+	-
Unimodal functions (f_1-f_7)	4	1	3	1	3	1	4	1
Multimodal functions (f_8-f_{13})	4	0	4	0	3	0	3	0
total	8	1	7	1	6	1	7	1

convergence of GameEA, which is considered as six levels of the factor. Thirty independent trials were executed, and Table 5 shows the experimental results of GameEA with different players' size. Objectively, even given the same inputs and parameters, due to the random factors (game randomness, random number, etc.), GameEA might converge to different solutions for one problem. To conclude whether the algorithm has performance differences of searching for the perfect solution between different population sizes is to distinguish what is the primary factor leading to the differences between the different parameters and the random error. This problem can boil down to a question that whether the six kinds of samples have the same distribution, namely to check whether the normal population means are equal among the observation groups with the same variance. Analysis of variance (ANOVA) (Anscombe 1948) was used to analyze the differences among group means.

Experiments and analysis

Table 6 shows the results of the homogeneity test of variances and one-sample Kolmogorov–Smirnov for Rosenbrock function with different parameters. The Levene statistic is 20.046, the degrees of freedom df_1 and df_2 are 5 and 174, respectively, and the two-tailed significance probability is 0.0001. The tests show that the overall sample meets the requirements of homogeneity of variance. Obviously, Table 6(b) provides evidence of the normality of each group of data. Using the same method, it can be proved that the data of the other test problems also meet the requirements of ANOVA, and hence the other detailed data are omitted.

We assume that this is a null hypothesis H_0 , namely that the convergence accuracy has no significant difference with different population size N on the significance level of 0.05 ($\alpha = 0.05$). Table 7 shows the results of ANOVA and the homogeneous subset for different groups of data obtained by GameEA with different parameters. ANOVA gives an overall test for the difference between the means of six groups. According to the principle, the within-groups only indicate the stochastic error caused by random variation, and the between-groups reflect both stochastic error and systematic error. If the population/players size has no effect on the convergence, the between-groups contain only random errors among the tested six groups, and then the ratio of between-groups and within-groups approximates 1. Otherwise, the between-groups contain not only random variance but also systematic error, yet it is obvious that the ratio of between-groups and within-groups is greater than 1, which implies the population size has a significant impact on the final results when this ratio reached a certain level.

As can be seen in Table 7, the observed value of F is found by dividing the between-group variance by the within-group variance, and all $F > F_{\alpha=0.05}$, the null hypothesis H_0 is rejected, which means the convergence accuracy has

Table 5. Experimental results of GameEA with different players size.

