Optimal Strategies of the Iterated Prisoner's Dilemma Problem for Multiple Conflicting Objectives

Optimal Strategies of the Iterated Prisoner's Dilemma Problem for Multiple Conflicting Objectives

Shashi Mittal and Kalyanmoy Deb

***Abstract*— In this paper, we present a new paradigm of searching optimal strategies in the game of iterated prisoner's dilemma (IPD) using multiple-objective evolutionary algorithms. This method is more useful than the existing approaches, because it not only produces strategies that perform better in the iterated game but also finds a family of nondominated strategies, which can be analyzed to decipher properties a strategy should have to win the game in a more satisfactory manner. We present the results obtained by this new method and discuss substrategies found to be common among nondominated strategies. The multiobjective treatment of the IPD problem demonstrated here can be applied to other similar game-playing tasks.**

***Index Terms*— Evolutionary algorithms, games, multiobjective optimization, prisoner's dilemma.**

		Player 2	
	Decision	Cooperate	Defect
Player 1	Cooperate	$R=3$ $R=3$	$S=0$ $T=5$
	Defect	$T=5$ $S=0$	$P=1$ $P=1$

R: Reward S: Sucker T: Temptation P: Penalty

Fig. 1. Classical choice for payoff in prisoners dilemma (player 1s payoffs are given first).

I. Introduction

THE PRISONER'S dilemma is a well known game that has been extensively studied in economics, political science, machine learning [1], [2], and evolutionary biology [3]. In this game, there are two players, each of whom can make one of the two moves available to them: cooperate (C) or defect (D). Both players choose their moves simultaneously and independent of each other. Depending upon the moves chosen by either player, each of them gets a payoff. A typical payoff matrix is shown in Fig. 1.

When both players cooperate, they are awarded at an equal but intermediate level (the reward R). When only one player defects, he receives the highest possible payoff (the temptation T), while the other player gets the sucker's payoff (the sucker S). When both players defect, they receive an intermediate penalty (the penalty P).

Several interesting properties of the game can be immediately observed. It can be seen that this is a nonzero sum game: that is, the sum of the payoffs of the two players is not always a constant. In a one-shot game, both players will choose to defect, because this move is guaranteed to maximize the payoff of the player no matter what his opponent chooses. However, it can be seen that both players would have been better off by choosing to cooperate with each other (hence the dilemma).

In game theory, the move (D, D) of the players is termed as a *Nash equilibrium* [4], which is a steady state of the game in which no player has an incentive to shift from its strategy. Nash [5] proved that any n-player game has a Nash equilibrium, when randomization in choosing the moves is permitted. However, as is clear from the prisoner's dilemma game, a Nash equilibrium may not necessarily be the social optimum.

The situation becomes more interesting when the players play this game iteratively (called the iterated prisoner's dilemma or IPD) and the payoffs are accumulated over each iteration. If both players play the game for infinite number of turns, and the payoff of each player is the discounted sum of the payoffs at each turn of the game, then it is possible to have an equilibrium which is better than (D, D). The equilibrium outcomes in iterated games are defined by folk theorems [6]. For prisoner's dilemma, there are infinitely many equilibrium outcomes; in particular, it is possible to have an equilibrium outcome in which both players always cooperate.

Suppose that there are a number of players, and each player plays the iterated game with other players in a round robin fashion, the scores being cumulated over all the games. The winner of the game is the player with the maximum payoff at the end of the round robin tournament. The problem that we consider in this paper is to find optimal strategies that will ensure victory in such a tournament against a fixed set

Manuscript received May 6, 2006; revised February 22, 2007. Current version published June 10, 2009. K. Deb was supported by the Academy of Finland and the Helsinki School of Economics during his stay in Finland under the Finland Distinguished Professor (FiDiPro) program. He also would like to express his sincere thanks to the Academy of Finland and the Foundation of Helsinki School of Economics (Grant 118319) for supporting this work during his stay in Finland under the FiDiPro program.

S. Mittal is with the Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: mshashi@mit.edu).

K. Deb is with the Department of Business Technology, Helsinki School of Economics, 00250 Helsinki, Finland, and is also with the Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: deb@iitk.ac.in).

Digital Object Identifier 10.1109/TEVC.2008.2009459

of opponents. This has been a widely studied problem by game theorists and artificial intelligence experts alike. Axelrod was the first to study this problem in detail [1], [7], [8]. He used a single-objective evolutionary algorithm (EA) for finding the optimal strategies. This is discussed in Section II. Since Axelrod, there have been several studies on this problem [9]–[13]. Evolutionary algorithms have also been investigated for finding optimal strategies in the spatial iterated prisoner's dilemma game [14], [15], and also for the case where there are intermediate moves available to the players or noise has been added in the environment [16].

However, in all these studies the problem of finding optimal strategies has been viewed as a single-objective optimization problem. That is, the objective is to find strategies that maximize their own score in a round robin tournament. In this paper, we present a new approach of finding optimal strategies by considering the problem as a multiple objective optimization problem. In its simplest form, the objective of maximizing self-score and the objective of minimizing an aggregate opponent score should result in a conflicting scenario. There exist a number of motivations for considering the usual IPD problem as a multiobjective optimization problem.

1) The use of conflicting objectives of the game and treating the problem as a true multiobjective optimization problem may result in a number of interesting tradeoff optimal strategies that may provide useful information of playing the IPD in its generic sense than what a single optimal strategy derived from optimizing a single objective will provide.
2) In solving difficult optimization problems [17], [18], it is recently experienced that the use of additional objectives as helper objectives tends to provide a better search power to an optimization procedure than the sole use of a single objective. Since multiple tradeoff solutions are preserved in the search process in the case of multiple conflicting objectives, the sustained diversity among the solutions help to find better solutions, thereby allowing a better convergence near the true optimum of the problem. A goal of this paper is also to investigate whether a multiobjective application can find better solutions than that obtained by a single-objective application.
3) Such an multiobjective approach has not been previously investigated in the IPD literature and due to advancement of evolutionary multiobjective optimization (EMO) literature and their innovative applications in various other optimization tasks [18], [19], it is worth an attempt to investigate the outcome of a systematic EMO application to this important problem.
4) Since a multiobjective optimization task finds a number of tradeoff optimal or near-optimal (high-performing) solutions, an analysis of these solution has often resulted in revealing important insights about the underlying problem in many application studies [20]. A major motivation of this paper is to hope to unveil any such important properties or strategies that will predominantly exist among the tradeoff high-performing solutions of the IPD game. Such information is difficult to achieve by any other means and, if exist, will provide valuable information about what high-level strategies a player may keep in mind while playing the game with maximum performance.
5) Most games, including IPD, are multiobjective in nature, involving goals related to maximizing payoff of the player, minimizing payoff of the opponent player, maximizing the difference in payoffs between a player's own and that of the opponent, minimizing the average or maximum payoff of opponents, if played against a number of opponents, and so on. In traditional game theory, all the objectives are combined into a single payoff function for the players. However, these objectives can be combined in several different ways. By looking at the different objectives from a multiobjective optimization point of view, it is possible to get optimal strategies for all possible combinations of the objectives, by simply looking at the strategies in the tradeoff curve.

In the remaining sections, we first briefly discuss Axelrod's original single-objective study in Section II. Thereafter, we discuss this approach in detail in Section III. The details of the simulation results are discussed in Sections IV, V, and VI. A brief summary of the significance of the obtained results and conclusions of the study are presented in Section VII.

