Pareto coevolution: Using performance against coevolved opponents in a game as dimensions for Pareto selection

Jason Noble
Informatics Research Institute
School of Computing, University of Leeds
Leeds LS2 9JT, UK
jasonn@comp.leeds.ac.uk

Richard A. Watson
DEMO Lab, Computer Science Dept.
Brandeis University
Waltham, MA 02454, USA
richardw@cs.brandeis.edu

Abstract

When using an automatic discovery method to find a good strategy in a game, we hope to find one that performs well against a wide variety of opponents. An appealing notion in the use of evolutionary algorithms to coevolve strategies is that the population represents a set of different strategies against which a player must do well. Implicit here is the idea that different players represent different "dimensions" of the domain, and being a robust player means being good in many (preferably all) dimensions of the game. Pareto coevolution makes this idea of "players as dimensions" explicit. By explicitly treating each player as a dimension, or objective, we may then use established multi-objective optimization techniques to find robust strategies. In this paper, we apply Pareto coevolution to Texas Hold'em poker, a complex real-world game of imperfect information. The performance of our Pareto coevolution algorithm is compared with that of a conventional genetic algorithm and shown to be promising.

1 INTRODUCTION

One of the inherent problems with learning game strategies through self-play is a tendency for such strategies to be brittle—to be over-specialised to a particular area of strategy space—and to fail to find robust, general strategies (see, e.g., Pollack & Blair, 1998, for discussion). The potential for strategies to have intransitive superiority relationships is an important key for understanding why this might happen. That is, although some player A might be beaten by some other player B, and B may in turn be beaten by C, it may not be the case that C beats A (Cliff & Miller, 1995). The existence of such intransitive superiority relationships can mean that although a search method persistently finds strategies that are better than the last strategy, it fails to find a strategy that is good in general. Intransitive superiority relationships suggest that a problem domain is multi-dimensional, in the sense that being good against one strategy does not necessarily mean that you are good against another (Watson and Pollack, this volume).

An appealing notion in the use of evolutionary algorithms to coevolve strategies is that the population represents a set of different strategies against which a player must do well. Implicit here is the idea that different players represent different "dimensions" of the domain, and being a robust player means being good in many (preferably all) dimensions of the game. However, the idea that players represent dimensions of the game remains implicit in standard coevolutionary algorithms. Pareto coevolution makes the concept of "players as dimensions" explicit. By explicitly treating each player as a dimension, or objective, we may then apply established multi-objective optimization techniques—in particular, principles such as Pareto dominance—to find robust strategies. This may help to prevent the effects of intransitive superiority from interfering with the discovery of good general solutions, because multi-objective optimization promotes a set of players with a different balance of abilities rather than promoting the single best-on-average strategy. Pareto coevolution was explored by Watson and Pollack (2000), and follows from work relating coevolution and Pareto dominance (Ficici & Pollack, 2000). Pareto coevolution is also developed in the domain of the cellular automata majority problem by Ficici and Pollack (2001). In this paper, we apply Pareto coevolution to Texas Hold'em poker.

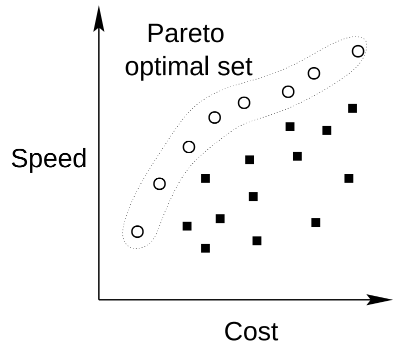

Figure 1: Solution points for a hypothetical car design problem, in which we want to maximize speed and minimize cost. The Pareto-optimal set is indicated.

1.1 PARETO SELECTION AND GAMES

A Pareto-optimal solution is one in which none of the relevant measurements or dimensions of quality or performance can be improved without reducing performance on one or more of the other dimensions. For example, if we were designing a car, and our goals were low cost and a high top speed, there might be a Pareto-optimal solution at $20,000 and 120 mph. This means that 120 mph is the fastest you can go for that price, and that $20,000 is the cheapest you can pay for that speed. An alternative design with the same price but a top speed of only 110 mph would clearly be inferior. However, there will almost always be more than one solution in the Pareto-optimal set of best possible compromises (see Figure 1). Perhaps there are also Pareto-optimal design possibilities at $25,000 and 130 mph, and at $15,000 and 100 mph. The spirit of the Pareto approach is not to somehow convert dimensions like speed and cost into a common currency in order to come up with the one true optimum, but to find all members of the Pareto-optimal set so that a human decision-maker, or some other method, can be allowed to choose between them.

