Artif Life Robotics (2004) 8:174–185
DOI 10.1007/s10015-004-0308-6

ORIGINAL ARTICLE

Kwee-Bo Sim · Ji-Yoon Kim

Solution of multiobjective optimization problems: coevolutionary algorithm based on evolutionary game theory

Received and accepted: October 28, 2003

Abstract When attempting to solve multiobjective optimization problems (MOPs) using evolutionary algorithms, the Pareto genetic algorithm (GA) has now become a standard of sorts. After its introduction, this approach was further developed and led to many applications. All of these approaches are based on Pareto ranking and use the fitness sharing function to keep diversity. On the other hand, the scheme for solving MOPs presented by Nash introduced the notion of Nash equilibrium and aimed at solving MOPs that originated from evolutionary game theory and economics. Since the concept of Nash Equilibrium was introduced, game theorists have attempted to formalize aspects of the evolutionary equilibrium. Nash genetic algorithm (Nash GA) is the idea to bring together genetic algorithms and Nash strategy. The aim of this algorithm is to find the Nash equilibrium through the genetic process. Another central achievement of evolutionary game theory is the introduction of a method by which agents can play optimal strategies in the absence of rationality. Through the process of Darwinian selection, a population of agents can evolve to an evolutionary stable strategy (ESS). In this article, we find the ESS as a solution of MOPs using a coevolutionary algorithm based on evolutionary game theory. By applying newly designed coevolutionary algorithms to several MOPs, we can confirm that evolutionary game theory can be embodied by the coevolutionary algorithm and this coevolutionary algorithm can find optimal equilibrium points as solutions for an MOP. We also show the optimization performance of the co-evolutionary algorithm based on evolutionary game theory by applying this model to several MOPs and comparing the solutions with those of previous evolutionary optimization models.

K.-B. Sim (✉) · J.-Y. Kim
School of Electrical and Electronic Engineering, Chung-Ang University, 221 Huksuk-Dong, Dongjak-Ku, Seoul 156-756, Korea
Tel. +82-2-820-5319; Fax +82-2-817-0553
e-mail: kbsim@cau.ac.kr

This work was presented, in part, at the 8th International Symposium on Artificial Life and Robotics, Oita, Japan, January 24–26, 2003

Key words Multiobjective optimization problems (MOPs) · Pareto optimal set · Game theory · Nash genetic algorithm · Evolutionary stable strategy (ESS) · Coevolutionary algorithm

1 Introduction

Multiobjective optimization problems (MOPs) are met everywhere because most of the real-world problems encountered by engineers involve simultaneous optimization of several competitive objective functions.[1] For example, in the case of bridge construction, a good design is characterized by low total mass and high stiffness. However, high stiffness requires high total mass. In this problem, total mass and stiffness are competitive objective functions that need to be optimized simultaneously. Aircraft design requires simultaneous optimization of fuel efficiency, payload, and weight. A Pegasus gas turbine engine design needs to optimize the low-pressure spool speed governor.[2] These are all MOPs. Like these, traditional optimization problems attempt to simultaneously minimize the cost and maximize the fiscal return. In these and most other cases, it is unlikely that each objective would be optimized by the same parameter choices. Hence, some trade-off between the criteria is needed to ensure a satisfactory design.

In searching for solutions to these problems, we find that there is no single optimal solution but rather a set of solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered. They are generally known as Pareto-optimal solutions.[3] Although there are many approaches for solving MOPs, we bring evolutionary optimization algorithms into focus.

This section refers to previously proposed traditional approaches and evolutionary approaches. In Sect. 2, the definition of MOPs and other concepts are outlined. Section 3 explains optimization approaches based on evolutionary game theory for solving MOPs. The first is a Nash genetic algorithm (Nash GA) proposed by Sefrioui and the second is a coevolutionary optimization algorithm based on a game model, as newly proposed in this article. In the final section, we evaluate optimization performance by comparing the optimized solutions of our new coevolutionary optimization algorithm with those of other evolutionary optimization algorithms for several test problems.

Traditional approaches

Classical methods for generating the Pareto-optimal set aggregate the objective functions of MOPs into a single parameterized objective function. Then the optimizer systematically varies the parameters of this function. Several optimizations are performed in order to achieve a set of solutions that approximate the Pareto-optimal set.[1] Some representatives of this class of techniques are the weighting method,[4] the constraint method,[4] goal programming, and the min–max approach.

Evolutionary approaches: non-Pareto approaches

The first exploration for treating objective functions separately in evolutionary algorithms (EAs) was launched by Schaffer. In his dissertation,[5,6] Schaffer proposed his vector-evaluated genetic algorithm (VEGA) for searching a solution set to solve MOPs. He created VEGA to find and maintain multiple classification rules in a set covering a problem. VEGA tried to achieve this goal by selecting a fraction of the next generation using one of each of the attributes (e.g., cost, reliability).[7] Other approaches that search solutions for MOPs include those of Fourman,[8] Kursawe,[10] and Hajela and Lin.[11] However, because none of them makes direct use of the actual definition of Pareto-optimality, different nondominated individuals are generally assigned different fitness values.[2]

Evolutionary approaches: Pareto-based approaches

Goldberg first proposed Pareto-based fitness assignment approaches known as Pareto genetic algorithm (Pareto GA). The idea of this algorithm is to assign high probability to all nondominated individuals in the population. This method consists of assigning rank 1 to the nondominated individuals and removing them from contention, then finding a new set of nondominated individuals, ranked 2, and so forth. He named this the Pareto ranking.[11]

Fonseca and Fleming[12] proposed a different scheme, whereby an individual's rank corresponds to the number of individuals in the current population by which it is dominated. Therefore, nondominated individuals are assigned the same rank, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface.[14] Horn and Nafpliotis also proposed tournament selection based on Pareto dominance.[15]

Distributive search is very important in Pareto GA. The goal of Pareto GA is to search all Pareto optimal solution sets distributed along the Pareto frontier. To achieve this goal, Goldberg and Richardson introduced the concept of fitness sharing.[13] It is within the range of possibility to search for distributive solutions using the fitness sharing that makes highly fitted candidates share fitness with others in their surroundings.[7]

With the introduction of nondominated Pareto-ranking and fitness sharing, Pareto GA has now become a standard in the sense that the Pareto GA provides a very efficient way to find a wide range of solutions to a given problem. Although this approach proposed by Goldberg was further developed[17] and led to many applications,[18–20] all of these approaches are based on the concept of Pareto ranking and use either sharing or mating restrictions to ensure diversity. In this paper, however, a multiple objective scheme based on the concept of Pareto optimality was developed, and we introduce two different approaches, Nash GA[21] and new coevolutionary optimization algorithm based on evolutionary game theory, to solve MOPs.

2 Definition of multiobjective optimization problems (MOPs) and other concepts

2.1 Definition of MOPs

General MOPs contain a set of n decision variables, a set of k objective functions, and a set of m constraints. In this case, objective functions and constraints, respectively, become functions and constraints of the decision variables. If the goal of MOPs is to maximize objective functions vector y, then maximize

$$y = f(x) = \left(f_1(x), \ldots, f_i(x), \ldots, f_k(x)\right)$$

subject to

$$e(x) = \left(e_1(x), \ldots, e_j(x), \ldots, e_m(x)\right) \le 0 \tag{1}$$

where

$$x = \left(x_1, x_2, \ldots, x_n\right) \in X, \quad y = \left(y_1, y_2, \ldots, y_k\right) \in Y.$$

In Eq. 1, x is called a decision variable vector and y is called an objective function vector. The decision variable space is denoted by X and the objective function space is denoted by Y. The constraint condition $e(x) \le 0$ determines the set of feasible solutions.[22] The set of solutions of MOPs consist of all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another.[23] In contrast to single-objective optimization problems (SOPs), MOPs have a set of solutions known as the Pareto-optimal set. This solution set is generally called nondominated solutions and is optimal in the sense that no other solutions are superior to them in the search space when all objectives are considered. Mathematically, the concept of Pareto optimality is given below.

