# A Comparison of Two Competitive Fitness Functions

**Liviu Panait**
George Mason University
http://www.cs.gmu.edu/∼lpanait/

**Sean Luke**
George Mason University
http://www.cs.gmu.edu/∼sean/

## Abstract

Competitive fitness is the assessment of an individual's fitness in the context of competition with other individuals in the evolutionary system. This commonly takes one of two forms: one-population competitive fitness, where competition is solely between individuals in the same population; and N-population competitive fitness, often termed competitive coevolution. In this paper we discuss common topologies for one-population competitive fitness functions, then test the performance of two such topologies, Single-Elimination Tournament and K-Random Opponents, on four problem domains. We show that neither of the extremes of K-Random Opponents (Round Robin and Random-Pairing) gives the best results when using limited computational resources. We also show that while Single-Elimination Tournament usually outperforms variations of K-Random Opponents in noise-free problems, it can suffer from premature convergence in noisy domains.

## 1 INTRODUCTION

Traditional evolutionary computation assesses the fitness of an individual independently of other individuals in the system. But there also exist evolutionary procedures where this is not the case: an individual's fitness is dependent on cooperation or competition with peers in the evolutionary run, and thus may change depending on the makeup of those peers.

Such procedures have several attractive features. First, they permit evolution to search for solutions to problems in the absence of any obvious way to gauge an objective (peer-independent) fitness. Consider: how does one determine the quality of a soccer player program *a priori?* Second, they can gradually ramp up problem difficulty as evolution finds better solutions. This promises to smooth out the search gradient. Third, they seem a natural match for finding solutions to problems that naturally require teamwork or that are most easily discovered through competition.

We are tempted to bring all these procedures under the aegis of *coevolution*, but there are nomenclature difficulties with the use of this term. In biology, coevolution is best reserved for situations where there is more than one population, and an individual's fitness is assessed in the context of individuals in other populations. Such multi-population coevolution is usually used as a self-adaptive mechanism to increase problem difficulty as members of the population become more adapt at solving the given problem. The classic example of multi-population competitive coevolution is [Hillis 1991], which coevolved a population of sorting networks and a population of problem sets. The fitness of sorting networks was based on the number of problem sets they properly solved, and the fitness of the problem sets was based on the number of sorting networks they stumped. [Rosin and Belew 1995] also used a two-population competitive system to evolve players for the games of Nim, Tic-Tac-Toe, and Go with a 7x7 board. Multi-population coevolution is also useful as *cooperative coevolution*. Here individuals from different populations each learn subparts of a common solution, and their fitness is based on the combination of those subparts. Examples of cooperative coevolution include [Eriksson and Olsson 1997; Potter and De Jong 2000; Wiegand et al. 2001].

One-population "coevolution" rarely if ever takes cooperative form. Instead, this technique is nearly universally used to evolve game players by competing amongst themselves. For lack of a standardized term for one-population techniques, we call these *one-population competitive fitness functions*; for the rest of this paper, whenever we say "competitive fitness functions" we imply the one-population sort. [Luke 1998] used such competitive fitness to evolve soccer-playing softbot teams, and [Fogel 2001] used the technique to evolve a highly human-competitive checkers program, Blondie24. One-population competitive fitness has also been used to find solutions to the Iterated Prisoner's Dilemma [Axelrod 1987], Tic-Tac-Toe [Angeline and Pollack 1993], Backgammon [Pollack et al. 1997; Pollack and Blair 1998], Othello [Smith and Gray 1993], pursuit-evasion [Cliff and Miller 1995], Go [Lubberts and Miikkulainen 2001] and Tag [Reynolds 1994].

One important part of a competitive system's success is its *topology:* how the fitness-evaluation context is established for a given individual. Do all individuals play against all other individuals in the population? Are they simply paired up for a single game each? Some topologies require a large number of games to evaluate an individual, but may be more accurate than those requiring fewer games.

This paper compares two topological families in one-population competitive-fitness games. We begin by discussing common topologies in the literature and their advantages and disadvantages. Then we introduce four problem domains, and show how various topologies fare in these domains and under different amounts of noise in the fitness-assessment process.