Within the field of evolutionary computation, various methods of approximating the Pareto-optimal set have been proposed as tools for multi-objective optimization (for reviews see, e.g., Fonseca & Fleming, 1995; Horn, 1997). The details differ, but, in essence, Pareto dominance is used as a selection criterion. Candidate solution A Pareto-dominates solution B if A is at least as good as B on all dimensions, and better than B on one or more. Pareto selection involves choosing the *non-dominated* solutions for reproduction.

Pareto selection is typically carried out with respect to a small number of dimensions, as in the car example above. This paper seeks to apply Pareto selection to the domain of games (von Neumann & Morgenstern, 1953) by using each player in an evolving population as a dimension, or objective, to be optimized—hence, Pareto coevolution.

Given a particular game, and a way of representing strategies in that game, we could list every possible strategy. We could also observe the performance of each strategy against every other, and the matrix so derived would allow us to see that some strategies Pareto-dominate others, e.g., that A performs as well as B when playing C, D, and E, and is better than B when playing F. We could then spell out the membership of the Pareto-optimal set. It is important to realize that the set might include surprising members: perhaps a strategy that does very poorly on average would nevertheless be included because of its exceptional performance against just one opponent.

The brute-force approach of calculating a performance matrix of all-against-all will work for a sufficiently simple game with a small number of possible strategies, but it obviously will not be feasible for games of any complexity. The size of the performance matrix will be equal to the number of possible strategies squared, and reliably calculating each entry in the matrix will require many trials if the game includes a stochastic element.

We have utilized a population-based coevolutionary approach, in which individual strategies from a population of modest size are selected at random to compete against each other for a number of trials. The accumulated data from many of these trials can be seen as a noisy, partial window onto the true performance matrix. Non-dominated strategies are preserved in a Pareto front, and novel strategies are generated through sexual reproduction of strategies in the front. In this way we hoped that our population would come to approximate the true Pareto-optimal set, and would provide robust general strategies.

1.2 HOLD'EM POKER AS A TEST CASE

To provide a convincing test of the hypothesis that Pareto coevolution can be used to find robust strategies, we wanted to avoid toy problems in favour of a real game. We have chosen poker, a card game of some depth in which a wide range of strategies and skill levels are exhibited by human players.

The specific poker variant we used was limit Texas Hold'em, one of the most popular versions of poker in modern casinos. The popularity of Hold'em must be partly due to the balance between public and private information in the game, which leaves a lot of room for convincing bluffs. A game of Texas Hold'em typically involves eight to ten players, and each complete hand has the following four-round structure.

The pre-flop: each player is dealt two cards face down. These are hole cards, or private cards. The player to the dealer's left makes a forced bet called the small blind, equal to one chip in our case. The next player must bet the big blind, which is equal to two chips. The third and subsequent players must then call (match the bet), raise (increase the bet) or fold (throw in their cards and forfeit all interest in the pot). As this is a limit game, any raises must be exactly two chips at this stage. In addition, no more than three raises are allowed in this or any other round of betting, unless there are only two players left, in which case raising can continue until someone runs out of chips.

The flop: when the previous round of betting is complete (all players have either called or folded), three cards are "flopped" face up in the middle of the table. These are community cards, and are available to all players. By mentally combining the community cards with their hole cards, players can now form a 5-card poker hand, such as two pair, or a flush. There is another round of betting, again starting with the player to the dealer's left. Players can check (decline to bet if no-one else has bet), call, raise by two chips only, or fold.

The turn: a fourth community card is turned face up, and there is another round of betting. Note that even though six cards are now available, players can only make five-card poker hands. The stakes increase now, and all raises must be four chips.

The river: a fifth community card is dealt face up, and there is a final round of betting, with four-chip raises. When the round of betting is complete, all players who still have an interest in the pot compare their hands, and the player with the strongest hand[1] takes the pot.