2.2 Definition of Pareto optimality

A decision vector $x \in X_f$ is said to be nondominated regarding a set $A \subseteq X_f$ iff

$$a \in A : a > x \tag{2}$$

where the feasible set X_f is defined as the set of decision vectors x that satisfy the constraints

$$e(x) : X_j = \{x \in X | e(x) \leq 0\}.$$

If it is clear within what context the set A is meant, it is simply left out. Moreover, x is said to be Pareto optimal iff x is nondominated regarding X_f. This equation means that there is no single optimal solution but rather a set of optimal trade-offs. None can be identified as better than the others unless preference information is included. The entirety of all Pareto-optimal solutions is called the Pareto-optimal set; the corresponding objective vectors form the Pareto-optimal front or surface.[22]

2.3 Definition of nondominated sets and fronts

Let $A \subseteq X_f$. The $p(A)$ gives the set of nondominated decision vectors in $A : p(A) = \{a \in A | a$ is nondominated regarding $A\}$. The set $p(A)$ is the nondominated set regarding A, the corresponding set of objective vectors $f(p(A))$ is the nondominated front regarding A. Furthermore, the set $X_p = p(X_f)$ is called the Pareto-optimal set and the set $Y_p = f(X_p)$ is denoted as the Pareto-optimal front. The Pareto-optimal set is comprised of the globally optimal solutions. However, as with SOPs, there may also be local optima that constitute a nondominated solution set within a certain neighborhood. These correspond to the concepts of global and local Pareto-optimal sets introduced by Deb.[24]

3 Evolutionary optimization approaches based on game theory

Since the mathematical basis was founded by von Neumann in the late 1920s, game theory has contributed to providing solutions to MOPs that are indulged in the sphere of mathematics and economics. Game theory introduces the notion of game and player associated with an optimization problem. In the case of a multiobjective design through game theory, each candidate involved is named as a player and has their own criterion to be the winner of the game. During the game, they try to improve their criteria until the system reaches equilibrium. In this section, we introduce two searching algorithms for finding optimized solutions of MOPs through the evolutionary game. The first algorithm results from a solution of a noncooperative game introduced by Nash in the early 1950s. This approach has brought in the concept of the "game player" for solving MOPs involved in game theory and economics.[21] The second algorithm is a coevolutionary algorithm using the game model, which is the newly proposed approach in this article. This approach searches the evolutionary stable strategy (ESS) as solutions of MOPs.

3.1 Nash equilibrium

Nash equilibrium is the solution of a noncooperative strategy for MOPs. This concept was introduced by Nash in 1952. According to Nash, each participant in the game has their own strategy set and objective function. During the game, each player searches for the optimal strategy for the objective function while the strategies of others are fixed. Nash frequency (σ) indicates how frequently the game strategy is changed by participants. Generally $\sigma = 1$, which means that the exchange of best strategies takes place at the end of each generation. Within this framework, evolutionary gaming is conducted and when no player can further improve their criterion, the system is then considered to have reached a state of equilibrium named Nash equilibrium.[25]

3.2 Nash genetic algorithm (Nash GA)

The idea of Nash GA is to bring together genetic algorithms and Nash strategy in order to find the Nash equilibrium as a solution to MOPs. In the following example, represented in Fig. 1, we present the attempts to optimize two different objectives and how such merging can be achieved with two players.

Let $s = XY$ represent the potential solution for a dual objective optimizations problem and two populations were allotted for each player. The optimization task of Player 1 is performed by population1 whereas that of Player 2 is performed by population2. Then X denotes the subset of variables handled by Player 1 and is optimized along criterion 1. Similarly, Y denotes the subset of variables handled by Player 2 and optimized along criterion 2. Thus, as advocated by Nash theory, Player 1 optimizes s with respect to the first criterion by regeneration of population1 for X, while Y is fixed by Player 2. Symmetrically, Player 2 optimizes s with respect to the second criterion by regeneration of population2 for Y, while X is fixed by Player 1.

Let X_{k-1} be the best value found by Player 1 at generation $k - 1$ and Y_{k-1} the best value found by Player 2 at generation $k - 1$. At generation k, Player 1 optimizes X_k while using Y_{k-1} in order to evaluate s (in this case, $s = X_k Y_{k-1}$). At the same time, Player 2 optimizes Y_k while using X_{k-1} in order to evaluate s (in this case, $s = X_{k-1} Y_k$). After the optimization process, Player 1 sends the best value X_k to Player 2 who will use it at generation $k + 1$. Similarly, Player 2 sends the best value Y_k to Player 1 who will use it at generation $k + 1$. Nash equilibrium is reached when neither Player 1 nor Player 2 can further improve their criteria.[21]

For this algorithm, Sefrioui et al.[26] also uses distance-dependent mutation, which is a technique evolved to maintain diversity in small populations. Instead of a fixed

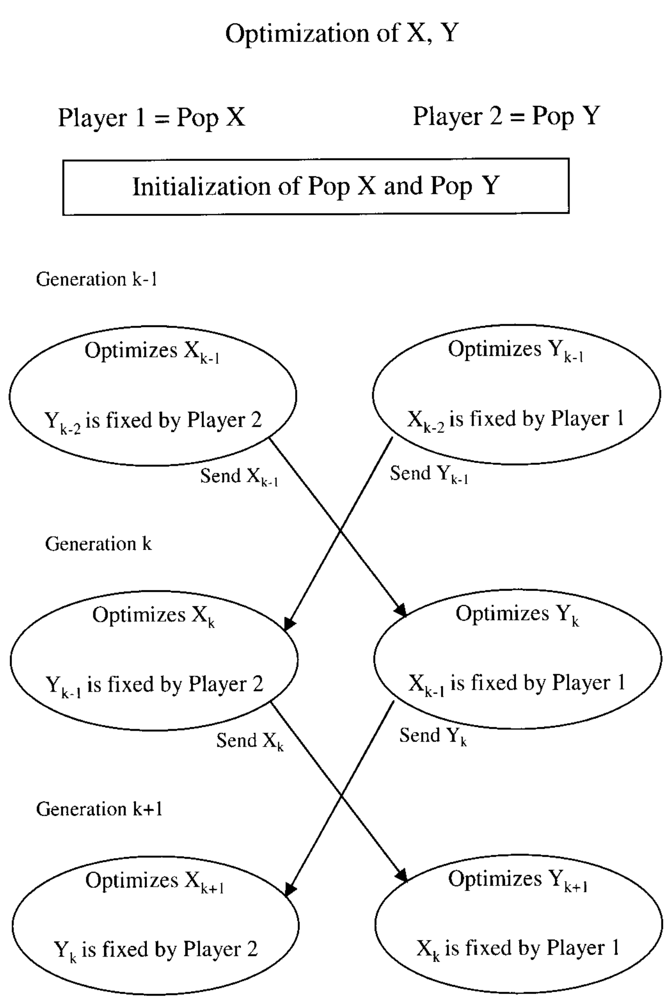

Fig. 1. A block diagram of Nash genetic algorithm (Nash GA)

mutation rate, each offspring has its mutation rate computed after each mating. This mutation rate depends on the distance between the two parents.

3.3 Evolutionary stable strategy (ESS)

The primary contribution of evolutionary game theory (EGT) is the concept of the evolutionary stable strategy (ESS). ESS was proposed by Maynard-Smith,[27] a biologist of worldwide fame. He defined an ESS as a strategy such that, if all the members of a population adopt it, then no mutant strategy could invade the population under the influence of natural selection. The ESS is a refinement concept of the Nash equilibrium that does away with the traditional assumption of agent rationality. Instead, Maynard-Smith showed that game–theoretic equilibria can be achieved through a process of Darwinian selection.[28] Nevertheless, the ESS is defined as a static concept, and since its introduction many other stability concepts have been proposed,[29] including those that are more properly rooted in dynamical systems theory.[30] The ESS corresponds to a dynamical attractor.[31]

From the idea that the target of these two solutions based on game theory was used to solve mathematical and economical problems covering MOPs and that the game mechanism can be implemented by the coevolutionary algorithm, we embodied the coevolutionary algorithm based on evolutionary game theory. In this approach, each population is allotted a game player and the fitness of individuals in the population is evaluated from each objective function and rewarded from the game matrix.