No.	N = 20					N = 50					N = 300				
	avg	stdev	gen-avg	time/s	time/s	avg	stdev	gen-avg	time/s	time/s	avg	stdev	gen-avg	time/s	
f ₁	0	0	21087	0.2256	1.0023	0	0	18139	1.0023	1.27E-156	0	6666	4.8285		
f ₂	1.56E-226	0	82369	5.0376	7.2957	3.79E-211	0	39991	7.2957	1.05E-80	1.02E-80	6666	7.3454		
f ₃	3.68E-14	8.63E-14	99982	6.4724	6.3858	2.62E-07	1.11E-06	39981	6.3858	4.24E-05	2.64E-05	6652	6.3976		
f ₄	0.680864	1.73E+00	87961	11.1543	11.3292	0.036319	1.96E-01	35575	11.3292	0.000841	4.44E-04	6621	14.2034		
f ₅	0	0	309	0.0001	0.0004	0	0	256	0.0004	0	0	222	0.0106		
f ₆	9.05E-04	3.60E-04	84979	7.1923	7.958	1.60E-03	4.72E-04	35801	7.958	6.40E-03	1.35E-03	6103	8.3175		
f ₇	7.194243	1.55E+01	17011	0.1114	0.2626	3	5.11E-12	10606	0.2626	3	0	1827	0.2851		
f ₈	0.397887	2.26E-16	17457	0.114	0.092	0.397887	2.26E-16	6376	0.092	0.397887	2.26E-16	1555	0.1999		
f ₉	-1.031628	4.52E-16	2632	0.003	0.001	-1.031628	4.52E-16	616	0.001	-1.031628	4.52E-16	410	0.0162		
f ₁₀	0	0	1017	0.0008	0.0034	0	0	861	0.0034	0	0	782	0.0997		
f ₁₁	0	0	3989	0.0133	0.0051	0	0	999	0.0051	0	0	883	0.1453		
f ₁₂	-1.26E+04	5.55E-12	3979	0.0148	0.0876	-1.26E+04	5.55E-12	3877	0.0876	-1.26E+04	5.55E-12	3720	2.8881		
f ₁₃	9.33E-16	2.10E-30	1538	0.0018	0.0079	8.29E-16	1.96E-30	1300	0.0079	1.04E-16	3.22E-31	1186	0.2339		
No.	N = 100					N = 200					N = 500				
	avg	stdev	gen-avg	time/s	time/s	avg	stdev	gen-avg	time/s	time/s	avg	stdev	gen-avg	time/s	
f ₁	0	0	17195	3.5933	4.9305	2.21E-215	0	9998	4.9305	7.63E-97	9.93E-97	3999	4.8131		
f ₂	1.33E-171	0	19996	7.3231	7.3424	3.66E-111	1.69E-110	9999	7.3424	7.95E-50	5.29E-50	3999	7.2306		
f ₃	1.13E-05	5.33E-05	19985	6.3918	6.4471	8.40E-08	8.96E-08	9985	6.4471	9.53E-03	2.60E-03	3987	6.3436		
f ₄	0.000036	1.48E-04	19685	13.8408	13.899	0.000191	1.25E-04	9766	13.899	0.00551	1.92E-03	3972	14.3059		
f ₅	0	0	237	0.0013	0.005	0	0	228	0.005	0	0	218	0.0285		
f ₆	2.54E-03	7.44E-04	18020	8.3217	8.242	4.65E-03	1.03E-03	9120	8.242	9.98E-03	1.96E-03	3654	8.297		
f ₇	3	8.76E-12	6824	0.4408	0.2408	3	0	2486	0.2408	3	0	1285	0.3906		
f ₈	0.397887	2.26E-16	2759	0.0747	0.1158	0.397887	2.26E-16	1781	0.1158	0.397887	2.26E-16	1286	0.3787		
f ₉	-1.031628	4.52E-16	529	0.003	0.0057	-1.031628	4.52E-16	366	0.0057	-1.031628	4.52E-16	405	0.0441		
f ₁₀	0	0	809	0.0118	0.0452	0	0	788	0.0452	0	0	772	0.2683		
f ₁₁	0	0	937	0.0182	0.0681	0	0	901	0.0681	0	0	875	0.3985		
f ₁₂	-1.26E+04	5.55E-12	3754	0.3414	1.2487	-1.26E+04	5.55E-12	3667	1.2487	-1.26E+04	5.55E-12	3664	7.8013		
f ₁₃	4.14E-16	1.16E-30	1239	0.0309	0.1039	0	0	1188	0.1039	0	0	1182	0.6433		

Table 6. Results of homogeneity test of variances and one-sample Kolmogorov–Smirnov for function Rosenbrock with different parameters.

(a) Test of Homogeneity of variances			
Levene Statistic	df1	df2	Sig.
20.046	5	174	0.0001
(b) One-sample Kolmogorov–Smirnov test			
Normal Parameters ^{a,b}	Sample Size	180	
	Mean	-9.987E-1	
	Std. Deviation	2.5003E-3	
Most Extreme Differences	Absolute	0.443	
	Positive	0.443	
	Negative	-0.299	
	Kolmogorov-Smirnov Z	5.946	
	Asymp. Sig. (2-tailed)	0.0001	

^aTest distribution is Normal. b. Calculated from data.