II. Axelrod's Study

Axelrod organized two tournaments [7] and invited strategies from a number of experts and game theorists for solving the IPD problem. To his surprise, he found that the winner in both the tournaments used a very simple strategy, namely the 'Tit for Tat'. This strategy cooperates on the first move, and then simply copies the opponent's last move in its subsequent moves. That such a simple strategy turned out to be the winner was quite surprising, and Axelrod set out to find other simple strategies with the same or greater power. Axelrod adopted a simple but elegant way for encoding strategies [1], and then used a single-objective evolutionary algorithm to obtain optimal strategies. His encoding scheme remained a standard way of handling the IPD problem and is described in somewhat detail here. In this paper, we also adopt a similar encoding scheme.

For each move in the game, there are four possibilities: both players can cooperate (CC or R for reward), the second player can defect while the first cooperates (CD or S for sucker), the first player can defect while the second cooperates (DC or T for temptation), or both players can defect (DD or P for penalty). To code a particular strategy, the particular behavioral sequence is coded as a three-letter string. For example, RRR would represent the sequence where both players cooperated over the previous three moves and SSP would represent the sequence where the first player was played for a sucker twice, and then finally defected. This three-letter sequence is then used to generate a number between 0 and 63, by interpreting it as a number in base 4. One such possible way is to assign a digit value to each of the characters in following way: $DD = P = 0$, $DC = T = 1$, $CD = S = 2$, and $CC = R = 3$. In this way, PPP would decode to 0, and SSR will decode to $(2 \cdot 4^2 + 2 \cdot 4^1 + 3 \cdot 4^0)$ or 43. With the

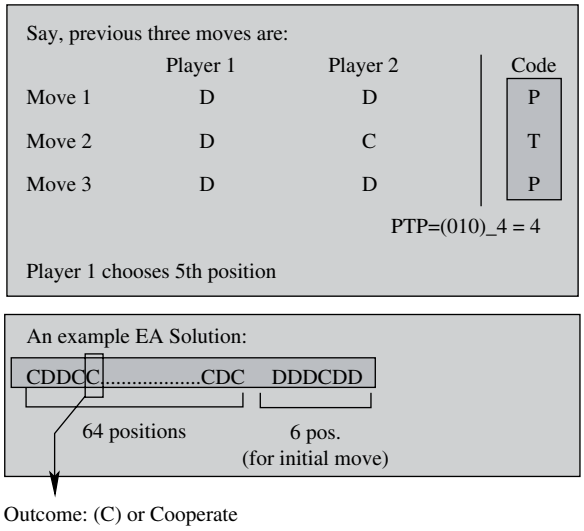

Fig. 2. Encoding a strategy for IPD used by Axelrod and also used here.

knowledge of past three moves, a player will then require one of two possible moves (a cooperate "C" or a defect "D") to be supplied by a GA solution. With these two options in each of 64 positions, a particular strategy can be defined by a 64-bit binary GA string of C (cooperate) and D (defect), where the ith C or D corresponds to the ith behavioral sequence dictated by the three-letter sequence of past three moves. Since a particular move depends on the previous three moves, the first three moves in a game are undefined in the above scheme. To account for these moves, six bits (with C and D, initially assigned at random) are appended to the above 64-bit string to specify a strategy's *premises*, or assumption about the pre-game behavior. Together, each of the 70-bit strings thus represent a particular strategy, the first 64 are used for rules and the next six are used for the premises. Fig. 2 shows such an example EA string. For the example string in the figure, the three-letter code comes out to be PTP for the previous three initial moves (given in the figure). This decodes to four, thereby meaning that player 1 should play the $(4+1)$ or 5th move (decoded value starts at zero, hence the addition of one) specified in the first 64-bit GA string. In this case, the fifth bit is C, meaning that the player 1 will cooperate.

Axelrod used the above encoding scheme to find optimal strategies using a single-objective genetic algorithm. He found that from a random start, the GA discovered strategies that not only performed quite well, but also beat the overall performance of 'Tit for Tat' strategy, mentioned earlier.

The encoding scheme used in this paper is the same as that mentioned above. However, in addition to a single-objective EA, we use an evolutionary multiobjective algorithm (EMO) to find optimal strategies. For most part of the paper, the following two objectives are chosen: 1) maximization of the self-score and 2) minimization of the opponent's score. Here the opponent's score means the cumulative score of all opponents, when playing against a particular strategy.

III. Using Multiple Objective Evolutionary Algorithms

Most studies of IPD considered a single objective of maximizing a player's own score. In this paper, for the first time, we treat the problem as a bi-objective optimization problem of maximizing the player's own score and simultaneously minimizing the opponent's score.

A. Motivation

The original formulation of the prisoner's dilemma game was represented as a single-objective optimization problem; that is, the main purpose was to find a strategy which maximizes a player's self-score. Such an objective will result in a single optimal strategy against a given set of opponents which would be good in terms of the player's own score, but will not care anything about opponents' scores. If in a game, a player plays with a strategy that is supposed to credit the player with a score of 400 (say) and also simultaneously allows an opponent to score 395 (say), and although the player wins the game, the satisfaction may be not enough. Moreover, often such games can be noisy and may involve some uncertainties. In such scenarios, the deterministic optimal strategy may not be very pragmatic. Such consideration for uncertainties calls for a *robust* optimization technique [21], [22]. But, in this paper, we consider a direct approach of introducing a second objective of minimizing the cumulative opponent's score to hopefully arrive at strategies that will keep a safe margin between self and opponents' scores. Thus, the IPD problem is re-looked here as a bi-objective optimization problem.

There is another advantage of casting the problem as a bi-objective optimization problem. Since the prisoner's dilemma game is a nonzero sum game, it is possible that there is a tradeoff between these two objectives, meaning that a strategy good for maximizing one's own score may result in a relatively high score of its opponents and vice versa. Later, we will show that this is actually the case for the problem chosen in this paper. Therefore, using a multiobjective evolutionary approach may actually allow us to find a set of tradeoff optimal strategies that can provide us with a better insight to the optimal strategies of playing the game as compared to a single-objective formulation. This is because using multiple conflicting objectives not one but a number of tradeoff optimal solutions can be found. These nondominated tradeoff solutions so obtained can then be analyzed to look for any pattern or insights about optimal strategies for the IPD. If any such patterns are discovered, they would provide important sub-strategies (rules or recipes) for playing the game for maximum self-score and minimum opponents' score.

In many practical situations, that can be modeled as a game, the issue of balancing one's own payoff against the opponent's can become important. This frequently arises in war-type situations, where one side may try to magnify the difference in payoff for propaganda purposes. In [23], for example, such a situation arising in missile defenze is presented. In that paper, the co-evolution in prisoner's dilemma, in which there can be intermediate levels of cooperation, has been studied.

B. NSGA-II Algorithm

For multiple objective optimization we use the NSGA-II algorithm [24]. NSGA-II has been successfully applied to many other multiple objective optimization problems [25] as well. Interested readers may refer to the original study of

NSGA-II for getting an algorithmic concept of an EMO procedure. A source code (in C programming language) of NSGA-II is also available from http://www.iitk.ac.in/kangal/soft.htm.

IV. Simulations and Test Cases

Both single-objective EA and EMO were used for getting optimal strategies. The simulation for both the algorithms followed these steps. In each generation, a certain number of strategies were generated, and each strategy was made to play against 16 other players (narrated in the Appendix), but a strategy did not play against itself. These opponent strategies have been used extensively in previous studies on IPD. Each game consisted of 150 moves. Then the strategies were sorted in the decreasing order of their cumulative scores, and the next generation was created using a recombination and a mutation operator. The payoff matrix was the same as that shown in Fig. 1.