3.4 Coevolutionary algorithm based on evolutionary game theory

In this section, coevolutionary algorithm based on the ESS concept designed for searching the Pareto front of MOPs is explained. Through the evolutionary game, players for each objective function try to optimize their own objective function and all individuals of the population are regenerated after players are rewarded. The reward value is determined from the previously defined game matrix.

For example, in the case of minimization MOPs which have the two variables x and y and objective functions $f_1(x, y)$ and $f_2(x, y)$, the architecture of populations for coevolutionary algorithm is designed as shown in Fig. 2 and Tables 1 and 2.

In Fig. 2, fitness F_n is determined from the game matrix. The game matrices are defined in Tables 1 and 2.

The pay of the game for each population, G_n, is calculated from the difference of two objective functions.

$$\begin{aligned} G_1 &= \left[(x_n, y_n), (x'_n, y'_n)\right] = f_1(x_n, y_n) - f_2(x'_n, y'_n) \\ G_2 &= \left[(x_n, y_n), (x'_n, y'_n)\right] = f_2(x'_n, y'_n) - f_1(x_n, y_n) \end{aligned} \tag{3}$$

From these pays, the fitness of each player is calculated.

$$\begin{aligned} F_n &= \left\{\alpha - G_1\left[(x_n, y_n), (x'_n, y'_n)\right]\right\}/2\alpha, \\ F'_n &= \left\{\alpha - G_2\left[(x_n, y_n), (x'_n, y'_n)\right]\right\}/2\alpha \end{aligned} \tag{4}$$

Where α is a constant to prevent F_n or F'_n from being zero so that α must be $\mathrm{Max}|G_k[(x_n, y_n), (x'_n, y'_n)]|$. From these establishments, the coevolutionary optimization algorithm is as follows:

[Step1] Two populations are randomly generated as in Fig. 2.
[Step2] The player selected in the first population plays with the second population's one and then is paid off using Table 1 and Eq. 3.
[Step3] The player in the second population is paid off using Table 2 and Eq. 3.
[Step4] The fitness F_n and F'_n are updated using Eq. 4.
[Step5] The process from [Step2] to [Step3] is executed for all individuals of each population one by one.
[Step6] Each population is regenerated separately.
[Step7] The process from [Step2] to [Step3] is executed until ESS is found.

Keeping these ideas, we apply Nash GA and coevolutionary algorithm based on game theory to MOPs.

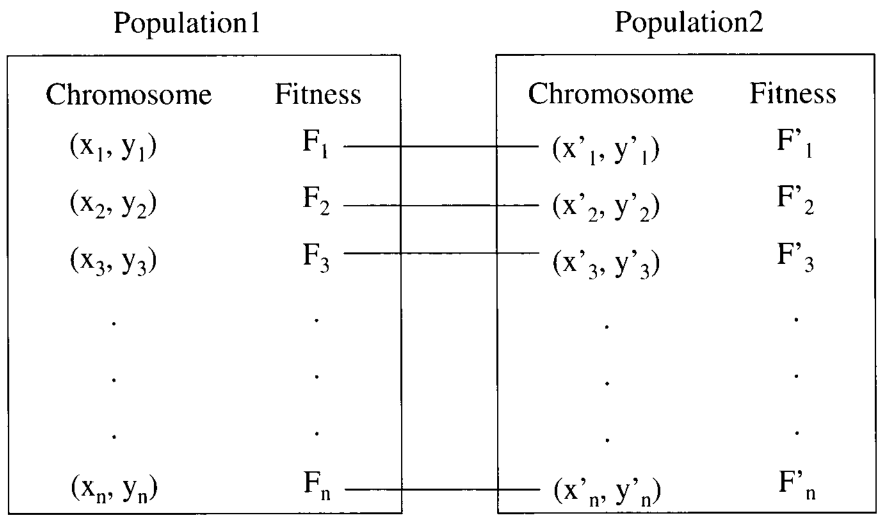

Fig. 2. Population architecture of coevolutionary algorithm

Table 1. The game matrix for population 1 of coevolutionary algorithm

	(x'_1, y'_1)	(x'_2, y'_2)	(x'_3, y'_3)	...	(x'_n, y'_n)
(x_1, y_1)	$G_1[(x_1, y_1), (x'_1, y'_1)]$	$G_1[(x_1, y_1), (x'_2, y'_2)]$	$G_1[(x_1, y_1), (x'_3, y'_3)]$	...	$G_1[(x_1, y_1), (x'_n, y'_n)]$
(x_2, y_2)	$G_1[(x_2, y_2), (x'_1, y'_1)]$	$G_1[(x_2, y_2), (x'_1, y'_1)]$	$G_1[(x_2, y_2), (x'_3, y'_3)]$	...	$G_1[(x_2, y_2), (x'_n, y'_n)]$
(x_3, y_3)	$G_1[(x_3, y_3), (x'_1, y'_1)]$	$G_1[(x_3, y_3), (x'_2, y'_2)]$	$G_1[(x_2, y_2), (x'_3, y'_3)]$	...	$G_1[(x_3, y_3), (x'_n, y'_n)]$
$\vdots$	...	...	...	...	...
(x_n, y_n)	$G_1[(x_n, y_n), (x'_1, y'_1)]$	$G_1[(x_n, y_n), (x'_2, y'_2)]$	$G_1[(x_n, y_n), (x'_3, y'_3)]$	...	$G_1[(x_n, y_n), (x'_n, y'_n)]$

Table 2. The game matrix for population 2 of coevolutionary algorithm

	(x'_1, y'_1)	(x'_2, y'_2)	(x'_3, y'_3)	...	(x'_n, y'_n)
(x_1, y_1)	$G_2[(x_1, y_1), (x'_1, y'_1)]$	$G_2[(x_1, y_1), (x'_2, y'_2)]$	$G_2[(x_1, y_1), (x'_3, y'_3)]$	...	$G_2[(x_1, y_1), (x'_n, y'_n)]$
(x_2, y_2)	$G_2[(x_2, y_2), (x'_1, y'_1)]$	$G_2[(x_2, y_2), (x'_2, y'_2)]$	$G_2[(x_2, y_2), (x'_3, y'_3)]$	...	$G_2[(x_2, y_2), (x'_n, y'_n)]$
(x_3, y_3)	$G_2[(x_3, y_3), (x'_1, y'_1)]$	$G_2[(x_3, y_3), (x'_2, y'_2)]$	$G_2[(x_3, y_3), (x'_3, y'_3)]$	...	$G_2[(x_3, y_3), (x'_n, y'_n)]$
$\vdots$	...	...	...	...	...
(x_n, y_n)	$G_2[(x_n, y_n), (x'_1, y'_1)]$	$G_2[(x_n, y_n), (x'_2, y'_2)]$	$G_2[(x_n, y_n), (x'_3, y'_3)]$	...	$G_2[(x_n, y_n), (x'_n, y'_n)]$

4 Test problems and evaluation

While various evolutionary approaches and variations of them were successfully applied to MOPs, in recent years some researchers have investigated particular topics of evolutionary multiobjective searches. In spite of the variety of approaches, there is a lack of studies that compare the performance and different aspects of these approaches. In this section, we provide a systematic comparison of several multiobjective evolutionary algorithms. The problems considered here are those of Zitzler et al.,[32] which cover six representative MOPs. A corresponding test function is constructed following the guidelines given by Deb.[24]

4.1 Test MOPs

In the previous section, we introduced various established evolutionary algorithms for solving MOPs. In spite of this variety, there is a lack of studies that compare the performance and different aspects of these approaches. Among these studies we introduce several here. On the theoretical side, Fonseca and Fleming[14] discussed the influence of different fitness assignment strategies on the selection process. On the practical side, Zitzler and Thiele[3,33] used a NP-hard 0/1 knapsack problem to compare several multiobjective EAs. In these studies, a systematic comparison of multiobjective EAs was provided, including a random search strategy as well as a single-objective EA using objective aggregation. The basis of this empirical study is formed by a set of well-defined, domain-independent test functions that allow the investigation of independent problem features. We thereby draw upon results where problem features that may make convergence of EAs to the Pareto-optimal front difficult are identified and, furthermore, methods of constructing appropriate test functions are suggested.[1] The functions considered here cover the range of convexity, nonconvexity, discrete, multimodal, deceptive, and nonuniform Pareto fronts. Deb[34] identified several features that may cause difficulties for multiobjective EAs in: (1) converging to the Pareto-optimal front, and (2) maintaining diversity within the population. Each of the test functions defined below is structured in the same manner and consists of the three functions f_1, g, and h.