## 2 COMPETITION TOPOLOGIES

Not all competitive fitness topologies are appropriate for all problems; the primary issue breaks down along lines of fitness-assessment methodology. Imagine if one were trying to evolve chess players. How does one establish that player A is better than player B? The *duel methodology* states that A is better than B if and only if A usually beats B in a match. This is the methodology behind single- and double-elimination tournaments. The *rennaisance-man methodology* says that A is better than B if A beats more competitors than B does on average (or scores against competitors by a wider margin on average), even if A would lose to B in a match. This is the methodology behind chess rankings, for example. It is interesting to note that many sports use a combination of these two methodologies, usually by using average success against opponents during the season to determine the entrants to a single elimination tournament, which then determines the final champion. Whether there is some innate superiority to this combination is questionable: more likely it is due to the excitement of duels: after all, “in the end there can be only one”.

There are other interesting issues in designing topologies which we will not delve into save to mention them here. One issue is whether or not individuals should play against themselves as part of their evaluation. Another is whether or not to permit statistical dependencies in fitness assessment: when individual A plays against individual B, should the outcome affect individual A's fitness alone, or should it also affect individual B's fitness?

### 2.1 ROUND ROBIN

One simple topology is *Round Robin*, where each individual plays every other individual in the population. An individual's fitness is the average of its scores against every other individual in the population [Axelrod 1987; Koza 1992]. The primary drawback to this method is the relatively large number of games necessary to evaluate a population of size $N$. The number of games is $(N^2 - aN)/b$, where $a = 0$ if individuals may play against themselves, else $a = 1$, and $b = 2$ if a game contributes to the fitness of both individuals, else $b = 1$. At first glance it would appear that Round Robin topologies would promote the rennaisance-man methodology. At the beginning of an evolutionary run, this is plausible. But as the run progresses, the trajectory of the run might shift to the “better” players, so to speak, so that near the end of the run it is searching not for individuals who win the most points on average, but oddly for individuals who win the most points on average against *other such individuals*.

### 2.2 RANDOM-PAIRING

The other extreme in the number of games is to pair all individuals up and play one game for each pair. This is the approach used in [Luke 1998] for evolving soccer team strategies. The justification for this low number of games was the extreme computational cost of a game: to be evaluated, the two teams were plugged in a simulator, and a standard game could last for up to 10 minutes. Random-Pairing requires only $N/2$ games for a population of size $N$. The cost savings is dramatic: for a population of 100 and ten minutes per game, Axelrod's Round Robin approach would require 833 hours per generation, whereas Random-Pairing would require about 8 hours. Smith and Gray [1993] also used this technique to evolve Othello players. The danger of Random-Pairing is that noisy evaluation might make it all but impossible to determine the real quality of an individual based on a single trial. Note too that like Round Robin, Random-Pairing has a similar tenuous claim to promoting the rennaisance-man methodology.

### 2.3 SINGLE-ELIMINATION TOURNAMENT

[Angeline and Pollack 1993] proposed using single-elimination tournaments (“SET”) rather than Round Robin or Random-Pairing. Here, individuals are paired at random, and play one game per pair. The losers of the games are eliminated from the tournament; ties are broken by random decision. The winners are again paired off at random, and play one game per pair, with the losers again eliminated. This continues until the tournament has only one “champion” left. The fitness of an individual is the number of games it played. Single-Elimination Tournament is simplest to implement when the population is a power of two. Angeline and Pollack reported good initial results when using SET to evolve players for the game of Tic-Tac-Toe.

SET has interesting properties. First, it would seem to promote the duel methodology rather than the rennaisance-man methodology. However, it only truly promotes the duel methodology under the *strong transitivity assumption:* that if player A beats player B, and player B beats player C, then player A must beat player C. Without this assumption, Single-Elimination Tournament's real dynamics can be murky. The other interesting property of SET is that it seems to allocate games to those players that most need them. A population of size $N$ needs only $N-1$ games. But "fitter" players will be evaluated in more of these games than the "less fit" players — the worst individuals play only one game each, while the champion plays $\ln(N)$ games. Since selection will tend to pick the fitter players, SET would seem to proportion more games, hence more accuracy, among those players more likely to be selected.