The art of the game consists of such points as knowing when your cards are likely to be strongest, knowing whether it's worth staying in the pot to improve your hand with subsequent community cards, reading the likely strength of your opponents' hands through their patterns of betting, and of course effective bluffing (see Sklansky, 1999, for a more authoritative discussion).

[1]Poker hands, from weakest to strongest, are: high card, a pair, two pair, three of a kind, a straight, a flush, a full house, four of a kind, and a straight flush.

2 METHODS

2.1 REPRESENTING POKER STRATEGIES

Our primary goal was to test the effectiveness of our Pareto coevolution algorithm, not to evolve world-class poker strategies. We have therefore used an economical representation scheme that is not able to capture many of the subtleties of expert-level poker. In deciding whether to fold, call, or raise, our strategies attend to the strength of their hand at each point in the game. They do not pay any attention to the behaviour of other players except insofar as they are aware of what the current bet is, and may choose to fold because the stakes have become too high for them.

The strategy representation begins with two probability values (real numbers between zero and one inclusive). The first gives the probability with which a player will bluff (i.e., pretend to have very strong cards) on any given hand. The second gives the probability with which a player will check-raise when given the opportunity—this is a deceptive play in which a player bets nothing, indicating weakness, and then raises when the bet comes around again.

Next there are 24 integers in groups of six, describing strategy for each of the four betting rounds (see section 1.2). Two integers describe the minimum cards that a player wants at this stage in order to remain in the hand, e.g., a pair of aces, three sevens, or a king-high flush. Another two integers describe the cards that a player would regard as a strong hand. One integer describes the amount that a player would prefer to bet at this stage, and a final integer gives the maximum amount a player will bet. If players have less than their minimum requirements, they will check if possible or fold if asked to bet. If players have equalled or exceeded their minimum requirements, they will raise until the betting reaches their preferred level. If betting goes higher than their preferred level, they will call until their maximum bet is exceeded, and then they will fold. But if their cards qualify as strong, they will call any bet.

Finally, four groups of four binary values modify the player's behaviour on each betting round. One bit indicates whether or not the player will ignore their normal preferred and maximum bets, and instead bet as much as they possibly can, if their cards qualify as strong. A second bit determines whether or not the player is willing to stay in the hand if their cards are no better than what is showing on the community cards (for example, if the player holds ace-king, and the flop is three queens, then the player's hand is three queens, but that hand is available to all the other players too). A third and a fourth bit indicate a willingness to stay in the hand if one card short of a straight or a flush respectively. (Note that the second bit does not apply to the pre-flop round, and the third and fourth bits do not apply to the pre-flop round or the river round.)

Some of the features that a more sophisticated strategy representation might cover include: whether or not the two pre-flop cards are the same suit (for possible flushes) or close in value (for possible straights), the player's position in the betting order, whether a player has paired the top, middle or bottom pair on the flop, the relative size of the player's stack of chips, whether the size of the pot justifies a risky bet, and how often other players are seen to fold early or to bluff. Nevertheless, as is apparent to us from playing against various evolved and hand-coded strategies,[2] the current strategy representation is adequate to produce poker strategies ranging from the very bad to the reasonably good.

2.2 A SIMPLE PARETO COEVOLUTION ALGORITHM

We began with a population of 100 random poker strategies. Ten strategies were selected at random to make up a table, and a game of 50 hands of poker was played out. Two hundred such games were played per generation, which meant that each strategy was assessed over an average of 1000 hands, and had a chance to play against most of the other strategies in the population.

Results from each of the 10,000 hands of poker played in a generation were collated in a matrix showing who had won or lost chips to whom. Pairwise comparisons were conducted on this matrix in order to identify Pareto-dominated strategies. Non-dominated strategies were maintained in a Pareto front, and the remaining slots in the population were filled through sexual reproduction of randomly chosen members of the Pareto front. Reproduction included multi-point crossover and mutation as in a standard genetic algorithm (GA).[3] After the population had been restocked, the win-lose matrix was wiped clean, and the cycle began again.