Clearly, in one particular game using the payoff values shown in Fig. 1, a player can score a maximum of (5×150) or 750 [if (s)he always defects, and the opponent always cooperates], and a minimum of 0 [if (s)he always cooperates, while the opponent always defects]. None of these two extremes is usually achieved in practice. According to Dawkins [3], a more useful measure of a strategy in the context of IPD is how close it comes to the *benchmark score*, which is the score a player will have if both the players always cooperate. In this case of the IPD problem, the benchmark score is found to be (3×150) or 450 in this case. For example, if the score of a player, averaged over all the players (s)he played, is 400, then (s)he has scored 89% of the benchmark score. This is a more useful way of denoting the score of a player, since it is more independent of the particular payoff matrix used, as well as the number of players against which the player played. In all the results presented in the next section, we will refer only to the average score of a player in a game, or the score as a percentage of the benchmark score.

V. Simulation Results Using Single-Objective EA

First, we present the results obtained using a single-objective EA. Since solutions are represented using a bit string, we have used the usual binary tournament selection operator [25], a single-point crossover operator, and a bitwise mutation operator [2]. In all simulations, we have used a crossover probability of 0.9 and mutation probability of $1/70$, so that, on an average, only one bit in a string of 70 bits gets mutated at a time.

The single-objective EA used is quite similar to the one used by Axelrod [1]. Two independent runs of the single-objective EA were performed: one, in which the self-score of the player was maximized, and the other in which the opponent's score was minimized. For each of the runs, the population size was fixed at 40 and the number of generations was 20 000, although the performance stabilized much earlier. The plot of the scores for the first 200 generations when the EA was run are shown in Figs. 3 and 4.

For maximizing the self-score alone using a single-objective EA, the fitness measure of a sample is its self-score, hence the fitness score is to be maximized. As is clear from the graphs, in the first case the mean fitness increases steadily with generation counter, and around 200 generations the maximum self-score of a solution in the population is 441, which is 98% of the benchmark score. When the EA is run for longer generations, the maximum fitness converges at 446 (99.1% of the benchmark score) and does not increase further. This is a typical trend observed in optimization studies, including using GAs, in which the progress happens fast in the beginning followed by a relatively slow progress, as it becomes more and more difficult to find better solutions near the optimal solution in difficult optimization problems.

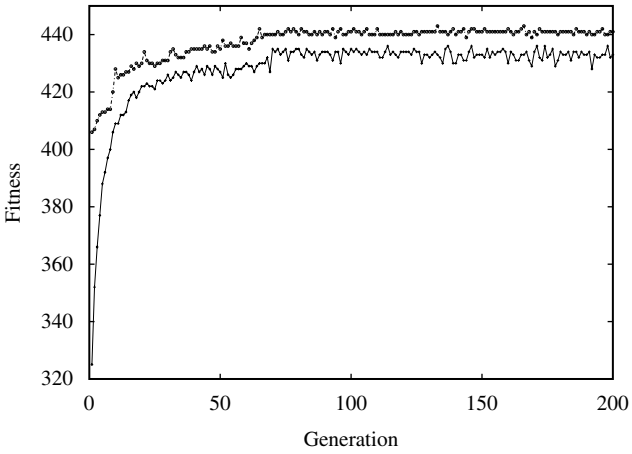

Fig. 3. Plot of the mean fitness (shown in solid line) and maximum (self-score) fitness (shown in dotted line) of population in the first 200 generations, when self-score is maximized.

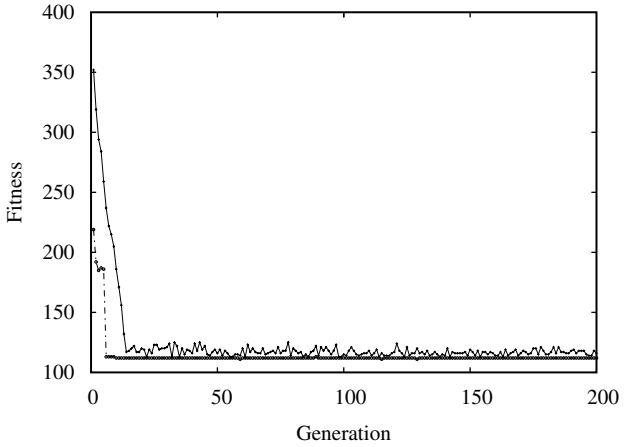

Fig. 4. Plot of the mean fitness (shown in solid line) and minimum (opponent's score) fitness (shown in dotted line line) of population in the first 200 generations, when opponent score is minimized.

When the obtained optimal strategies are fielded in a round robin tournament, these strategies win with a big margin. Tables I and II show the outcome (average of 20 runs) of two tournaments. In the first tournament, there are 16 strategies and "Tit for Tat" is the winner with an average score of 387 (with a margin of 12.2% of the benchmark score). In the second tournament, when the single-objective optimal strategy obtained by the maximization of the self-score is fielded, it wins by a huge margin, scoring as high as up to 99.1% of the benchmark score. This is in line with the results obtained by

TABLE I
TOURNAMENT 1 WITH THE 16 PLAYERS

Player	Average score
Tit for Tat	387
Soft Majority	379
Tit for two tats	379
Spiteful	376
Hard Tit For Tat	370
Always Cooperate	359
Periodic Player CCD	354
Naive Prober	353
Pavlov	351
Periodic Player CD	351
Remorseful Prober	351
Random Player	323
Hard Majority	317
Suspicious Tit for Tat	310
Periodic Player DDC	309
Always Defect	305

TABLE II
TOURNAMENT 2 WITH THE SINGLE-OBJECTIVE OPTIMUM STRATEGY INCLUDED

Player	Average score
Strategy SO	446
Tit for Tat	390
Hard Tit For Tat	374
Soft Majority	369
Tit for two tats	369
Spiteful	363
Naive Prober	360
Remorseful Prober	357
Always Cooperate	350
Periodic Player CCD	335
Pavlov	334
Periodic Player CD	334
Suspicious Tit for Tat	318
Hard Majority	311
Random Player	309
Periodic Player DDC	296
Always Defect	296

Axelrod. We refer to the strategy obtained by maximizing the self-score as "Strategy SO."

When the opponent score alone is minimized with a single-objective EA, the minimum fitness stabilizes at 112 (24.9% of benchmark score), as shown in Fig. 4. The strategies so obtained perform poorly in a round robin tournament (their performance is quite similar to that of the "Always Defect" strategy). We refer to this strategy as "Strategy SO-min." Table III shows the average score of the players when this strategy is included in the tournament. It can be seen that this strategy performs as bad as the "Always Defect" strategy. As such, it seems that there is little incentive in minimizing the opponent score alone. However, this objective along with maximizing the self-score objective has a remarkable benefit, which we shall discuss in Section VI.

A. Niched EA

Before leaving the single-objective optimization study altogether, we apply a single-objective GA with a niching-cum-speciation strategy to investigate whether this method could produce better strategies than that obtained by the previous algorithm. The use of a niching strategy, in effect, reduce the selection pressure introduced by the selection operator of an EA. Thus, a niched EA is expected to maintain diversity in the population, thereby facilitating an increased chance of convergence to one or more optima [26]. We also employ a speciation strategy by restricting mating between similar solutions [27]. The algorithm used niching and speciation in the sample space, together with a stochastic universal selection (SUS) operator for maximizing the self-score. The niching procedure is the *sharing* approach [26], which requires a user-defined parameter σ_{share}. We have followed the guidelines provided in the literature [28] and used different σ_{share} values, ranging between 5 and 30. The mating restriction operator ensures that two parent solutions used for crossover has at least σ_{share} bits in common. If no two strings with above requirement were found in the population, the crossover is not performed and the first parent is simply copied in the offspring population.