### 2.4 K-RANDOM OPPONENTS

In K-Random Opponents, each individual plays against $K$ individuals picked at random from the population. If a given game between two individuals affects the fitness of just the first individual, then a total of $K(N-1)$ games must be played. This is the approach taken in evolving tag players [Reynolds 1994]. K-Random Opponents can also be used to affect the fitness of both individuals in a game. For example, to evolve the Blondie24 checkers player, Fogel [2001] had every individual play as red against five opponents chosen at random with replacement from the population. An individual's fitness was based not only on its five games as red, but also as its additional games as a black opponent.

This approach does not distribute games very evenly throughout the population, however. With some forethought, it's possible to adapt K-Random Opponents so that a given game affects both individuals, with each individual using the same number of games per evaluation. The technique, which we will use in experiments below, works as follows. Each individual maintains a count of the number of games it has played, and who it has played against. When an individual $I$ is to be evaluated, an opponent is chosen at random from the population to play against $I$ with the constraint that no individual may play against $I$ more than once. At the end of the game, the number-of-games counters for $I$ and for the opponent are incremented. If either counter reaches $K$, then that individual is "removed" from the population in the sense that it may no longer be considered as a future opponent. A new opponent for $I$ is chosen, and this process continues until individual $I$ has been removed. Then a new player $J$ is picked, and evaluation continues similarly. At some point, for some individual $K$, there may exist no individuals in the population which can play $K$. When this occurs, opponents for $K$ are picked at random, without replacement, from among the removed individuals in the population. This approach yields between $\lceil (KN)/2 \rceil$ and $\lceil (KN)/2 \rceil + \lfloor K^2/2 \rfloor$ games.

Round Robin and Random-Pairing may be viewed as extremes of $K$ for this second kind of K-Random Opponents. When $K = N-1$, K-Random Opponents is identical to Round Robin. When $K = 1$, K-Random Opponents is identical to Random-Pairing. Later in the paper, we will examine K-Random Opponents to determine what value of $K$ seems to give the best results: as it will turn out, it is neither of these extremes.

### 2.5 HALL OF FAME

One last approach in the literature is a family of "hall of fame" techniques, where individuals in the population are evaluated against the good individuals discovered so far in the evolutionary run. Karl Sims used a simple hall of fame when evolving creatures which competed to snatch a cube [Sims 1994]. Individuals were evaluated against the fittest individual discovered in the previous generation.

## 3 PROBLEM DOMAINS

The problem domains we will test against fall into two categories. First, we use two true competitive fitness domains, namely versions of the Nim game. Second, we have adapted two standard evolutionary algorithm problems and cast them into a competitive fitness form. They are the well-studied Rosenbrock and Rastrigin problem sets. These algorithms are cast into competitive form using a technique proposed by Ken De Jong: each individual's Rosenbrock (or Rastrigin) value is assessed, and an individual's score in a game against an opponent is based on difference in their values.

### 3.1 THE INTERNAL ROSENBROCK DOMAIN

The Rosenbrock function is a well-known minimization problem widely used to study properties of different evolutionary algorithms [De Jong 1975]. The Rosenbrock function for genomes of $n$ variables is:

$$Ros(x_1,\ldots,x_n) = \sum_{i=1}^{n} 100\left({x_i}^2 - x_{i+1}\right)^2 + \left(1 - {x_i}^2\right)$$

Rosenbrock is converted to the “Internal Rosenbrock” competitive fitness function as follows. When a player $A$ plays an opponent $B$, the score for $A$, known as $Reward(A:B)$, is given by the following normalizing formula:

$$Reward(A:B) = \frac{Ros(B) - Ros(A)}{\max(Ros) - \min(Ros)}$$

...where $\max(Ros)$ and $\min(Ros)$ are the maximum and minimum values of the Rosenbrock function over the entire domain, which we had precomputed. Thus $Reward(A:B)$ ranges from -1 to 1, where 0 represents a draw. Note that this is a zero-sum, transitive game, hence $Reward(B:A) = -Reward(A:B)$. Keep in mind that Rosenbrock is a minimization function: therefore the smaller $Ros(A)$ is compared to $Ros(B)$, the higher the reward for $A$.