significant difference with different population size N on the significance level of 0.05. Moreover, the multiple comparisons of homogeneous subset means is another kind of expression of the multiple comparison of group means, and there is no significant difference among the data sets belonging to the same homogeneous subset. In Table 7(a) there are two homogeneous subsets, the probability of the hypothesis – that the group means are equal when the population sizes are set to 50, 100, 200, 300, and 500 on the significance level α – is 0.864 for the first subset, and this probability is bigger than $\alpha = 0.05$, so that the hypothesis is rejected, which means the group balance variance is good; and the other subset only has one group data, and there is no possibility of discussing its balance. It can be concluded the population size belonging to the two homogeneous subsets have significant impact on GameEA's convergence. So to consider this conclusion, we can infer that the population size N corresponding to the first homogeneous subset is more suitable for GameEA to solve the function f_4 . Similarly, the population size belonging to the set of {20, 50, 100, 200, and 300} will lead GameEA to improve the performance for function f_3 . For function f_6 , the experiment data were divided into six homogeneous subsets on the significant level 0.05, and the average variance increases with the increase of the value of N , which indicates the average convergence accuracy decreases with the increase of N . We modified the significance level to 0.01, and then those groups with $N = 20$ or 50 or 100 remain at the same homogeneous subset. Based on the observed factors, we make a conclusion: GameEA with the population size $N = 50$ or 100 will give more satisfactory performance when adopting it to optimize the uni-modal function problem.

For multimodal function f_{14} , there are three homogeneous subsets, namely $\{N|200, 500, 300, \text{ and } 100\}$, $\{N|100, 50\}$, and $\{N|50, 20\}$. The observed value of F_α of each subset is 0.13, 0.96, and 0.676, respectively. Taking this into consideration and referring to Table 5, it can be suggested that GameEA

Table 7. Summary of ANOVA and a homogeneous subset for six group data obtained by GameEA with different parameters.

	Sum of Squares	df	Mean Square	F	F _α	
(a) f₄: Rosenbrock						
Between Groups	11.329	5	2.266	4.474	0.001	
Within Groups	88.122	174	0.506			
Total	99.450	179				
Homogeneous subset (Duncan ^a)						
Group	Subset for α= 0.05					
	1			2		
100	3.5981E-5					
200	1.9145E-4					
300	8.4040E-4					
500	5.5087E-3					
50	3.6319E-2					
20			6.8086E-1			
F _α	0.864		1.000			
(b) f₃: Schwefel 2.21						
Between Groups	2.26E-3	5	4.53E-4	402.593	1.3E-93	
Within Groups	1.96E-4	174	1.13E-6			
Total	2.46E-3	179				
	1			2		
20	3.6772E-14					
200	8.3989E-8					
50	2.6248E-7					
100	1.1265E-5					
300	4.2420E-5					
500		9.5290E-3				
F _α	0.893	1.000				
(c) f₆: Noisy Quartic						
Between Groups	1.76E-3	5	3.52E-4	276.416	8.7E-81	
Within Groups	2.22E-4	174	1.27E-6			
Total	1.98E-3	179				
	1	2	3	4	5	6
20	9.053E-4					
50		1.598E-3				
100			2.545E-3			
200				4.649E-3		
300					6.398E-3	
500						9.981E-3
F _α	1.000	1.000	1.000	1.000	1.000	1.000
(d) f₁₄: Ackley						
Between Groups	2.62E-29	5	5.24E-30	5.684	7.0E-5	
Within Groups	1.61E-28	174	9.23E-31			
Total	1.87E-28	179				
	1		2		3	
200	0					
500	0					
300	1.0367E-16					
100	4.147E-16		4.147E-16			
50			8.293E-16		8.293E-16	
20					9.330E-16	
F _α	0.130		0.096		0.676	

^aUses Harmonic Mean Sample Size = 30.

will show predominance performance for the multimodal function optimization when its population size is $N = 200$.

Summary of the analysis

According the statistical analysis results, under the significance level $\alpha = 0.05$, players size has a crucial impact on the performance of GameEA. The size affects the convergence accuracy, capability of solving the deceptive problems, and the exploration ability of insight into the global optimal solution. If the size is too small, there will not be enough schemas to be exploited (Dejong and Spears 1991), resulting into a premature; if the population size is too large, it may require unnecessary large computational resources and result in an extremely long running time. When employing GameEA to solve problems, setting the players' size depends on the complexity of the problem. It is suggested that the players' size is set to 50 or 100 for uni-modal function and to 200 for multimodal function.