One problem that became apparent in trial runs was that the entire population, or very close to it, would often be included in the Pareto front. This was presumably due in part to noise in our evaluation process—even over 1000 hands, the luck of the deal had a significant influence on success, making the true worth of a strategy hard to discern. Furthermore, each strategy could expect only about 100 hands against each opponent, and sometimes did not get to play against a specific opponent at all.

In order to keep exploring new regions in strategy space, we needed to limit the size of the Pareto front. We set the maximum size of the front at 50 strategies, which meant that up to half the population was preserving accumulated wisdom, while the other half was exploring new possibilities. But in the event that more than 50 strategies were non-dominated at the end of a generation's 10,000 hands, we needed a principled way of deciding which strategies would be maintained in the front and which would be discarded. In devising a metric for this purpose, we wanted to stay as close as possible to the Pareto selection ideal, i.e., that one should not assume that the dimensions of success are equally weighted. Strictly speaking, the method we devised does violate this—and we suspect that any method for keeping less than the full Pareto front must—but it does not use an average or sum of scores across different dimensions. Instead we have used a count on the number of dimensions in which a player excels.

Our method was to eliminate those strategies that were "nearly dominated," until our front size was less than or equal to 50. A strategy is nearly dominated if the number of opponents that it is superior to, with respect to its best competitor, is low. The best competitor is defined as the strategy that minimizes this number of opponents. To elaborate: in determining whether a strategy A is Pareto-dominated by B, or vice versa, we look at the scores of A and B against all other strategies. We count the number of strategies, or dimensions, for which A scores higher than B. If this count is zero, then A is dominated by B, and will not be a member of the Pareto front in any event—there is

[2] Code (in C) for playing poker against evolved and hand-coded strategies is available on the web at http://www.comp.leeds.ac.uk/jasonn/Research/Pareto/ . Code for running our Pareto selection algorithm is also available.

[3] There were 37 genetic loci, the crossover rate was 0.1 per locus, and the mutation rate was 0.02 per locus. For the genetic parameters that were real or integer values, mutation was implemented as a small gaussian perturbation, with a mean of zero and a standard deviation of 0.05, 1, or 2 for probabilities, hand rankings, and betting amounts respectively (see section 2.1 for details). Ten percent of mutations were denoted as catastrophic and resulted in a new random value for that parameter.

nothing that A can do that B cannot do better. If this count is greater than zero, then A is not dominated by B. If we look at these counts for A compared with all other strategies, the minimum count gives an indication of how close A came to being dominated. In order to limit the size of the front, we throw out strategies for which this count was equal to one, then two, then three, etc., until the membership of the Pareto front is less than or equal to 50.

In summary, our implementation of Pareto coevolution involved selection based on non-dominance, given the noisy, partial window onto the true payoff matrix that is obtained from the results of a generation's 10,000 poker hands. We also developed a heuristic for limiting the size of the Pareto front. However, there were potential problems with our procedure. Although a strategy that is dominated with respect to the current population must also be dominated with respect to all possible strategies, the converse is not true (Schaffer, 1985). So one strategy might remain in our Pareto front despite being dominated by another, as yet unseen (or already discarded). Noise in the evaluation process, combined with our elimination heuristic, might prevent non-dominated strategies from being recognized as such in the first place. Another possible complication is perhaps specific to the game of poker: success is measured in the context of the other players at a table, but this is not explicitly controlled for. Strategy A might tend to do very well against strategy B when matched directly, but not at a table where C and D were present.

2.3 MEASURING EFFECTIVENESS OF THE ALGORITHM

In order to determine the effectiveness of our Pareto coevolution procedure, we compared its performance with that of a regular coevolutionary GA. This merely provides a baseline performance measure to give us an indication of whether Pareto coevolution can improve performance and robustness of evolved strategies, compared to regular coevolution where fitness is based on an average score over opponents in the population.

The parameters for the GA (i.e., population size, number of hands played per generation, mutation rate, crossover rate, etc.) were the same as those used for the Pareto coevolution algorithm. Strategies were selected for reproduction based on their profit or loss after 10,000 hands: specifically, the scores were normalized with the minimum set equal to zero, and roulette-wheel selection applied to the normalized scores.

Both algorithms were run 20 times for 100 generations each time. We can view the comparison of the two algorithms as a test of which one can produce the best strategies given a million hands (100 generations $\times$ 10,000 hands) worth of information.