**Parameters** Internal Rosenbrock experiments used a genome of 100 real values each between -5.12 and 5.12, a population size of 32, a 0.5 probability of mutation, 1-point crossover with a probability of 1.0, 5-individual elitism, binary tournament selection, and a maximal run limit of 50,000 games.

### 3.2 THE INTERNAL RASTRIGIN DOMAIN

The Rastrigin function is another well-known test in function optimization; it is considered difficult to minimize because it has a single global optima with numerous local optima in its vicinity [Cervone et al. 2000]. The Rastrigin function is defined as

$$Rastrigin(x_1 \ldots x_n) = \sum_{i=1}^{n} {x_i}^2 + a(1 - \cos(2\pi x_i))$$

...where $a$ is a constant (set to 10.0 in our experiments). Like Rosenbrock, Rastrigin is a minimization problem. Rastrigin is converted to the “Internal Rastrigin” competitive function in exactly the same way as Rosenbrock was converted (though $\max(Ras)$ was estimated).

**Parameters** Internal Rastrigin experiments used a genome of 100 real values each between -5.12 and 5.12, a population size of 32, a 0.5 probability of mutation, 1-point crossover with a probability of 1.0, 5-individual elitism, binary tournament selection, and a maximal run limit of 100,000 games.

### 3.3 THE NIM VERSION 1 DOMAIN

There are many variations on the game of Nim, and we have chosen two different versions as competitive fitness function domains. The Nim Version 1 domain follows the Nim game as described in [Rosin and Belew 1995, 1996]. This version uses 4 heaps containing 3, 4, 5, and 4 stones respectively. Players take turns removing stones from these heaps. A player may remove as many stones as he likes from any single heap. Whichever player takes the last stone wins the game. Given these rules, there exists a well-understood optimal player strategy for the first player.

A genomic representation for a player behavior in this game is a vector of 599 bits, one for each possible situation ($4 \times 5 \times 6 \times 5 - 1$, because the $\langle 3,4,5,4 \rangle$ position does not ever need to be considered). A player makes its decision as follows: for each pile $p$ from 1 to 4, and for each number $x$ of stones for the given pile in decreasing order down to 1, the individual considers whether or not to remove $x$ stones from pile $p$. Removing these stones yields a new game state which corresponds to one of the 599 bits in the genome vector. If this bit value is 1, then the player commits to making that move, and no other consideration is made. If all such valid states have 0 bit values, the player makes the first valid move it had considered.

As the existence of a perfect strategy depends on who goes first, a competition between two individuals consists of 2 games, each player starting one of them. $Reward(A:B)$ is the sum of scores for player $A$ in these two games. For each game, a 0.5 is rewarded for a win and a -0.5 for a loss. The sum of the rewards for the two games is therefore -1, 0, or 1.

**Parameters** Experiments in this domain used a genome of 599 bits, a population size of 128, a 0.003 probability of mutation, 1-point crossover with a probability of 1.0, 10-individual elitism, binary tournament selection, and a run limit of no more than 100,000 games.

### 3.4 THE NIM VERSION 2 DOMAIN

The second version of Nim used in this paper contains a single heap, but the number of stones a player can remove is bounded by a minimum and a maximum value. For these experiments, the heap starts at 200 stones, and each player is allowed to pick 1, 2 or 3 stones at a time. In this configuration, the second player has an optimal strategy which will force a win.

Just as in Nim Version 1, in this game the individuals are represented as vectors, with a similar mapping of bits to the 199 possible states (excepting the initial state). Decision-making is also similar. The player first considers removing 3 stones (assuming that 3 stones are left in the heap). If 1

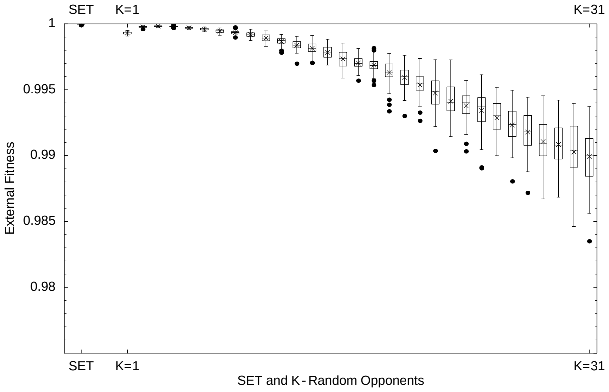

Figure 1: Ranking of SET and K-Random Opponents for Internal Rosenbrock Domain with 0% noise