Conclusion

Development of GameEA was motivated by studies that attempt to develop simpler and more efficient game evolutionary algorithms. Unlike many search algorithms, GameEA has a payoff expectation mechanism, and the player makes a decision by evaluating the payoff expectation. Moreover, GameEA's strategy for generating a trial population includes Imitation and Belief learning operators. According to the GameEA's elite conservation strategy for updating the players set, the offspring replaces its parent and inherits the parents' information only when the offspring is better than its parent. Experiments have been conducted on various benchmark functions (unimodal functions and multimodal function). We have compared the performance of the GameEA with that of StGA, IMGA, RTS, and DPGA. We have also analyzed some additional experiments to determine the influence of player size on evolution. In future work, we will conduct experiments with more intensive problems, including multi-objective optimization problems involving more complicated applications.

Funding

The authors would like to thank the financial support provided by the National Natural Science Foundation of China under Grants 61640209 and 51475097, Foundation for Distinguished Young Talents of Guizhou Province under Grant QKHRZ[2015]13, Science and Technology Foundation of Guizhou Province under Grants JZ[2014]2004, JZ[2014]2001, ZDZX[2013]6020, LH[2016]7433, ZDZX[2014]6021, and National Key Technology Support Program of China under Grant 2012BAH62F00.

ORCID

Guanci Yang  <http://orcid.org/0000-0001-8761-5195>

References

- Alba, E., and M. Tomassini. 2002. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 6 (5):443–62. doi:10.1109/TEVC.2002.800880.
- Anscombe, F. J. 1948. The validity of comparative experiments. *Journal of the Royal Statistical Society. Series A (General)* 111 (3):181–211. doi:10.2307/2984159.
- Borgers, T., and R. Sarin. 1997. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory* 77 (1):1–14. doi:10.1006/jeth.1997.2319.
- Bulo, S. R., A. Torsello, and M. Pelillo. 2009. A game-theoretic approach to partial clique enumeration. *Image and Vision Computing* 27 (7):911–22. doi:10.1016/j.imavis.2008.10.003.
- Colman, A. M. 2003. Cooperation, psychological game theory, and limitations of rationality in social interaction. *Behavioral and Brain Sciences* 26 (2):139. doi:10.1017/S0140525X03000050.
- Dejong, K. A., and W. M. Spears. 1991. An analysis of the interacting roles of population-size and crossover in genetic algorithms. *Lecture Notes in Computer Science* 496:38–47.
- Dorigo, M., and L. M. Gambardella. 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions On* 1 (1):53–66. doi:10.1109/4235.585892.
- Fontanari, J. F., and L. I. Perlovsky. 2008. A game theoretical approach to the evolution of structured communication codes. *Theory in Biosciences* 127 (3):205–14. doi:10.1007/s12064-008-0024-1.
- Friedman, D., S. Huck, R. Oprea, and S. Weidenholzer. 2015. From imitation to collusion: Long-run learning in a low-information environment. *Journal of Economic Theory* 155:185–205. doi:10.1016/j.jet.2014.10.006.
- Ganesan, T., I. Elamvazuthi, and P. Vasant. 2015. Multiobjective design optimization of a nano-CMOS voltage-controlled oscillator using game theoretic-differential evolution. *Applied Soft Computing* 32:293–99. doi:10.1016/j.asoc.2015.03.016.
- Harik, G. R. 1995. Finding multimodal solutions using restricted tournament selection. *Paper Read at Proceedings of the 6th International Conference on Genetic Algorithms* 24–31.
- Hausknecht, M., J. Lehman, R. Miikkulainen, and P. Stone. 2014. A neuroevolution approach to general Atari game playing. *IEEE Transactions on Computational Intelligence and AI in Games* 6 (4):355–66. doi:10.1109/TCAIG.2013.2294713.
- He, J., and X. Yao. 2005. A game-theoretic approach for designing mixed mutation strategies. In *LECTURE NOTES IN COMPUTER SCIENCE*, eds. L. Wang, K. Chen, and Y. S. Ong, 279–88. BERLIN: SPRINGER-VERLAG BERLIN.
- Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press.
- Hu, H., and H. W. Stuart Jr. 2002. An epistemic analysis of the Harsanyi transformation. *International Journal of Game Theory* 30 (4):517–25. doi:10.1007/s001820200095.
- Jiang, G., S. Shen, K. Hu, L. Huang, H. Li, and R. Han. 2015. Evolutionary game-based secrecy rate adaptation in wireless sensor networks. *International Journal of Distributed Sensor Networks* 11 (3):975454. doi:10.1155/2015/975454.
- Karaboga, D., and B. Basturk. 2007. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization* 39 (3):459–71. doi:10.1007/s10898-007-9149-x.