Minimize

$$t(x) = \left[f_1(x_1), f_2(x)\right]$$

subject to

$$f_2(x) = g(x_2,\ldots,x_n)\cdot h\left[f_1(x_1), g(x_2,\ldots,x_n)\right] \tag{5}$$

where $x = (x_1, \ldots, x_n)$. The function f_1 is a function of the first decision variable only, g is a function of the remaining $m - 1$ variables, and the parameters of h are the function values of f_1 and g. The test functions differ in these three functions as well as in the number of variables m and in the values the variables may take.[1]

The test function t_1 has a convex Pareto-optimal front

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,\ldots,x_n) &= 1 + 9\cdot\left(\textstyle\sum_{i=2}^{n} x_i\right)\Big/(n-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{6}$$

where $n = 30$, and $x_i \in [0,1]$.

The test function t_2 has nonconvex Pareto-optimal front

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,\ldots,x_n) &= 1 + 9\cdot\left(\textstyle\sum_{i=2}^{n} x_i\right)\Big/(n-1) \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned} \tag{7}$$

where $n = 30$, and $x_i \in [0,1]$.

The test function t_3 represents the discreteness feature; its Pareto-optimal front consists of several non-contiguous convex parts:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,\ldots,x_n) &= 1 + 9\cdot\left(\textstyle\sum_{i=2}^{n} x_i\right)\Big/(n-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \end{aligned} \tag{8}$$

where $n = 30$, and $x_i \in [0,1]$.

The test function t_4 contains 21^9 local Pareto-optimal sets and therefore tests for the EA's ability to deal with multimodality:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,\ldots,x_n) &= 1 + 10(n-1) + \textstyle\sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right] \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{9}$$

where $n = 30$, $x_1 \in [0,1]$ and $x_2, \ldots, x_n \in [0,1]$.

The test function t_5 describes a deceptive problem and distinguishes itself from the other test functions in that x_i represents a binary string:

$$\begin{aligned} f_1(x_1) &= 1 + u(x_1) \\ g(x_2,\ldots,x_n) &= \textstyle\sum_{i=2}^{n} v\left[u(x_i)\right] \\ h(f_1, g) &= 1/f_1 \end{aligned} \tag{10}$$

where $u(x_i)$ gives the number of ones in the bit vector x_i (unitation),

$$v\left[u(x_i)\right] = \begin{Bmatrix} 2 + u(x_i) & \mathit{if} & u(x_i) < 5 \\ 1 & \mathit{if} & u(x_i) = 5 \end{Bmatrix}$$

and $n = 11$, $x_1 = \{0,1\}^{30}$ and $x_2, \ldots, x_n \in \{0,1\}^5$.

The test function t_6 includes two difficulties caused by the nonuniformity of the objective space: firstly, the Pareto-optimal solutions are nonuniformly distributed along the global Pareto front [the front is biased for solutions for which $f_1(x_1)$ is near one]; secondly, the density of the solutions is least near the Pareto-optimal front and highest away from the front:

$$\begin{aligned} f_1(x_1) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\ g(x_2,\ldots,x_n) &= 1 + 9\cdot\left[\left(\textstyle\sum_{i=2}^{n} x_i\right)\Big/(n-1)\right]^{0.25} \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned} \tag{11}$$

where $n = 10$, $x_i \in [0,1]$.

We apply two evolutionary optimization algorithms based on game theory previously introduced to these six MOPs and analyze the results.

4.2 Experimental results and analysis

Optimized solutions of MOPs by two evolutionary optimization algorithms, the Nash GA and coevolutionary algorithm based on game theory are shown in Figs. 3–20. These solutions are known as the Pareto-optimal set or Pareto front. In analysis of these results we cite figures that display optimized solutions using the genetic algorithms by Zitzler et al.[32] The results of the eight different evolutionary optimization algorithms used by Zitzler et al.[32] to solve previously introduced test MOPs are shown in Figs. 3, 6, 9, 12, 15, and 18. In these figures, the evolutionary algorithms used are as follows:

RAND: A random search algorithm.
FFGA: Fonseca and Fleming's multiobjective EA.
NPGA: The niched Pareto genetic algorithm.
HLGA: Hajela and Lin's weighted-sum based approach.
VEGA: The vector evaluated genetic algorithm.
NSGA: The nondominated sorting genetic algorithm.
SOEA: A single-objective evolutionary algorithm using weighted-sum aggregation.

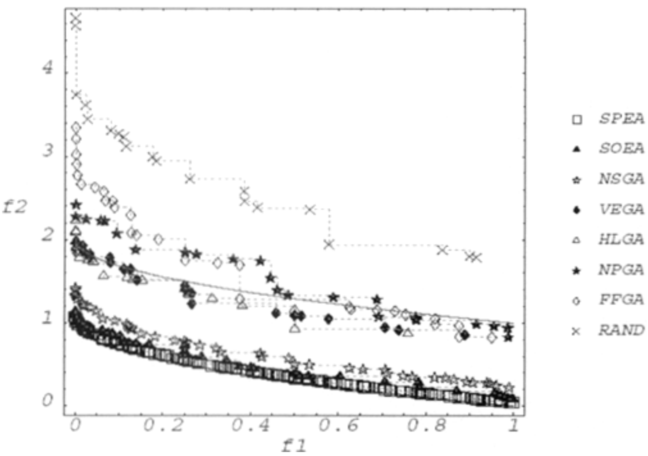

Fig. 3. The Pareto fronts of T1 multiobjective optimization problems (MOPs) searched by other evolutionary algorithms (EAs) (from Ziztler et al.[30]). For algorithm codes, refer to text