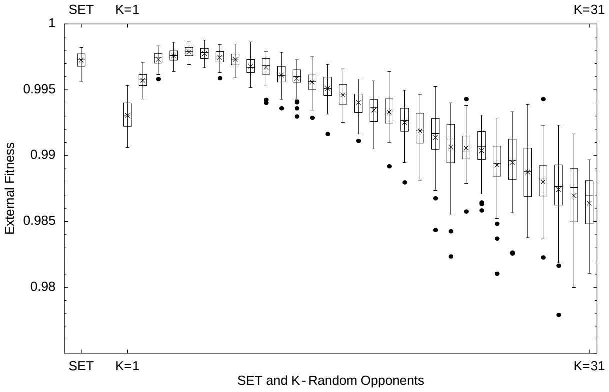

Figure 2: Ranking of SET and K-Random Opponents for Internal Rosenbrock Domain with 30% noise

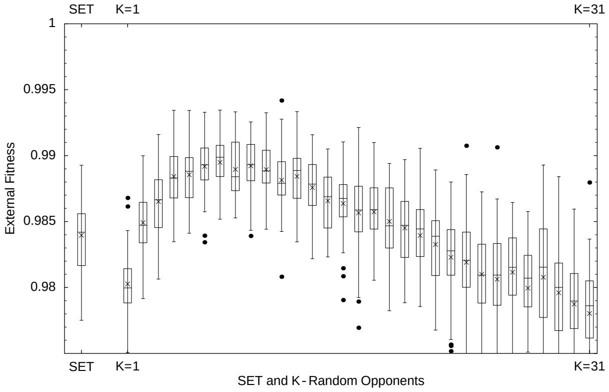

Figure 3: Ranking of SET and K-Random Opponents for Internal Rosenbrock Domain with 40% noise

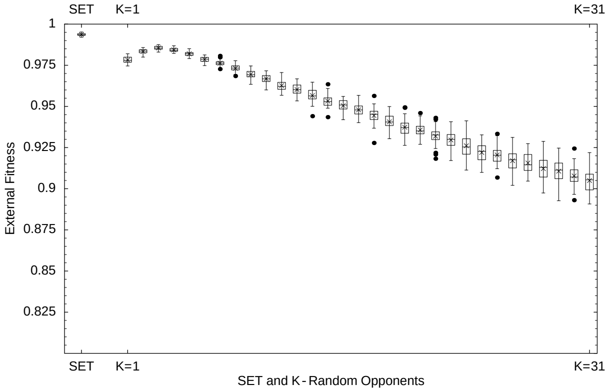

Figure 4: Ranking of SET and K-Random Opponents for Internal Rastrigin Domain with 0% noise

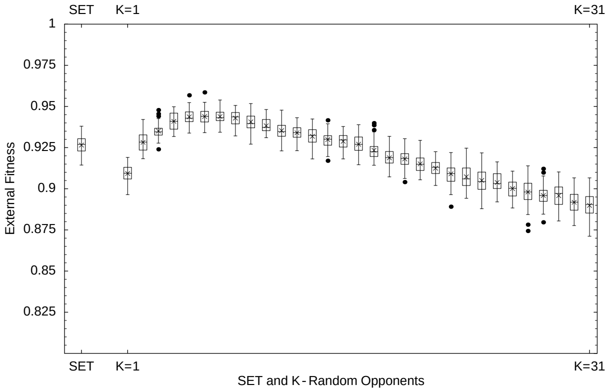

Figure 5: Ranking of SET and K-Random Opponents for Internal Rastrigin Domain with 30% noise