- Koh, A. 2012. An evolutionary algorithm based on Nash dominance for equilibrium problems with equilibrium constraints. *Applied Soft Computing* 12 (1):161–73. doi:[10.1016/j.asoc.2011.08.056](https://doi.org/10.1016/j.asoc.2011.08.056).
- Kontogiannis, S., and P. Spirakis. 2005. Evolutionary games: An algorithmic view. In *Lecture Notes in Computer Science*, eds. O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. VanMoorsel, and M. VanSteen, 97–111. Berlin: Springer-Verlag Berlin.
- Lee, D., L. F. Gonzalez, J. Periaux, K. Srinivas, and E. Onate. 2011. Hybrid-game strategies for multi-objective design optimization in engineering. *Computers & Fluids* 47 (1):189–204. doi:[10.1016/j.compfluid.2011.03.007](https://doi.org/10.1016/j.compfluid.2011.03.007).
- Li, X., Z. Shao, and J. Qian. 2002. An optimizing method based on autonomous animals: Fish-swarm algorithm. *System Engineering Theory and Practice* 22 (11):32–38.
- Liu, W., and X. Wang. 2008. An evolutionary game based particle swarm optimization algorithm. *Journal of Computational and Applied Mathematics* 214 (1):30–35. doi:[10.1016/j.cam.2007.01.028](https://doi.org/10.1016/j.cam.2007.01.028).
- Maass, W., T. Natschläger, and H. Markram. 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* 14 (11):2531–60. doi:[10.1162/089976602760407955](https://doi.org/10.1162/089976602760407955).
- Madani, K., and M. Hooshyar. 2014. A game theory-reinforcement learning (GT-RL) method to develop optimal operation policies for multi-operator reservoir systems. *Journal of Hydrology* 519 (A):732–42. doi:[10.1016/j.jhydrol.2014.07.061](https://doi.org/10.1016/j.jhydrol.2014.07.061).
- Mejia, M., N. Pena, J. L. Munoz, O. Esparza, and M. A. Alzate. 2011. A game theoretic trust model for on-line distributed evolution of cooperation in MANETs. *Journal of Network and Computer Applications* 34 (1):39–51. doi:[10.1016/j.jnca.2010.09.007](https://doi.org/10.1016/j.jnca.2010.09.007).
- Misra, S., and S. Sarker. 2015. Priority-based time-slot allocation in wireless body area networks during medical emergency situations: An evolutionary game-theoretic perspective. *IEEE Journal of Biomedical and Health Informatics* 19 (2):541–48. doi:[10.1109/JBHI.2014.2313374](https://doi.org/10.1109/JBHI.2014.2313374).
- Nax, H. H., and M. Perc. 2015. Directional learning and the provisioning of public goods. *Scientific Reports* 5 (8010). doi:[10.1038/srep08010](https://doi.org/10.1038/srep08010).
- Neshat, M., G. Sepidnam, M. Sargolzaei, and A. N. Toosi. 2014. Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artificial Intelligence Review* 42 (4):965–97. doi:[10.1007/s10462-012-9342-2](https://doi.org/10.1007/s10462-012-9342-2).
- Niyato, D., E. Hossain, and H. Zhu. 2009. Dynamics of multiple-seller and multiple-buyer spectrum trading in cognitive radio networks: A game-theoretic modeling approach. *IEEE Transactions on Mobile Computing* 8 (8):1009–22. doi:[10.1109/TMC.2008.157](https://doi.org/10.1109/TMC.2008.157).
- Nowak, M. A., and K. Sigmund. 2004. Evolutionary dynamics of biological games. *Science* 303 (5659):793–99. doi:[10.1126/science.1093411](https://doi.org/10.1126/science.1093411).
- Park, T., and K. R. Ryu. 2010. A dual-population genetic algorithm for adaptive diversity control. *IEEE Transactions on Evolutionary Computation* 14 (6):865–84. doi:[10.1109/TEVC.2010.2043362](https://doi.org/10.1109/TEVC.2010.2043362).
- Periaux, J., H. Q. Chen, B. Mantel, M. Sefrioui, and H. T. Sui. 2001. Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. *Finite Elements in Analysis and Design* 37 (5):417–29. doi:[10.1016/S0168-874X\(00\)00055-X](https://doi.org/10.1016/S0168-874X(00)00055-X).
- Qin, Z., T. Wan, Y. Dong, and Y. Du. 2015. Evolutionary collective behavior decomposition model for time series data mining. *Applied Soft Computing* 26:368–77. doi:[10.1016/j.asoc.2014.09.036](https://doi.org/10.1016/j.asoc.2014.09.036).
- Ram K. M., R. Jayasree, and D. Suresh. 2014. Applications of Game Theory. Academic Foundation, New Delhi.
- Reynolds, R. G. 1999. Cultural algorithms: Theory and applications. In *New ideas in optimization*, eds. D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price, 367–78. UK: McGraw-Hill Ltd.