All other EA parameters were the same as before. The strategies obtained using the niching-cum-speciation procedure to maximize the self-score were found to be inferior to the previous single-objective procedure. The maximum fitness of the final population obtained was in the range 410–420 (whereas the earlier EA run without niching and speciation found a maximum self-score of 446), and the mean score of the population settled around 370 in a long run of this algorithm. This shows that using the niching-cum-speciation strategy in single-objective GA of maximizing self-score is not found to be helpful in improving the quality of the solution to this problem. Although a niching operator helps to maintain diversity in a population and a speciation operator helps to create meaningful solutions, an appropriate setting of the additional parameter (σ_{share}) becomes an important task. Despite trying with a wide range of σ_{share} values, our limited study is unable to produce any solution having a better self-score than that obtained without the use of any niching or speciation operator. The sharing scheme used here needed a proportionate selection operator (SUS used here) that introduces a selection pressure for the better population members, which is difficult to control. Although the solution obtained by the niching-cum-speciation EA is not a bad solution (better than the originally known "Tit for Tat" solution having an average score of 387), it is inferior to the solution obtained without these additional operators. The controlled selection pressure achievable with the tournament selection and EA parameters used in the original EA run was able to find a better solution.

In the next section, we use a different diversity-preserving strategy in which the problem is cast as a bi-objective problem with two apparently conflicting objectives in the hope of finding better scoring strategies of the IPD problem without requiring to supply any new additional parameter.

VI. SIMULATION RESULTS USING MULTIOBJECTIVE EA

The parameters used in NSGA-II are as follows: population size of 200 and maximum number of generations of 20 000. Optimized results were found much earlier, but to make sure that no further improvement is possible, we have run NSGA-II

TABLE III
TOURNAMENT 3 WITH THE SO-MIN STRATEGY INCLUDED

Player	Average score
Tit for Tat	372
Soft Majority	375
Tit for two tats	365
Spiteful	363
Hard Tit For Tat	356
Naive Prober	341
Always Cooperate	337
Remorseful Prober	336
Periodic Player CCD	335
Periodic Player CD	334
Pavlov	344
Random Player	309
Hard Majority	306
Suspicious Tit for Tat	300
Always Defect	296
Strategy SO-min	296
Periodic Player DDC	296

till 20 000 generations. In all cases, we have performed more than one simulations using different initial populations and obtained similar results. Here, we only show the results of a typical simulation run.

A. Evolution of Pareto-Optimal Strategies

Starting from a random population (marked with "+"), the strategies ultimately converged to a nondominated front shown using "x" in Fig. 5. The figure shows that the NSGA-II is able to successfully steer the initial population towards the Pareto-optimal region iteratively with generation and is able to find a good range of tradeoff strategies. The figure also shows that there is a remarkable tradeoff relationship between the two objectives: larger values of self-score and smaller values of opponent's score. The reader should note that we refer to Pareto-optimal strategies here only with respect to optimization over the strategies used in this paper (given in the Appendix), rather than with respect to optimization over the space of all possible strategies. However, the opponent strategies used in this paper are some of the commonly used strategies in the IPD literature. If any other strategy must be played against, a similar optimization task can be performed by including it in the list of opponent strategies. Nevertheless, this paper demonstrates the usefulness of including an additional objective in finding optimal IPD strategies.

After applying the NSGA-II to obtain the nondominated front, next we employed a simplified local search procedure (changing the bit positions from C to D or D to C systematically one at a time till no further improvement is possible) from each member of this front with a composite single objective function derived from the location of different points on the obtained frontier [25]. It was found that there was little or no improvement on the solutions. Therefore, the nondominated front found using NSGA-II is indeed very close to the true optimal front. The use of local search from EMO solutions ensures that the final solution is at least locally optimal.

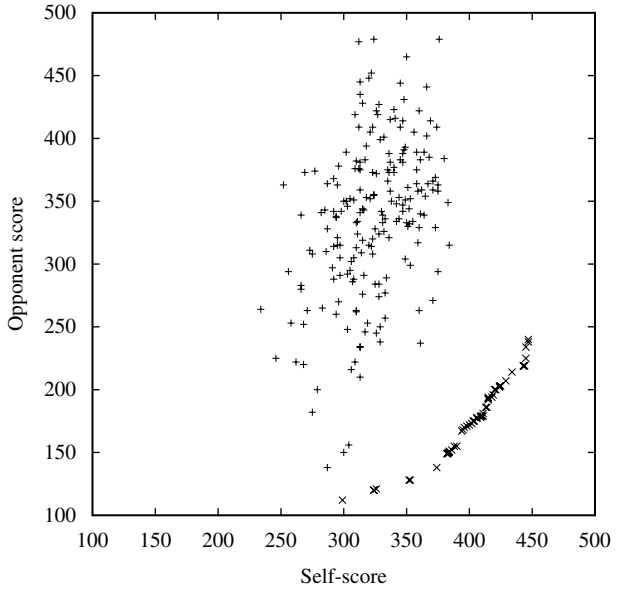

Fig. 5. Initial random solution (shown with "+") and the nondominated front (shown in "x"), when NSGA-II is run for 20 000 generations.

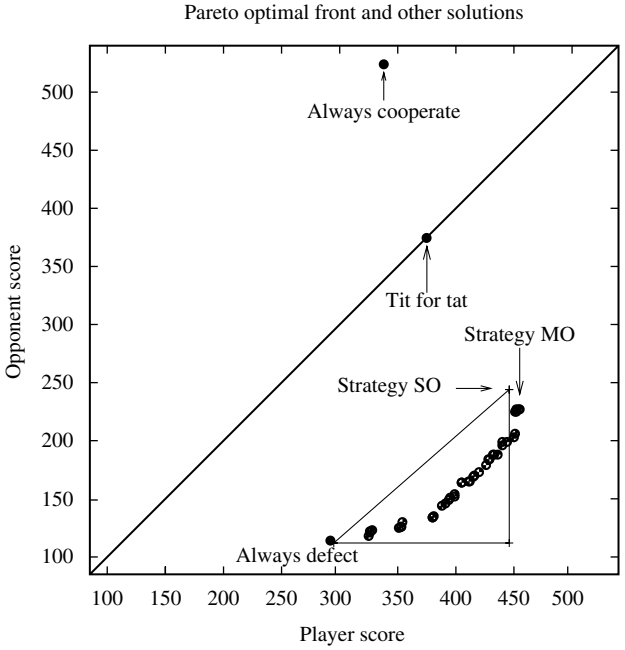

Fig. 6. Nondominated solutions, together with the single objective EA results (the upper and the left vertexes of the triangle) and a few other strategies.

In Fig. 6, the nondominated front obtained with NSGA-II and solutions of a few other strategies, including the single-objective EA results (discussed earlier), are shown. Table IV shows five different tradeoff solutions obtained using NSGA-II. The most significant outcome of the EMO is, however, the evolution of strategies that perform much better than those obtained using earlier methods. The strategy with the maximum self-score (the maximum score, 451 is slightly better than that for the optimal strategy obtained using single-objective EA, 446) had a mean opponent score (214) that was significantly lower than that for the single-objective

TABLE IV
FIVE CHOSEN STRATEGY STRINGS FROM NSGA-II SOLUTIONS, INCLUDING THE BEST SELF-SCORE (FIRST ROW) AND WORST OPPONENT SCORE (LAST ROW)

Strategy string	Self-score	Opponent score
DDDDDDCCCDDCDDDDDDCDDDDCDCCDDCDDDDCDCCCDDCDCDCDCDDCCDDDDCDDDDDCCCDCCDD	455	227
DDCCDDDDCDDDDDDDDDCDDDDCDCCDDCDDDCCDCCCDDCDCDCDDDDCCDDDDCDDDDDCCCDCCCD	415	169
DDCDDDDCCDCDDDDDDDCDDDDCDCCCDCDDDCCDCCCDDCDDDCDDDDCCDDDDCDDDDDCCCDCCDC	381	135
DDCDDDDCCDCDDDDDDDCDDDDCDCCCDCDDDDCDCCCDDCDCDCDDDDCDDDDDDDDDDCCCDCCCD	328	123
DDCDDDDDCDDCDDDCDDCDDDDCDCCDDCDDDDCDCCCDDDDDDCDCDDCDDDDDCDCDDDCCCCCCDC	298	112