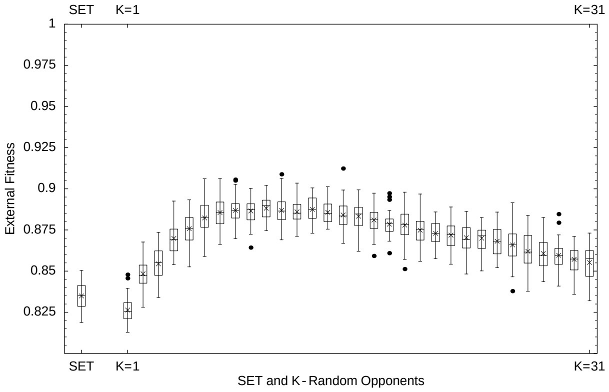

Figure 6: Ranking of SET and K-Random Opponents for Internal Rastrigin Domain with 40% noise

is in the bit position corresponding to the resulting state after removing those 3 stones, then the player will make that move. Otherwise, the player considers removing 2 stones. Barring that, it will consider removing 1 stone. If all three resultant states have 0 in their bit positions, then the player will remove the most stones permissible. A competition between two individuals is done identically to the Nim Version 1 problem.

**Parameters** Experiments in this domain used a genome of 199 boolean values, a population size of 128, a 0.03 probability of mutation, 1-point crossover with a probability of 1.0, 10-individual elitism, binary tournament selection, and a run limit of no more than 100,000 games.

## 4 EXPERIMENTS

The experiments presented here probe the following question. You have 3 months until the deadline to submit an evolved game player to a computer gaming competition. Evaluation is expensive and you'll only get one shot. With a fixed maximum number of games playable until competition-time, what topologies are likely to get good results?

We will compare SET and various K-random opponents topologies over the four problem domains, using a single-population, generational genetic algorithm, with binary tournament selection, mutation, crossover, and elitism. Experimental runs are done by evaluating individuals up to some maximal number of games; the maximal number was previously specified in the parameters for each domain. Keep in mind that an *evaluation* is not the same thing as a *game*. Some topologies require a great many games played before an individual's fitness is determined. Thus each graph compares different topologies' performances given the same number of resources.

Ultimately we are trying to determine what topology is likely to give the "best results". To compare topologies, we need a final *external fitness* used for comparing best-of-run results between topologies, as opposed to the subjective *internal fitness* used to select individuals during the runs themselves. For the Internal Rosenbrock and Internal Rastrigin problems, the external fitness of an individual is clearly objective and clearly computable: it's just the individual's performance on the Rosenbrock or Rastrigin functions.

For the Nim games however, we are faced with the classic external-fitness conundrum: the only obvious external fitness measures available are subjective, that is, they're determined in the context of other individuals. In the absence of any clear objective measure, we must resort to a subjective way to score the final performance of the best-of-run individuals for any given Nim topology. To do this, our approach is to determine the better topologies by literally playing their "best" individuals against each other. For any given application of a topology, we perform 50 independent runs. For each run, we determine a "best of run" by taking the best-of-generation individuals from each generation, and placing them in a single elimination tournament. Thus for each application of a given topology, we have 50 best-of-run individuals. To compare several topologies for a particular problem domain, we then take the 50 best-of-run individuals of each topology and play all of them in a Round Robin tournament. The "quality" of a best-of-run individual in the final tournament is equal to its average score against others in the tournament. Thus the "quality" of a particular topology is the mean of the qualities of its best-of-run individuals. This may not necessarily be an ideal comparison metric (we don't know if an ideal even exists), but we feel it is a reasonable one.