In deciding which of the two algorithms had produced better results, we were faced with a somewhat paradoxical problem of measurement. Precisely because the fitness of a strategy cannot be given in isolation, but can only be measured with respect to a particular opponent or set of opponents, it is difficult for us to provide a single, general measure of the strength of the evolved strategies. The familiar Red Queen effect means that it will not help to look at performance against the other strategies in the population, as the zero-sum nature of poker ensures that mean fitness will always be zero.

We decided to construct two sets of five hand-coded reference strategies for the purposes of comparison, using the same representational scheme as the evolving populations (see section 2.1). These reference strategies are not claimed to be in any way optimal; they merely represent some typical, more-or-less reasonable playing styles. For example, we constructed several conservative strategies, that would not bet unless they had quite strong cards. Some of the strategies were deceptive, either because of frequent bluffing, or through "slowplaying," i.e., hiding the strength of one's cards until late in the hand. Other strategies tended to call all bets as long as they held a reasonable hand.

Assessment of the strategies evolved under our two different selection regimes was carried out by having each strategy in the population play alone against a table stocked with reference strategies, for a fixed sequence of 1000 hands. The overall profit or loss of each evolved strategy was recorded. The same random seed was used to deal out the same sequence of cards in every assessment run, in an attempt to reduce some of the noise inherent in the process. The reference strategies were divided into an alpha and a beta group, and assessment was carried out against each of these groups. Note that the ten reference strategies were simply sorted at random into the two assessment groups; there was no intention that the alpha group should be superior to the beta group, for instance. We wanted to be sure that we had not accidentally constructed an unusual or eccentric reference point, and comparison of results against two distinct groups gave us some some insurance against this possibility.

It is important to be clear about what good performance against these two reference groups might mean. Strategies under both selection regimes never encountered any of the reference strategies during the course of evolution. Strategies were selected solely for their

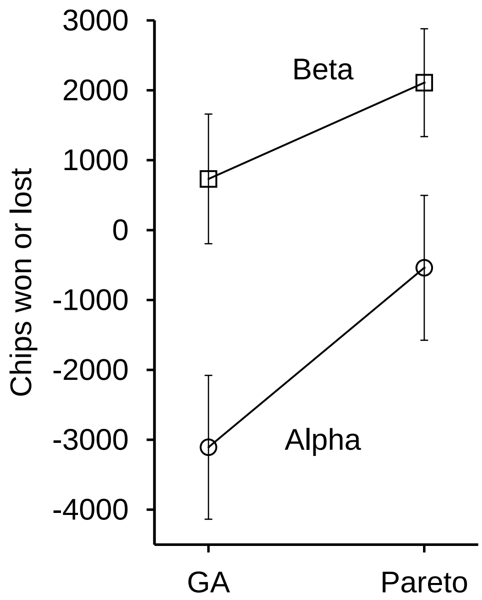

Figure 2: Mean performance (±1 standard error) of evolved strategies in 1000 hands of play against the alpha and beta reference groups; strategies evolved with a coevolutionary GA compared with strategies evolved under Pareto coevolution. Results summarised across 20 runs in each case.

ability to do well against other members of their population, either in the Pareto sense or in the conventional sense of having a high average score. If they managed to do well against an arbitrary set of hand-coded strategies, that gives us some indication that they would do well against a wide range of strategies, i.e., that they are robust and have not adapted to their conspecifics in an overly brittle manner.

3 RESULTS

Figure 2 shows that after 100 generations of evolution, strategies evolved under Pareto selection had a higher mean performance against both of the reference groups than did the strategies evolved using a conventional GA. As the standard error bars indicate, this difference is more pronounced in performance against the alpha group.

Figure 2 also indicates that the alpha reference group was significantly harder to beat than the beta group—both strategies lose to the former and win from the latter on average. This difference was not intended, but the fact that there is no evidence of a strong interaction between selection regime and reference group performance (i.e., the two lines in Figure 2 are roughly parallel) is a reassuring indicator that the two reference groups are measuring something like general ability.