TABLE V
TOURNAMENT 4 WITH STRATEGY MO INCLUDED

Player	**Average score**
Strategy MO	448
Tit for Tat	391
Hard Tit For Tat	375
Soft Majority	370
Tit for two tats	370
Spiteful	363
Naive Prober	358
Remorseful Prober	344
Always Cooperate	337
Periodic Player CCD	336
Periodic Player CD	334
Pavlov	334
Suspicious Tit for Tat	319
Hard Majority	312
Random Player	310
Periodic Player DDC	296
Always Defect	296

TABLE VI
TOURNAMENT 5 WITH BOTH STRATEGY MO AND STRATEGY SO INCLUDED

Player	**Average score**
Strategy MO	434
Strategy SO	428
Tit for Tat	394
Hard Tit For Tat	378
Naive Prober	364
Soft Majority	361
Tit for two tats	361
Spiteful	351
Remorseful Prober	349
Always Cooperate	330
Suspicious Tit for Tat	326
Periodic Player CCD	319
Periodic Player CD	319
Pavlov	319
Hard Majority	307
Random Player	295
Always Defect	288
Periodic Player DDC	285

optimal strategy (244). Fig. 6 shows the single-objective optimum strategy (Strategy SO) and the multiple-objective optimal self-score strategy (Strategy MO). Strategy MO so obtained not only outperformed other strategies in a round robin tournament (see Tables V and VI), but also defeated the Strategy SO (Table VI). This clearly shows that the addition of minimizing opponent's score as a helper objective in an EMO is able to find better self-score strategies as compared to the single-objective EA of maximizing self-score alone. Since an EMO maintains a good diverse population due to the consideration of two conflicting objectives and the use of a diversity preserving mechanism, the search power of EMO is usually better than that in a single-objective EA. In complex optimization problems in which the search of the individual optimal solution is difficult using a single-objective optimization algorithm, a two or more objective consideration may lead to a better optimum, in general.

Note that the score for Strategy MO given in Table VI is slightly less than the self-score of this strategy when it is obtained using the NSGA-II algorithm. This is because the scores presented in each tournament are averaged over several tournaments (this is done to negate the effect of the variation in scores due to the presence of randomized strategies in the tournament), whereas when computing the optimal strategy using NSGA-II algorithm, the tournament is held only once in each generation. Therefore the score for the Strategy MO is slightly lower in the tournament scores.

The other extreme solution on the Pareto-optimal front (the one corresponding to the minimum opponent's score) is the same as that obtained by minimizing the opponent's score in Table III, and has the same performance as the "Always Defect" strategy. Interestingly, even though many bit positions in the string for this strategy are C (shown in the last row of Table IV), they are not used during the round robin tournament. The most frequently used bit positions for this strategy are D, thereby behaving almost like the "Always Defect" strategy.

B. Sensitivity of the Strategy With the Best Self-Score

To further analyze the best self-score strategy for the frequently repeated bit positions, we performed the following experiment. If the position i is frequently used, and the strategy at this position in the strategy string is C, it is changed to D and vice versa. Only one position in the strategy string is changed at a time. This experiment was carried out on the Strategy MO.

When this is done for positions $0, 1, 4, 5$, etc., where the original strategy is to defect (D) and in the new strategy string

the strategy is changed to cooperate (C), then there is a slight decrease in the self-score of the player from 448 to around 440. This is expected, since at certain moves in the game the player gets the sucker's payoff (player 1 cooperates and player 2 defects). In the round robin tournament, the modified strategy still wins by a large margin against the other 16 players, but does not perform as good as the original strategy.

A more interesting situation arises when the strategies for the bit positions 29 and 63 are changed from C to D. In the first case, the self-score of the player falls drastically to 361 and its rank in the round robin tournament falls to five, while in the latter case the self-score falls to 370 and the rank to two. A closer analysis of the frequent usage of bit positions revealed that 29 is used most frequently against the players "Soft Majority" and "Hard Majority." The position 29 decodes to TRT, which means that the opponent cooperated on all the previous three moves, while player 1 defected on the first and the third moves and cooperated on the second move. Such a strategy pattern fools the "Majority" players by giving them the illusion that the most moves played against them are C. Thus the "Majority" players are exploited by the evolved strategies by this pattern of moves. When the strategy at 29 is changed from C to D, the modified strategy and the "Majority" players indulge in a series of mutual defections, which considerably decreases the self-score of the modified player. When the strategy at position 63 is changed from C to D, then the modified strategy enters into mutual defections with the "Tit for Tat" and the prober players, again leading to a significant decrease in the self-score.

This experiment shows that it is very important to cooperate at crucial points, to either sustain a chain of mutual cooperation (as is the case with position 63) or to fool the opponent player (the case with position 29). This sensitivity analysis also demonstrates that the obtained best self-score strategy by NSGA-II is at least locally optimal, as a local perturbation of this strategy makes a deterioration of the self-score.

C. Maximizing Difference Between Self-Score and Opponent Score

One possible way of incorporating both the above-mentioned objectives is to maximize the difference between the self-score and the opponent score. We carried out this single-objective optimization using a genetic algorithm. The parameters of this single-objective optimization are the same as those mentioned in Section V. The maximum difference between the two scores settles down to 230 when the EA is run for a large number of generations. However, the best self-score strategies obtained by this optimization method have a low self-score of 370. When this strategy (referred to as "MaxDiff" strategy) are fielded in a round robin tournament, their performance is well below the strategies obtained by maximizing self-score (in Sections V and VI so far), as shown in Table VII. In Fig. 7, this strategy is shown together with the Pareto-optimal front obtained using NSGA-II. First of all, the plot shows that the MaxDiff strategy is dominated by the Pareto-optimal front, indicating that it does not perform better than some of the strategies obtained using NSGA-II. Hence it is not sufficient to simply maximize a single goal of the difference between the self-score and the opponent scores to get good strategies for IPD. The difference between the self and opponent scores are marked for a few selected solutions on the NSGA-II frontier. It is clear that the intermediate portion of the front possesses strategies having a large difference in scores. The single-objective GA was not able to find the strategy having the maximum difference (a value of 248) in self and opponent scores found by NSGA-II. This once again demonstrates that the NSGA-II diversity-preserving feature is able to find a better strategy than a single-objective GA. It is this positive aspect of a multiobjective EA applied to a difficult problem that we highlight in this paper. There is another important benefit for finding multiple tradeoff solutions which we discuss next.