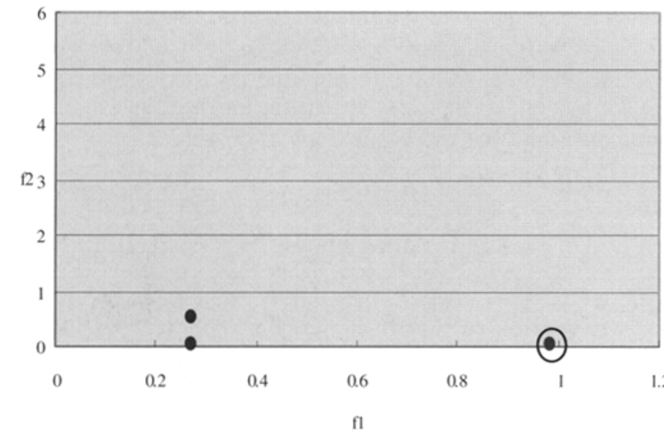

Fig. 4. The Nash equilibrium point of T1 MOPs searched by Nash GA

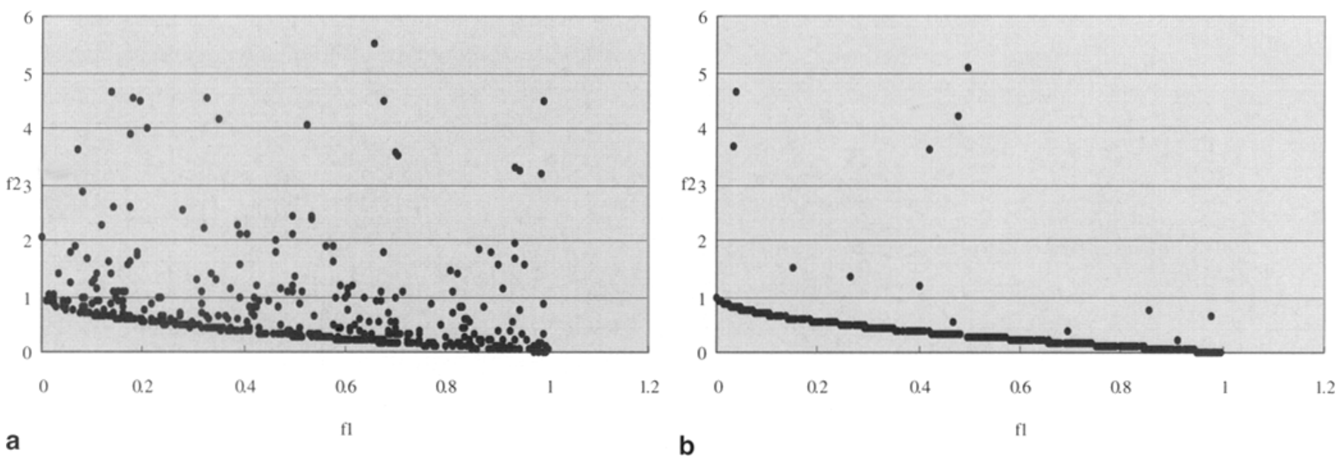

Fig. 5a,b. The Pareto front of T1 MOPs searched by coevolutionary algorithm based on game theory using **a** simple mutation and **b** distance-dependent mutation

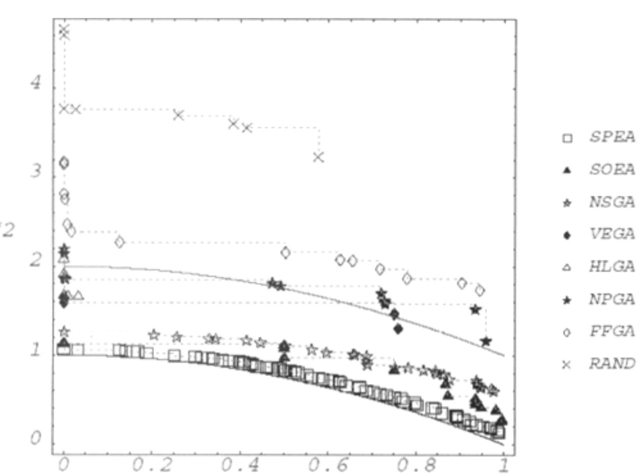

Fig. 6. The Pareto fronts of T2 MOPs searched by other EAs (from Ziztler et al.[30])

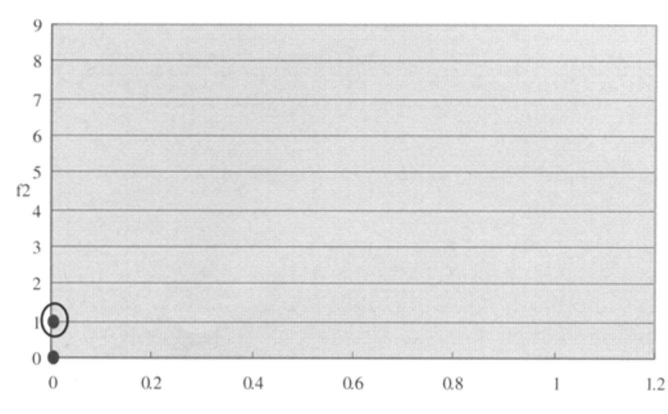

Fig. 7. The Nash equilibrium point of T2 MOPs searched by Nash GA

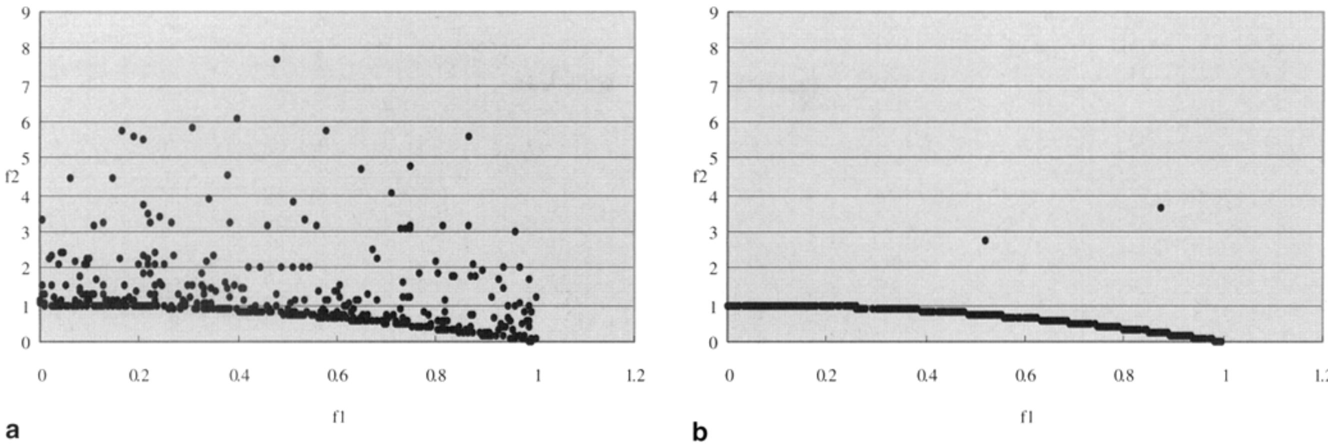

Fig. 8a,b. The Pareto front of T2 MOPs searched by coevolutionary algorithm based on game theory using **a** simple mutation and **b** distance-dependent mutation

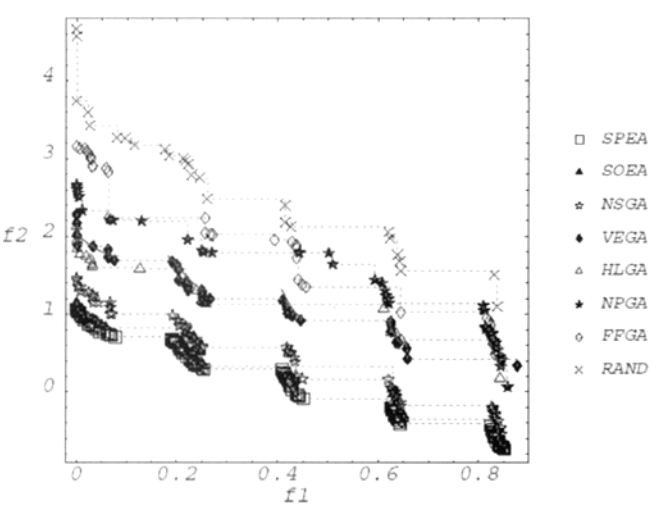

Fig. 9. The Pareto fronts of T3 MOPs searched by other EAs (from Ziztler et al.[30])