- Rosenstrom, T., P. Jylha, L. Pulkki-Raback, M. Holma, I. T. Raitakari, E. Isometsa, and L. Keltikangas-Jarvinen. 2015. Long-term personality changes and predictive adaptive responses after depressive episodes. *Evolution and Human Behavior* 36 (5):337–44. doi:10.1016/j.evolhumbehav.2015.01.005.
- Shi Y. 2011. Brain Storm Optimization Algorithm. In: Tan Y., Shi Y., Chai Y., Wang G. (eds) *Advances in Swarm Intelligence. ICSI 2011. Lecture Notes in Computer Science*, vol 6728. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21515-5_36.
- Simon, A., G. Jacob, and H. Charles. 1999. Stochastic Game Theory: Adjustment to Equilibrium Under Noisy Directional Learning, Virginia Economics Online Papers, University of Virginia, Department of Economics, <https://EconPapers.repec.org/RePEc:vir:virpap:327>.
- Spiliopoulos, L. 2012. Pattern recognition and subjective belief learning in a repeated constant-sum game. *Games and Economic Behavior* 75 (2):921–35. doi:10.1016/j.geb.2012.01.005.
- Stahl, D. O. 2000. Rule learning in symmetric normal-form games: Theory and evidence. *Games and Economic Behavior* 32 (1):105–38. doi:10.1006/game.1999.0754.
- Szubert, M., W. Jaskowski, and K. Krawiec. 2013. On scalability, generalization, and hybridization of coevolutionary learning: A case study for Othello. *IEEE Transactions on Computational Intelligence and AI in Games* 5 (3):214–26. doi:10.1109/TCAIG.2013.2258919.
- Tembine, H., E. Altman, R. El-Azouzi, and Y. Hayel. 2010. Evolutionary games in wireless networks. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics* 40 (3SI):634–46. doi:10.1109/TSMCB.2009.2034631.
- Wei, G., A. V. Vasilakos, Y. Zheng, and N. Xiong. 2010. A game-theoretic method of fair resource allocation for cloud computing services. *Journal of Supercomputing* 54 (2):252–69. doi:10.1007/s11227-009-0318-1.
- Wikipedia. 2015. *Game theory*.
- Woldemariam, K. M., and G. G. Yen. 2010. Vaccine-enhanced artificial immune system for multimodal function optimization. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics* 40 (1):218–28. doi:10.1109/TSMCB.2009.2025504.
- Yang, X. 2010. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation* 2 (2):78–84. doi:10.1504/IJBIC.2010.032124.
- Yang, X., and S. Deb. 2009. *Cuckoo search via Lévy flights*.. Paper read at 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE: Coimbatore 210-214. doi: 10.1109/NABIC.2009.5393690
- Yang, G., Y. Wang, S.B. Li, and Q. Xie. 2016. Game evolutionary algorithm based on behavioral game theory. *Journal of Huazhong University of Science and Technology (Natural Science Edition)* 44(7): 68–73.
- Yang G., A. Zhang, S. Li, W. Yang, Y. Wang, Q. Xie, and L. He. 2017. Multi-objective evolutionary algorithm based on decision space partition and its application in hybrid power system optimisation. *Applied Intelligence* 46 (4):827–844. doi: 10.1007/s10489-016-0864-1