TABLE VII
TOURNAMENT 6 WITH THE MAXDIFF STRATEGY INCLUDED

Player	Average score
Tit for Tat	372
MaxDiff	370
Soft Majority	369
Tit for two tats	369
Spiteful	362
Hard Tit For Tat	357
Always Cooperate	350
Naive Prober	341
Remorseful Prober	336
Periodic Player CCD	336
Pavlov	334
Periodic Player CD	334
Hard Majority	311
Random Player	308
Suspicious Tit for Tat	301
Always Defect	295
Periodic Player DDC	295

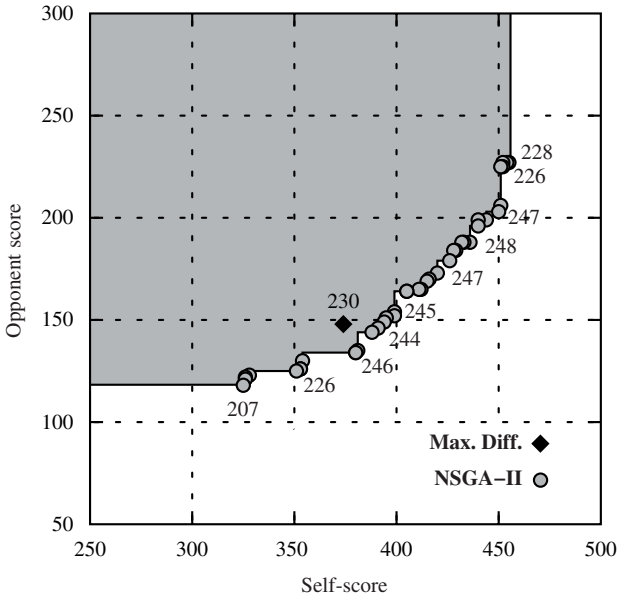

Fig. 7. MaxDiff strategy shown along with the Pareto-optimal front.

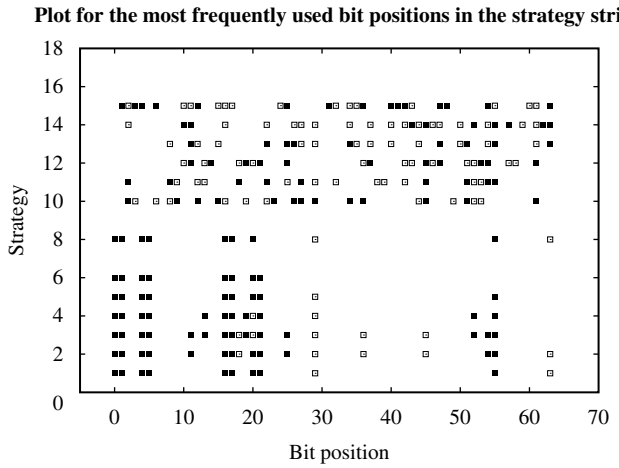

Fig. 8. Plot of the frequently used bit positions in the strategy strings. The cooperative moves are shown with white boxes, while the defecting moves are shown in black boxes. The upper half of the plot corresponds to frequently used bit positions for players with randomly chosen strategies, whereas the bottom half corresponds to those of the Pareto-optimal strategies.

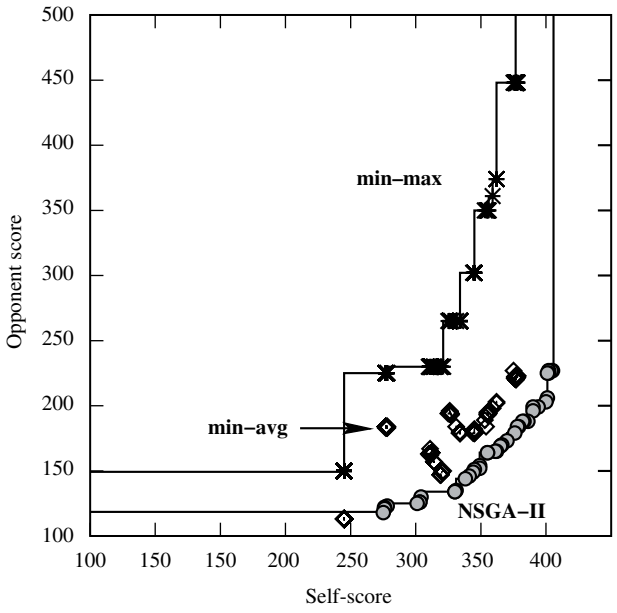

Fig. 9. Pareto-optimal front obtained when maximum of the opponent score is minimized (maximum opponent score is represented by 'X' and corresponding average score is represented by diamonds) against the Pareto-optimal front obtained earlier (shown in circles).

D. Common Properties of Pareto-Optimal Strategies

Nondominated strategies obtained by the proposed procedure are locally optimal and each corresponds to a certain tradeoff between the two conflicting objectives considered in this paper. Since they are all high-performing solutions, it is expected that they will possess some interesting properties that may be common to all such solutions. An understanding and deciphering of such properties has revealed interesting insights in a number of optimization tasks [20]. We attempt to perform such a study with the tradeoff solutions obtained for the IPD problem here.

To have a closer look at the tradeoff strategies, the number of times each bit position in the string was used in a round robin tournament was recorded, and plotted for different strategies. Fig. 8 shows the combined plot for six Pareto-optimal strategies (chosen from Fig. 6), and for six random strategies (for comparison). In the plot, the frequency distribution for six Pareto-optimal strategies are given in the lower half (strategies marked as 1 to 6 with self-score decreasing along the positive y-axis), and for six randomly chosen strings in the upper half (strategies marked as 10 to 15). The cooperative moves are shown in white boxes, and the defecting moves are shown in black boxes. Only those bit positions that were used more than 20 times (out of 150 moves) in the round robin tournament are shown. In our particular encoding scheme, $DD = P = 0$, $DC = T = 1$, $CD = S = 2$, and $CC = R = 3$. The plot reveals that only a few of the bit positions of a strategy are used more than 20 times. Also, the Pareto-optimal strategies show some interesting similarities with respect to the usage of a particular bit position. For example, positions 0, 1, 4, 5, 17, 18, 21, and 55 turn out to be "Defecting" in all of the six Pareto-optimal solutions. There are also some trends in "Cooperating," such as 29 and 63 bit positions, coming out as common strategies of these high-performing solutions. We discuss a few of them in the following:

1) Bit Position 0 *Resulting in a Defect:* This decodes to *PPP*, i.e. both the players have been defecting with each other over the previous three moves. Since both players are defecting, it is expected that the player 1 should also defect as a good strategy for preventing the opponent's score to be high in the subsequent moves.

2) Bit Position 1 *Resulting in a Defect:* This decodes to *PPT*. The opponent defected on the first two moves, but did not do so in the third move, while player 1 defected in all the three moves. In this case, the strategy is to defect, so as to "exploit" the foolish opponent.

3) Bit Positions 4 and 5 Resulting in a Defect: These bit-positions decode to *PTP* and *PTT*, respectively. These cases are similar to the previous case, and again the moves are to defect to exploit the opponent.

4) Bit Position 17 Resulting in a Defect: This implies *TPT*. That is, player 1 has defected on all the previous three moves, but the opponent was foolish enough to cooperate. Clearly, in this case, player 1 should defect to exploit the opponent's foolishness.

5) Bit Position 55 Resulting in a Defect: This is implies *RTR*. This again a case of exploitation, since the opponent cooperated on all the previous three moves even though player 1 defected once. Hence the move in this situation is to defect.

6) Bit Position 63 Resulting in a Cooperation: This implies *RRR*, that is, the players have cooperated on all the previous three moves. Since both the players are cooperating, the best move in this case is to continue cooperating.

The eighth solution in the figure on y-axis is for the single-objective optimum strategy of maximizing self-score alone. Interestingly, the frequently used moves in this strategy is similar to the first strategy of the bi-objective Pareto-optimal solutions (with the highest self-score). Thus, a recipe for

TABLE VIII
SCORES FOR THE LEAVE-ONE-OUT TEST PROCEDURE, AND ITS COMPARISON WITH THE SCORES OF STRATEGY MO

	Leave-One-Out scores			Strategy MO scores		
Player	**Self-score**	**Opponent score**	**Result**	**Self-score**	**Opponent score**	**Result**
Always Cooperate	600	225	win	746	6	win
Always Defect	147	162	lose	147	162	lose
Tit for Tat	447	447	draw	446	446	draw
Suspicious Tit for Tat	445	450	lose	444	449	lose
Pavlov	443	88	win	452	77	win
Spiteful	151	161	lose	151	161	lose
Random Player	396	101	win	476	71	win
Periodic Player CD	448	78	win	448	78	win
Periodic Player DDC	344	114	win	347	107	win
Periodic Player CCD	450	325	win	547	57	win
Tit for Two Tats	158	148	win	599	224	win
Soft Majority	590	230	win	590	230	win
Hard Majority	585	235	win	588	233	win
Hard Tit for Tat	157	162	lose	302	482	lose
Naive Prober	441	446	lose	439	444	lose
Remorseful Prober	356	106	win	415	285	win

maximizing self-score by minimizing the opponent's score is to learn to defect when the opponent is either defecting or foolishly trying to cooperate when player 1 is continuously defecting. Another recipe to follow is to cooperate with the opponent when the opponent has indicated its willingness to cooperate in the past moves.