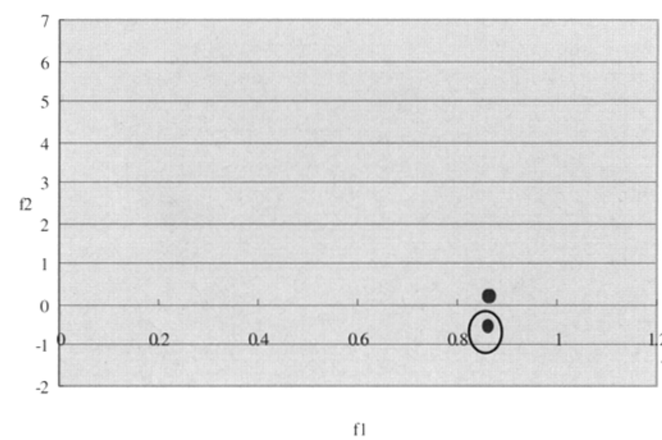

Fig. 10. The Nash equilibrium point of T3 MOPs searched by Nash GA

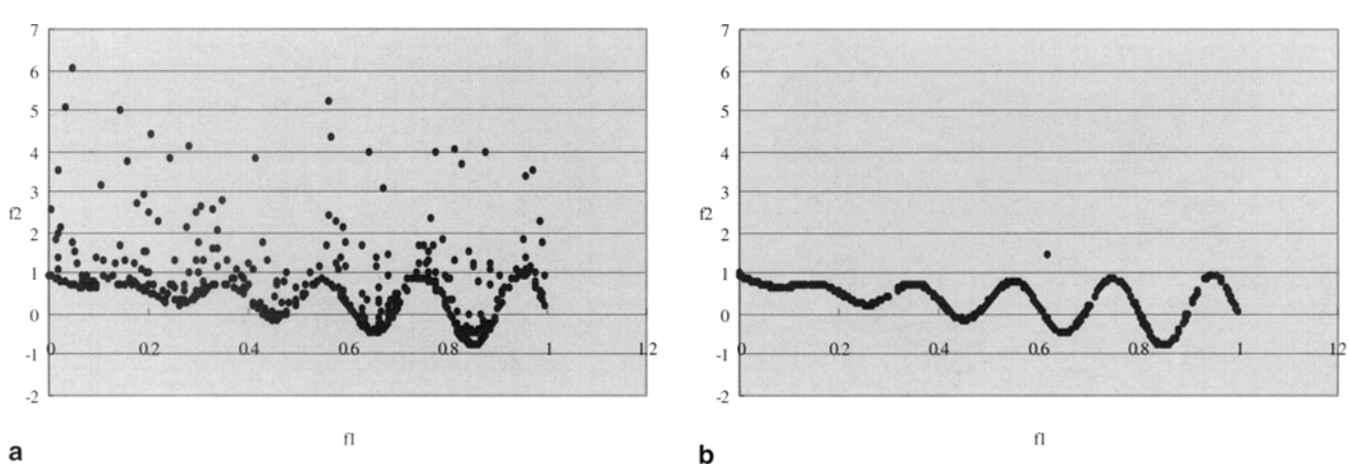

Fig. 11a,b. The Pareto front of T3 MOPs searched by coevolutionary algorithm based on game theory using **a** simple mutation and **b** distance-dependent mutation

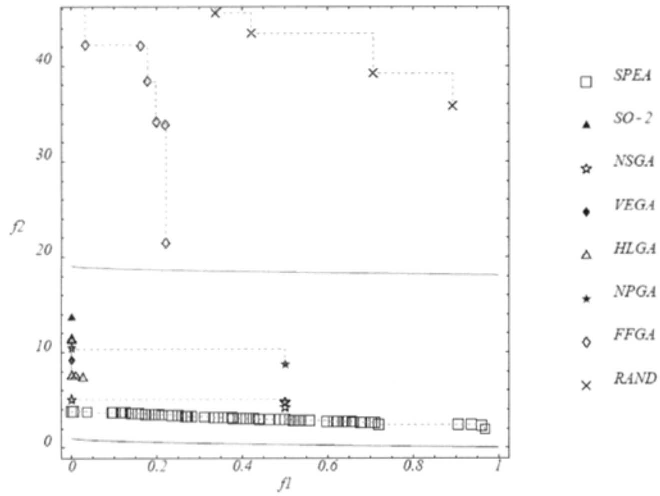

Fig. 12. The Pareto fronts of T4 MOPs searched by other EAs (from Ziztler et al.[30])

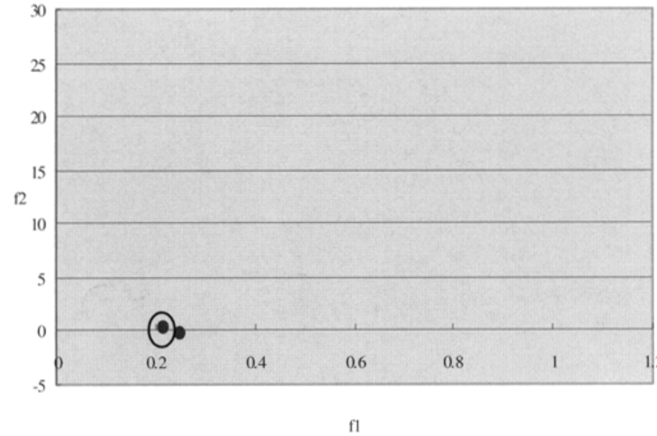

Fig. 13. The Nash equilibrium point of T4 MOPs searched by Nash GA

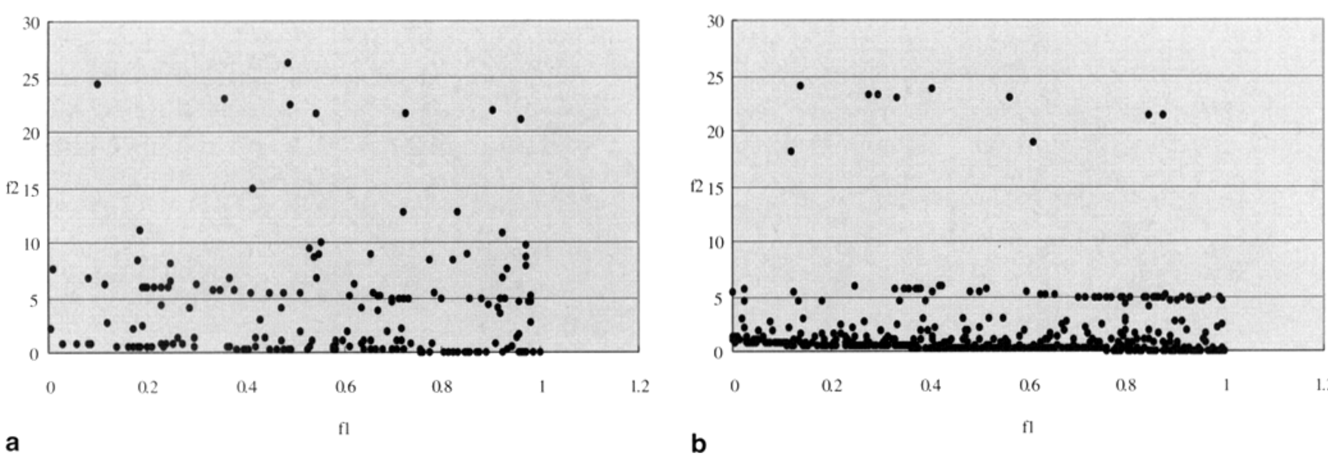

Fig. 14a,b. The Pareto front of T4 MOPs searched by coevolutionary algorithm based on game theory using **a** nonelitism and **b** elitism

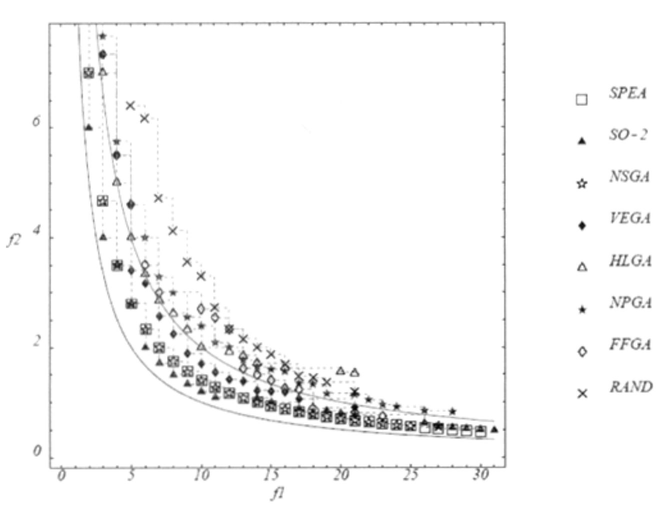

Fig. 15. The Pareto fronts of T5 MOPs searched by other EAs (from Ziztler et al.[30])