If we look in detail at the evolved strategies across the two selection regimes, the most striking difference is that the Pareto strategies bluffed less often on average (20% vs. 36%). This fact alone explains a lot of the difference in success between the two conditions: in those populations where a high level of bluffing obtained, performance against the reference strategies was always very poor. This is because the only type of bluffing available to these strategies was a simple-minded approach in which they pretended they had a royal flush right from the beginning of the hand and never gave up their bluff no matter how determined the opposition. The Pareto selection process seems to have made it easier for the population to discover the folly of this sort of bluffing.

There were other differences: the Pareto strategies had lower standards for staying in at the preflop and at the river. They tended to bet more, and were more likely to bet as much as possible if they had strong cards (except on the final round of betting). They were less likely to stay in the hand if they weren't beating the community cards, and were more likely to wait for straights and flushes if they were one card short. Readers who play poker may be interested in seeing a complete strategy description. The following is a high-performing evolved strategy from Pareto run 17, in which the average wins were 2009 and 5267 chips against the alpha and beta groups respectively.

- Never bluff, and check-raise 11% of the time.
- At the pre-flop stage, bet as much as possible if you have an ace or a pair—otherwise fold.
- On the flop, stay in as long as you are beating the community cards. If you are one short of a straight or a flush, stay in in any event. Try to bet just two chips, but call bets up to 42 chips. If you have a straight or better, call any bet.
- On the turn, keep waiting for a straight or a flush, but otherwise fold if you have less than a pair of sixes or if you are not beating the community cards. If you stay in, try to bet 6 but call bets up to 59 chips. If you have two pair, with the top pair sixes or better, bet as much as you can.
- On the river, if you have a pair of aces or better, then bet as much as possible. Otherwise fold, and definitely fold if your two aces are community cards.

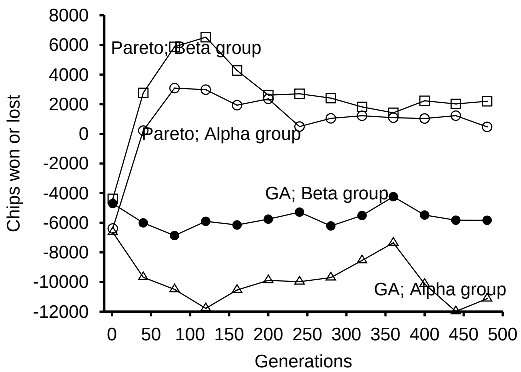

Figure 3: Mean performance of evolving strategies in 1000 hands of play against the alpha and beta reference groups, for both Pareto and standard coevolution, over 500 generations. Data taken from runs with a common random seed value of zero.

Some peculiarities were noted regarding the Pareto selection condition. The Pareto front, of maximum size 50, was always full, which means that some non-dominated strategies were being eliminated in every generation. The mean age of strategies in the Pareto front was approximately two generations, and the median age was always one generation. This suggests a front made up of mostly very young strategies with a few older ones, which is not unexpected, but the mean age of only two generations indicates an extremely rapid turnover of strategies. When Pareto populations were examined at the end of a run, they were not as diverse as we would have hoped. Again, this was not a complete surprise, as reproduction with crossover was employed, but it indicates that the Pareto front has not been completely successful in preserving a range of very different strategies that are non-dominated with respect to each other.

We looked briefly at what happened when evolution continued for more than 100 generations, and found that in many cases performance against the reference groups actually worsened. Figure 3 gives an example of this, with mean performance data over time for an extended version of run zero, showing both Pareto and standard coevolution against the two reference groups. The Pareto-evolved players are declining in performance and moving closer to zero profit, while the GA strategies are making significant losses but with no clear trend up or down.

4 CONCLUSIONS

Our Pareto coevolution algorithm was superior to a GA at finding robust Texas Hold'em strategies within 100 generations. This fact should not be over-interpreted: clearly, we worked with only one game, two small groups of arbitrary reference strategies, and a particular set of parameter values. Nevertheless, our finding does show that Pareto coevolution of strategies in games can work in principle and is an idea worth exploring.