Another matter to note is that, as the self-score decreases (as solutions go up on the y-axis from 1 to 6), the strategies chose to become more defecting and the frequency of cooperation reduces. To minimize the opponent's score, the payoff matrix indicates that player 1 should defect more often. When this happens, self-score is also expected to be low against intelligent players, because both players will engage in defecting more often.

For random strategies (marked as 10 to 15 in Fig. 8), no such pattern is observed. It can be seen that, for the Pareto-optimal strategies, most of the bit positions are either sparingly used, or are not used at all. For example, the strategy with the least self-score always makes defecting moves, even though there are many C in its strategy string, showing that it behaves almost like the Always Defect strategy, which we have discussed before as well.

E. Maximizing Self-Score and Minimizing Maximum of Opponent Score

We also carried out another simulation in which we used NSGA-II to minimize two other objectives: maximize self-score, and minimize the maximum of the opponent scores (in the previous case, we had minimized the average score of the opponents, instead). The Pareto-optimal front obtained is shown in Fig. 9 by marking the obtained strategies (maximum opponent score) with a star. When the average score over all opponent players are computed and plotted for these strategies (marked with a diamond), they are found to be dominated by previous Pareto-optimal solutions. The second objective value of Pareto-optimal solutions using this method is worse than before, as the maximum of the opponents' scores is going to be always more than average of opponents' scores. A good spread in solutions is still obtained, but since this new objective is nondifferentiable and hence may not represent a smooth landscape, the obtained front in this case is not as smooth as before. As discussed above, the solutions obtained are also inferior from earlier solutions. Based on this paper, we may conclude that minimizing the average score of opponents is a better optimization strategy than minimizing the maximum score obtained by any opponent.

F. Cross-Validation of the Optimization Procedure

Since we are evolving optimal strategies by optimizing against a fixed set of 16 players, it is possible that the optimal strategies so obtained are "overspecialized;" that is, the evolved strategies may not perform well against other players which were not used during the optimization phase. To cross-validate the effectiveness of the optimization procedure, we performed the "Leave-One-Out" test, which is a standard test used for evaluating machine learning techniques. In this test, during the optimization phase, one of the 16 players is not used, and then the evolved optimal strategy is pitted against this omitted player in a tournament. We compare the performance of the optimal strategy against the left out player, to the performance of the Strategy MO (which is obtained when all the players are included in the optimization phase) against this player. The optimization procedure used in the Leave-One-Out test is the same as before, i.e. NSGA-II evolved up to 20 000 generations. The results are are shown in Table VIII for each of the 16 opponents left out one at a time.

The table shows that the win-lose-draw record in the Leave-One-Out test procedure is exactly the same as that for the Strategy MO. This shows that our optimization procedure does not lead to overspecialization of the evolved strategies. The average of self-score over all the players for the Leave-One-Out phase is 385, which is considerably less than that for Strategy MO (446); however, the average of the opponent score (217) is almost the same as that for Strategy MO

(220). It is expected that the self-score in the Leave-One-Out phase will be lower than that of Strategy MO. However, even with a lower self-score, the win-loss record remains exactly the same. The difference in scores in the case of Random Player and Remorseful Prober can be attributed to random factors, since both these strategies are driven by randomness in them. Difference in scores also arises in the case of more "cooperating" players like Always Cooperate, Periodic Player CCD, and Tit for Two Tats. The evolved strategy in these cases did not exploit these cooperating strategies to the maximum possible extent. This indicates that it may be important to have prospective cooperating players during the optimization phase.

Before concluding this section, it will be worthwhile to compare the performance of the optimal strategies obtained using our procedure vis-a-vis the Tit for Tat strategy against unseen opponents. In the Leave-One-Out procedure, the strategy with the highest self-score has an average score of 385, which is almost the same as the score of 387 for the Tit for Tat strategy (see Table I). But whereas the average self-score of Tit for Tat strategy is usually the same as the average opponent score (see Fig. 6), the average opponent score in the Leave-one-out procedure is 217, which is much less than the average self-score. Once again this shows the robustness of our procedure: on average, the optimal strategies will perform at least as well as the Tit for Tat strategy with respect to self-score against unseen opponents, and much better with respect to the opponent score.

VII. Conclusion

We have presented a new paradigm for searching optimal strategies in the age-old yet important IPD problem using multiobjective optimization procedures. Such an optimization strategy has not been used before in the literature for this problem. It has been revealed that such a solution strategy has several advantages over the existing single-objective methods for finding useful optimal game-playing strategies.

1) The use of a bi-objective consideration of maximizing a player's own payoff and minimizing the average payoff of opponents has been able to find a better game-playing strategy than the sole consideration of maximizing a player's own payoff.
2) The obtained tradeoff frontier has outperformed many existing strategies (those given in the Appendix), which includes the well-known "Tit for Tat," when all these strategies are used in the optimization procedure.
3) The suggested optimization procedure has been found to be robust, in the sense that even if one of the 16 opponents were not considered in the optimization process, the resulting strategy is still able to outperform the left-out strategy, thereby showing the generalizing ability of the suggested optimization procedure.
4) The use of a pair of objectives (maximizing own payoff and minimizing average opponent's payoff) has outperformed other useful objective pairs considered in this paper.
5) An investigation of the sub-strategies commonly existing among obtained tradeoff solutions has resulted in a number of useful information about playing the IPD game successfully. For example, we find that all strategies on the nondominated front defect at bit positions 0, 1, 4, 5, 17, and 55. An analysis of these sub-strategies has provided us salient insights for playing the IPD with satisfaction. Such important information is not possible to achieve from a single solution obtained by using a single-objective EA alone. Such information is useful in playing the game efficiently.
6) Another interesting observation obtained from the results is that it is important to cooperate at certain preconditions of the IPD game. We notice that when the strategy at position 29 (TRT) or 63 (RRR) is changed from C to D, there is a significant drop in the self-score of Strategy MO. We then analyzed the reason behind this drop and observe that cooperation is important at certain times either to fool the opponent into cooperating and then exploiting it, or to sustain a chain of mutual cooperation for gaining payoffs. For a one-shot game, the optimal strategy for the two players is clearly to defect. Our results provide insights on obtaining cooperative behavior with appropriate payoffs in the iterated version of the game.

Hopefully, there will be far-reaching implications of this paper in IPD problem, and such a systematic multiobjective optimization approach will find further application in other related game-playing problems in the near future.

There are several ways in which the work presented here can be extended. It will be interesting to examine the multiobjective evolution of IPD game strategies under the co-evolutionary setting, where each strategy is evaluated by placing it against other strategies in the population. One other approach that may be tried is to change the profile of the players when evolving the optimal strategy. Then it would be interesting to see which properties of the optimal strategy are stable with change in opponent pool and what properties are possibly required in a robust strategy. We leave these for future research.

Appendix

Details about the different strategies used in the round-robin tournament are given below.