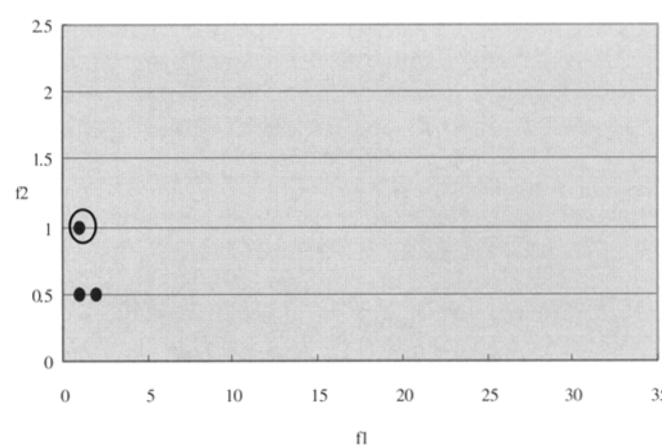

Fig. 16. The Nash equilibrium point of T5 MOPs searched by Nash GA

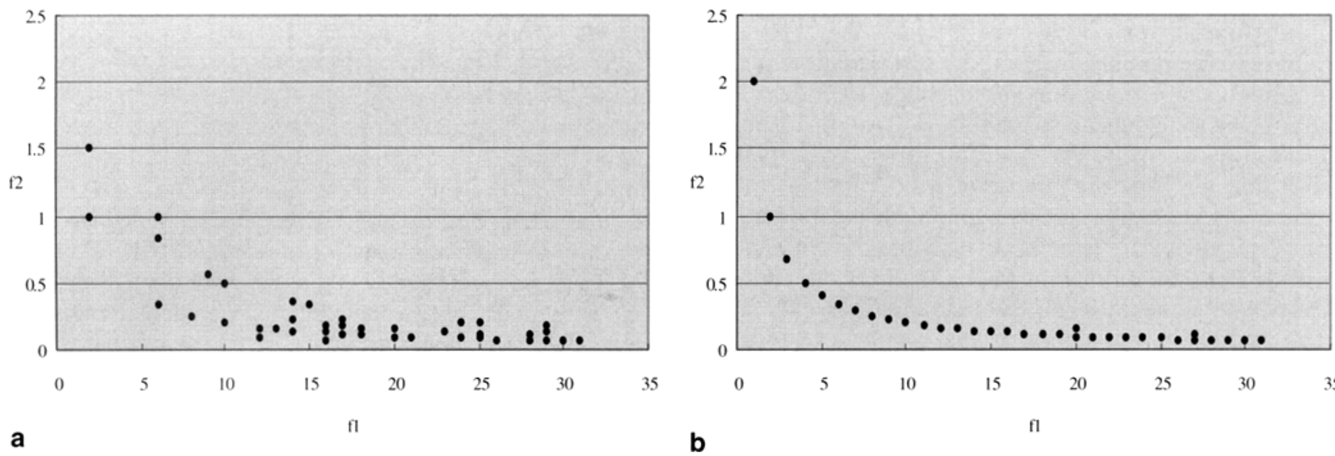

Fig. 17a,b. The Pareto front of T5 MOPs searched by coevolutionary algorithm based on game theory using **a** simple mutation and **b** distance-dependent mutation

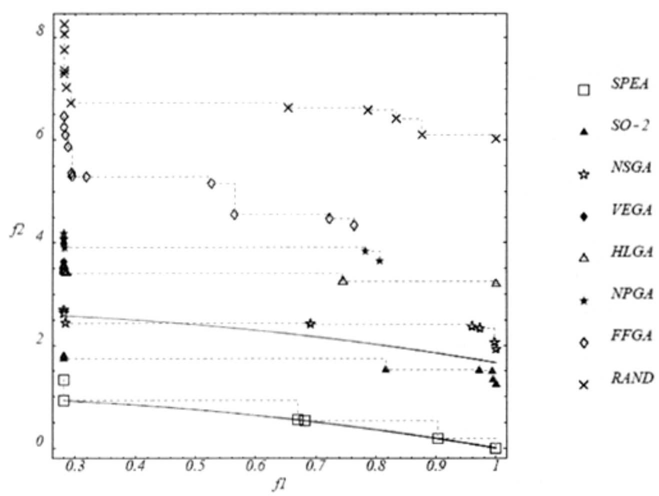

Fig. 18. The Pareto fronts of T6 MOPs searched by other EAs (from Ziztler et al.[30])

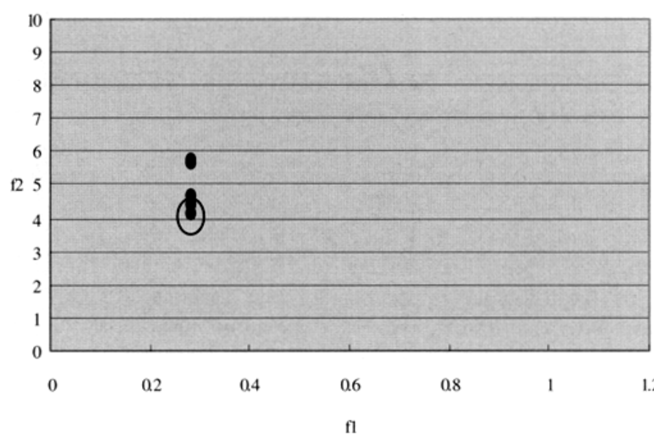

Fig. 19. The Nash equilibrium point of T6 MOPs searched by Nash GA

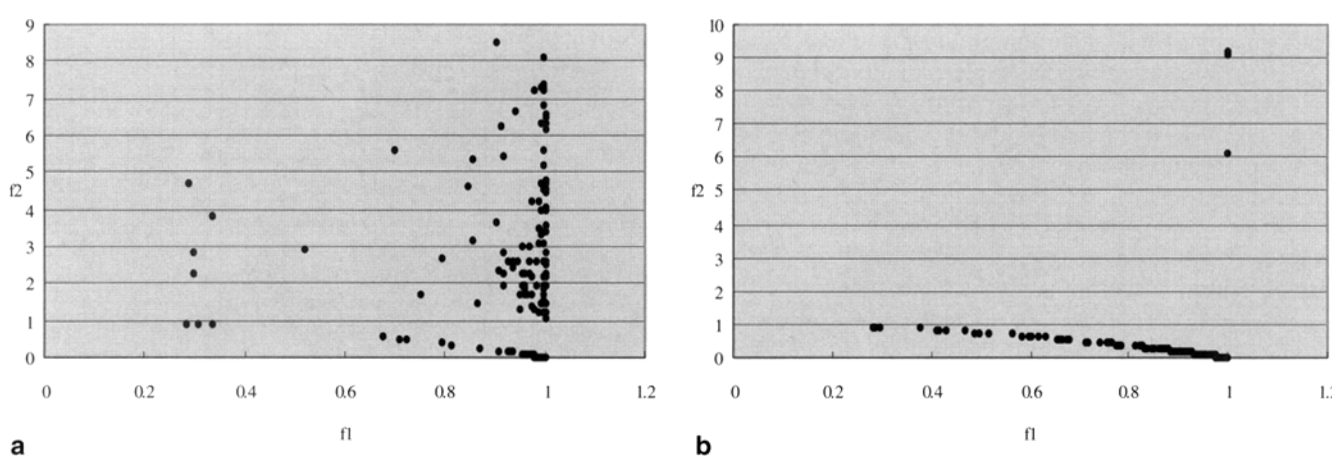

Fig. 20a,b. The Pareto front of T6 MOPs searched by coevolutionary algorithm based on game theory using **a** simple mutation and **b** distance-dependent mutation

SPEA: The strength Pareto evolutionary algorithm.

In our experiments, we used the two mutation methods of general simple mutation and distance-dependent mutation.[14] Distance-dependent mutation involves using the distance between the two mates in order to compute the mutation rate. The mutation rate is no longer a constant parameter, but is dynamically computed for both children and depends on the parents. Every time a couple of individuals is chosen for mating, the mutation rate is computed and will be applied to their children after crossover occurs. If the parents are quite close, that would lead to a high mutation rate for their children, whereas the mutation rate will be smaller if they are distant. As a distance criteria we took a relative distance that takes into account the bounds of each variable of the individual. We apply this nonuniform mutation to Nash GA and coevolutionary algorithm based on game theory. Figures 4, 7, 10, 13, 16, and 19 show the optimized experimental results using Nash GA. The circles in these figures indicate the Nash equilibrium points as solutions of test MOPs. Figures 5, 8, 11, 14, 17, and 20 display optimized experimental results using coevolutionary algorithm based on game theory.