The algorithms we have presented for selecting non-dominated strategies and for discarding excess strategies from the Pareto front could probably be improved upon so as to use the multi-dimensional information from the games played more efficiently and effectively. Our current method maintains only a rough approximation to the Pareto front as compared to existing multi-objective optimization methods, because of the unusually high number of objectives we are using. However, in regular coevolution the multidimensional information is discarded completely, in favor of a single "performance on average" dimension. To put it another way, fitness evaluation and selection are noisy, incomplete processes under both selection regimes—noisy because of the stochastic element, and incomplete in the sense that we cannot observe performance against all possible opponents. But in Pareto coevolution, we are trying to use the information gained from 10,000 hands of poker more intelligently: instead of simply taking an average, we use the specifics of who beat who, and we remove the unwarranted assumption that every other strategy is equally worth beating.

The long term behaviour shown in Figure 3 is somewhat disturbing. It seems that our Pareto-selected strategies cannot hold onto their collective wisdom over time (although the same effect was observed with the more successful GA-evolved strategies). This effect may be due to the population chasing its own tail into eccentric regions of the strategy space; if this is the case, then we need to refine our coevolution algorithm. But note that we are not selecting for maximization of scores against the reference strategies—we are selecting for not being dominated by anyone else in the population. It is an open question as to whether the long term reduction in success apparent in Figure 3 is a sign of "population senility." It may represent a movement towards careful compromise strategies that do not make spectacular wins, but instead make modest profits against a wide range of opponents, and are careful not to lose to anyone.

This paper is the preliminary exploration of an idea, and so we have many questions for future work. One of the most pressing is about the explore-exploit balance in our algorithm: is 50% of the population a reasonable size for the ongoing Pareto front? Would we benefit from having a "genetic freezer" for storing past champion strategies, and then re-inserting them into the front at regular intervals? How big is the true Pareto-optimal set likely to be in a game like poker, and what chance do we have of getting a reasonable approximation to it with our method?

We also want to look at reproduction of Pareto-selected strategies. In the current paper we have used standard sexual reproduction, partly to facilitate comparison with the GA. It seems worth exploring asexual reproduction, or at least much lower levels of crossover, to see if we can avoid the unfortunate degree of convergence reported in section 3. It would be interesting to see whether asexual reproduction also resulted in an increase in the mean age of in the Pareto front.

Once we have refined our Pareto coevolution algorithm, it would be sensible to test it against more than just a standard GA. If we view the problem as how to learn the most you can from one million hands of poker, then we should ultimately be testing Pareto coevolution against a range of established evolutionary computation and machine learning techniques. In the meantime, our experiments have provided a simple illustration of Pareto coevolution, and begun to explore some of the issues involved.

Acknowledgments

The authors are grateful to Sevan Ficici for invaluable discussion and insight into the principles of Pareto selection as applied to games, and to Anthony Bucci for assistance in developing the heuristic to reduce the Pareto front. We would also like to thank four anonymous reviewers for their comments.

References

Cliff, D., & Miller, G. F. (1995). Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations. In Morán, F., Moreno, A., Merelo, J. J., & Chacón, P. (Eds.), *Advances in Artificial Life: Third European Conference on Artificial Life (ECAL '95)*, Vol. 929 of *Lecture Notes in Artificial Intelligence*, pp. 200–218. Springer, Berlin.

Ficici, S. G., & Pollack, J. B. (2000). A game-theoretic approach to the simple coevolutionary algorithm. In Schönauer, M., et al. (Eds.), *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pp. 467–476. Springer, Berlin.

Ficici, S. G., & Pollack, J. B. (2001). Pareto optimality in coevolutionary learning. Computer science technical report CS-01-216, Brandeis University.

Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, *3*(1), 1–16.

Horn, J. (1997). Multicriteria decision making and evolutionary computation. In *Handbook of Evolutionary Computation*. Institute of Physics Publishing, London.

Pollack, J. B., & Blair, A. D. (1998). Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, *32*(3), 225–240.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pp. 93–100. Lawrence Erlbaum.

Sklansky, D. (1999). *The Theory of Poker* (Fourth edition). Two Plus Two Publishing, Henderson, NV.

von Neumann, J., & Morgenstern, O. (1953). *Theory of Games and Economic Behavior*. Princeton University Press.

Watson, R. A., & Pollack, J. B. (2000). Symbiotic combination as an alternative to sexual recombination in genetic algorithms. In Schönauer, M., et al. (Eds.), *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pp. 425–434. Springer, Berlin.