1) **Always Cooperate**: Cooperates on every move.
2) **Always Defect**: Defects on every move.
3) **Tit for Tat**: Cooperates on the first move, then simply copies the opponent's last move.
4) **Suspicious Tit for Tat**: Same as Tit for Tat, except that it defects on the first move.
5) **Pavlov**: Cooperates on the first move, and defects only if both the players did not agree on the previous move.
6) **Spiteful**: Cooperates, until the opponent defects, and thereafter always defects.
7) **Random Player**: Makes a random move.
8) **Periodic player CD**: Plays C, D periodically.
9) **Periodic player DDC**: Plays D, D, C periodically.
10) **Periodic player CCD**: Plays C, C, D periodically.
11) **Tit for Two Tats**: Cooperates on the first move, and defects only when the opponent defects two times.
12) **Soft Majority**: Begins by cooperating, and cooperates as long as the number of times the opponent has cooperated

is greater than or equal to the number of times it has defected, else it defects.

13) **Hard Majority**: Defects on the first move, and defects if the number of defections of the opponent is greater than or equal to the number of times it has cooperated, else cooperates.
14) **Hard Tit for Tat**: Cooperates on the first move, and defects if the opponent has defects on any of the previous three moves, else cooperates.
15) **Naive Prober**: Like Tit for Tat, but occasionally defects with a probability of 0.01.
16) **Remorseful Prober**: Like Naive Prober, but it tries to break the series of mutual defections after defecting.

REFERENCES

[1] R. Axelrod, "The evolution of strategies in the iterated prisoner's dilemma," in *Proc. Genetic Algorithms Simulated Annealing*, Los Altos, CA: Morgan Kaufmann, 1987.

[2] D. E. Goldberg, *Genetic Algorithms for Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[3] R. Dawkins, *The Selfish Gene*, 2nd ed. New York: Oxford Univ. Press, 1989.

[4] M. Rubenstein and M. Osborne, *A Course in Game Theory*. Cambridge, MA: MIT Press, 1994.

[5] J. F. Nash, "Equilibrium points in n-person games," in *Proc. Nat. Academy Sci.*, vol. 36. 1950, pp. 48–49.

[6] D. Fudenberg and E. Maskin, "The folk theorem in repeated games with discounting or incomplete information," *Econometrica*, vol. 54, no. 3, pp. 533–554, 1986.

[7] R. Axelrod and W. Hamilton, "The evolution of cooperation," *Sci.*, vol. 211, no. 4489, pp. 1390–1396, 1981.

[8] R. Axelrod, *The Evolution of Cooperation*. New York: Basic Books, 1989.

[9] D. Fogel, "Evolving behaviors in the iterated prisoner's dilemma," *Evol. Comput.*, vol. 1, no. 1, pp. 77–97, 1983.

[10] M. Nowak and K. Sigmund, "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game," *Nature*, vol. 364, no. 6432, pp. 56–58, 1993.

[11] B. Beaufils, J. P. Delahaye, and P. Mathieu, "Our meeting with gradual, a good strategy for the iterated prisoner's dilemma," in *Proc. Artificial Life V: Proc. 5th Int. Workshop Synthesis Simulation Living Syst.*, Cambridge, MA: MIT Press, 1996, pp. 202–209.

[12] D. Bragt, C. Kemenade, and H. Poutré, "The influence of evolutionary selection schemes on the iterated prisoner's dilemma," *Comput. Econ.*, vol. 17, no. 2–3, pp. 253–263, 2001.

[13] D. Jang, P. Whigham, and G. Dick, "On evolving fixed pattern strategies for iterated prisoner's dilemma," in *Proc. 27th Conf. Australasian Comput. Sci.*, Dunedin, New Zealand, 2004, pp. 241–247.

[14] M. R. Frean and E. R. Abraham, "A voter model of the spatial prisoner's dilemma," *IEEE Trans. Evol. Comput.*, vol. 5, no. 2, pp. 117–121, Apr. 2001.

[15] H. Ishibuchi and N. Namikawa, "Evolution of iterated prisoner's dilemma game strategies in structured demes under random pairing in game playing," *IEEE Trans. Evol. Comput.*, vol. 9, no. 6, pp. 552–561, Dec. 2005.

[16] S. Y. Chong and X. Yao, "Behavioral diversity, choices and noise in the iterated prisoner's dilemma," *IEEE Trans. Evol. Comput.*, vol. 9, no. 6, pp. 540–551, Dec. 2005.

[17] K. Deb, P. Jain, N. Gupta, and H. Maji, "Multiobjective placement of electronic components using evolutionary algorithms," *IEEE Trans. Components and Packaging Technol.*, vol. 27, no. 3, pp. 480–492, Sep. 2004.

[18] J. D. Knowles, D. W. Corne, and K. Deb, *Multiobjective Problem Solving From Nature* (Springer Natural Computing Series). New York: Springer-Verlag, 2008.

[19] C. A. C. Coello, D. A. Van Veldhuizen, and G. Lamont, *Evol. Algorithms for Solving Multiobjective Problems*. Boston, MA: Kluwer, 2002.

[20] K. Deb and A. Srinivasan, "Innovization: Innovating design principles through optimization." in *Proc. Genetic Evol. Comput. Conf. (GECCO-2006)*, New York: The Association of Computing Machinery (ACM), 2006, pp. 1629–1636.

[21] J. Branke, "Creating robust solutions by means of an evolutionary algorithm," in *Proc. Parallel Problem Solving Nature (PPSN-V)*, 1998, pp. 119–128.

[22] K. Deb and H. Gupta, "Introducing robustness in multiobjective optimization," *Evol. Comput. J.*, vol. 14, no. 4, pp. 463–494, 2006.

[23] P. J. Darwen and X. Yao, "Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: Application to missile defense." *Int. J. Comput. Intell. Applicat.*, vol. 2, no. 1, pp. 83–107, 2002.

[24] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[25] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[26] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 1st Int. Conf. Genetic Algorithms Their Applicat.*, 1987, pp. 41–49.

[27] K. Deb, "Genetic algorithms in multi-modal function optimization," M.S. thesis, Department of Engineering Mechanics, University of Alabama, Tuscaloosa, AL, 1989.

[28] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 42–50.

Shashi Mittal received the Bachelor of Technology degree in computer science and engineering at the Indian Institute of Technology Kanpur, India. He is currently a graduate student at the Operations Research Center, Massachusetts Institute of Technology, Cambridge.

His research interests are in the area of combinatorial optimization and game theory.

Kalyanmoy Deb received the Bachelor's degree in mechanical engineering from the Indian Institute of Technology, Kharagpur, India. He received the Masters and Ph.D. degrees from the University of Alabama, Tuscaloosa, in 1989 and 1991, respectively.

He was with the Engineers India Limited, New Delhi. He is currently the Deva Raj Chair Professor at the Indian Institute of Technology, Kanpur, and currently visiting Helsinki School of Economics as a Finland Distinguished Professor. His main research interests are in the areas of computational optimization, modeling and design, and evolutionary algorithms. He has written two text books on optimization and has published or presented more than 240 papers in international journals and conferences. He has pioneered and is a leader in the field of evolutionary multiobjective optimization. He is Associate Editor and on the editorial board of a number of major international journals.

Dr. Deb is a Fellow of the Indian National Academy of Engineering, the Indian National Academy of Sciences, and the International Society of Genetic and Evolutionary Computation. He is the recipient of the prestigious Shanti Swarup Bhatnagar Prize in Engineering Sciences in India for the year 2005. He has also received the Thomson Citation Laureate Award from Thompson Scientific for having the highest number of citations in computer science during 1996–2005 in India. He was a recipient of the Fredrick Wilhelm Bessel Research award from the Alexander von Humboldt Foundation, Germany, in 2003.