Each experimental result of coevolutionary algorithm was determined twice using simple uniform mutation and distance-dependent mutation. The genetic algorithm parameters used were number of generations: 500, population size: 30, one-point crossover rate: 0.3, and in the case of simple mutation, mutation rate: 0.06. In the first experimental case using Nash GA, the circled points show the final Nash equilibrium points for each test problem. By comparing with the Pareto fronts in the cited figures of Zitzler et al.[32] with these results we can see that Nash equilibrium point exists in the Pareto front. Therefore, we conclude that the Nash GA can find MOPs solutions. However, this algorithm cannot search the Pareto optimal set but rather a single solution which depends on the initial population condition. In the second experimental case using coevolutionary algorithm based on game theory, we conducted the experiment twice for each test MOPs. In the first trial, we used only simple uniform mutation, while in the second trial, we used distance-dependent mutation. For every test MOP, a coevolutionary algorithm based on game theory using distance-dependent mutation can find ESS more rapidly and clearly than using simple mutation except for T4 test MOP. In the case of T4 test MOP, we use elitism rather than distance-dependent mutation because the T4 test problem is multimodal and Zitzler et al. also used elitism to solve this problem.[32] By comparing the Pareto front found by Zitzler et al. with these second experimental results for each MOPs, we can see that the ESS found by coevolutionary algorithm based on game theory is very similar to the Pareto front. Therefore, we conclude that the algorithm newly proposed in this study can search optimal solution sets of MOPs, and this algorithm needs fewer parameter resources than the evolutionary algorithms used in the experiments of Zitzler et al.[32]

5 Conclusions

In this article, we introduced several approaches to solve MOPs. In the introduction, established optimization algorithms based on the concept of the Pareto-optimal set were introduced. Contrary to these algorithms, we introduced the theoretical backgrounds of the Nash genetic algorithm (Nash GA) and evolutionary stable strategy (ESS), which are based on evolutionary game theory (EGT). Moreover, ESS is the basis of the coevolutionary algorithm based on game theory as newly proposed in this study. Generally, ESS is the equilibrium solution of a noncooperative game model, but we confirmed that this strategy can be used as an optimal solution set of MOPs. In future work, we will implement this idea in a robot controller for an environment which has more than one conflict object functions.

Acknowledgment This research was supported by Braintech 21 from the Ministry of Science and Technology of Korea.

References

1. Zitzler E, Deb K, Thiele L (1999) Comparison of multiobjective evolutionary algorithms: empirical results. Technical Report 70, Computer Engineering Networks Laboratory (TIK), Swiss Federal Inst Technol (ETH) Zurich, Switzerland, Proceedings of the 1999 Genetic and Evolutionary Computation Conference, Workshop Program, Orlando, Florida, pp 121–122
2. Fonseca CM, Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part I: a unified formulation. IEEE Trans Syst Man Cy 28:26–37
3. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms – a comparative case study. In: Eiben AE, Beack T, Schoenauer M, et al. (eds) Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Springer, Berlin Heidelberg New York, pp 292–301
4. Cohon JL (1978) Multiobjective programming and planning. Academic, New York
5. Schaffer JD (1984) Some experiments in machine learning using vector evalutated genetic algorithms. Ph.D. dissertation, Vanderbilt University
6. Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of the International Conference on Genetic Algorithms and their Applications, pp 93–100
7. Horn J, Nafpliotis N, Goldberg DE (1994) A niched pareto genetic algorithm for multiobjective optimization. IEEE World Congress on Computational Intelligence (ICEC '94) 1:82–87
8. Fourman MP (1985) Compaction of symbolic layout using genetic algorithms. In: Grefenstette JJ (ed) Genetic Algorithms and Their Applications: Proceedings of the 1st International Conference on Genetic Algorithms. Lawrence Erlbaum, Princeton, pp 141–153
9. Fonseca CM, Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionry algorithms-part II: application example. IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans 28:38–47
10. Kursawe F (1991) A variant of evolution strategies for vector optimization. In: Schwefel H-P, Maenner R (eds) Parallel Problem Solving from Nature, 1st Workshop Proceedings, vol 496 of Lecture Notes in Computer Science. Springer, Berlin Heidelberg New York, pp 193–197
11. Hajela P, Lin C-Y (1992) Genetic search strategies in multicriterion optimal design. Struct Optim 4:99–107
12. Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization.

Proceedings of the Fifth International Conference, Kaufmann, San Mateo, CA, pp 416–423
13. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, MA
14. Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. Evolution Comput 3:1–16
15. Horn J, Nafpliotis N (1993) Multiobjective optimization using the niched Pareto genetic algorithm. IlliCAL Report 93005, University of Illinois at Urbana-Champaign, Urbana, IL
16. Goldberg DE, Richardson JJ (1987) Genetic algorithms with sharing for multi-modal function optimization. Genetic Algorithms and Their Application: Proceedings of the Second ICGA, Lawrence Erlbaum, Hillsdal, pp 41–49
17. Srinivas N, Deb K (1995) Multiobjective optimization using nondominated sorting in genetic algorithms. Evolution Comput 2:221–248
18. Poloni C (1995) Hybrid GA for multi objective aerodynamic shape optimization. In: Galan M, Winter G, Periaux J, et al. (eds) Genetic algorithms in engineering and computer science. Wiley, Las Palmas, Spain, pp 397–415
19. Makinen R, Neittaanmaki P, Periaux J, et al. (1996) Parallel genetic solution for multiobjective MDO. Parallel CFD 96, Elsevier, Capri
20. Bristeau M-O, Glowinski R, Mantel B, et al. (1999) Genetic algorithms for electromagnetic backscattering: multiobjective optimization. In: Rahmat-Samii Y, Michielssen E (eds) System design using evolutionary optimization: genetic algorithms. Wiley, New York
21. Sefrioui M, Periaux J (2000) Nash genetic algorithms: examples and applications. Proceedings of the 2000 Congress on Evolutionary Computation CEC00, IEEE, pp 509–516
22. Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: methods and applications. Dissertation for the degree of Doctor of Technical Sciences, Swiss Federal Institute of Technology, Zurich
23. Zitzler E (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach
24. Deb K (1998) Multiobjective genetic algorithms: problem difficulties and construction of test functions. Technical Report No. CI-49/98, Department of Computer Science/XI, University of Dortmund, Germany
25. Nash JF (1951) Noncooperative games. Ann Math 54:289
26. Sefrioui M, Periaux J, Ganascia J-G (1996) Fast convergence thanks to diversity. In: Fogel LJ, Angeline PJ, Baeck T (eds) Proceedings of the Fifth Annual Conference on Evolutionary Programming, San Diego, CA. IEEE Computer Society
27. Maynard-Smith J (1982) Evolution and the theory of games. Cambridge University Press, Cambridge
28. Ficici SG, Pollack JB (2001) Game theory and the simple coevolutionary algorithm: some preliminary results on fitness sharing. GECCO 2001 Workshop on Coevolution: Turning Adaptative Algorithms upon Themselves, pp 2–7
29. Lessard S (1990) Evolutionary stability: one concept, several meanings. Theor Popul Biol 37:159–170
30. Rowe GW, Harvey IF, Hubbard SF (1985) The essential properties of evolutionary stability. J Theor Biol 115:269–285
31. Strogatz SH (1994) Nonlinear dynamics and chaos. Addison-Wesley, Reading, MAs
32. Zitzler E, Deb K, Thiele L (1999) Comparison of multiobjective evolutionary algorithms: empirical results. Proceedings of the 1999 Genetic and Evolutionary Computation Conference, Workshop Program, Orlando, Florida, pp 121–122
33. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evolution Comput 3:257–271
34. Deb K (1999) Multiobjective genetic algorithms: problem difficulties and construction of test problems. Evolution Comput 7:205–230