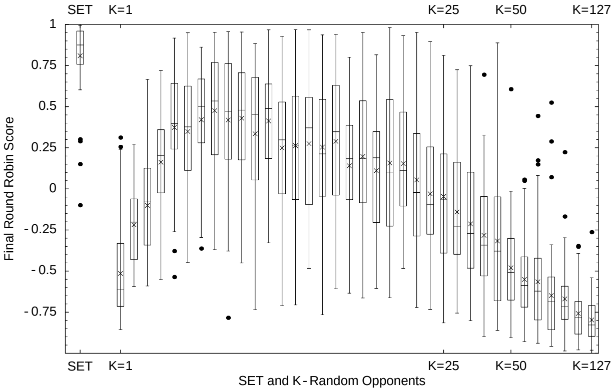

Figure 7: Ranking of SET and K-Random Opponents for the Nim Version 1 game

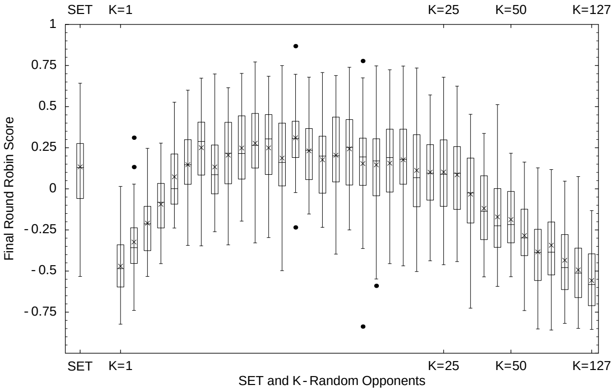

Figure 8: Ranking of SET and K-Random Opponents for the Nim Version 2 game

### 4.1 RESULTS

We ran all experiments on the ECJ 7 evolutionary computation system [Luke 2001]. Figures 1 through 8 show boxplots[1] comparing SET with K-Random Opponents. Figures 1 through 6 use values of *K* ranging from 1 to 31; Figures 7 and 8 use *K* values of 1 to 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 127. The vertical access plots external fitness values of the best-of-run individuals for various topologies. In the Rosenbrock and Rastrigin domains, the external scores were the actual Rosenbrock or Rastrigin function values for the best-of-run individuals. In the Nim domains, the final Round Robin competition to determine external scores consisted of every best-of-run individual plotted in the combined graph.

Figures 2 and 3 show the effects of adding noise to the Rosenbrock domain, and Figures 5 and 6 show similar effects for the Rastrigin domain. Noise was added by flipping a coin with the given noise probability that the players' scores were to be swapped. Noise was *not* used in the display of external fitness results.

**K-Random Opponents Results** We found that the overall layout of the graphs is very similar across all four domains: as the value of *K* increased, external fitness rose, then dropped. The dome-like results for K-Random Opponents suggests that neither Random-Pairing (where $K = 1$) nor Round Robin (where *K* is large) is likely to yield a good result. Indeed, we imagine that Round Robin will often come in dead last! In the Internal Rosenbrock and Internal Rastrigin domains with no noise, Random-Pairing performed reasonably well, but with more noise, it did increasingly poorly.

Why is this happening? Our hypothesis is that in noisy or intransitive domains, only a few games per evaluation is not sufficient to cut through the noise, and evolution proceeds slowly. Then as the number of games per evaluation increase, at some point it becomes overkill: more games are simply cutting the total available evolution time.

This result is similar to the one obtained for non-coevolutionary EAs [Grefenstette and Fitzpatrick 1985] when determining the optimal number of evaluations of an individual in a noisy environment, where the fitness was calculated as the average of the results of several evaluations. Grefenstette and Fitzpatrick too reported that one sample might not provide enough information, while too many samples might not leave enough generations for good results when the total number of evaluations is bounded. They reported that ten samples per evaluation gave the best

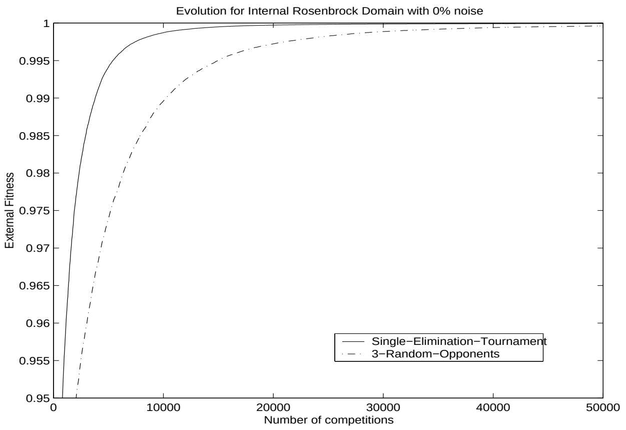

Figure 9: Best-so-far curves for Internal Rosenbrock Domain with 0% noise

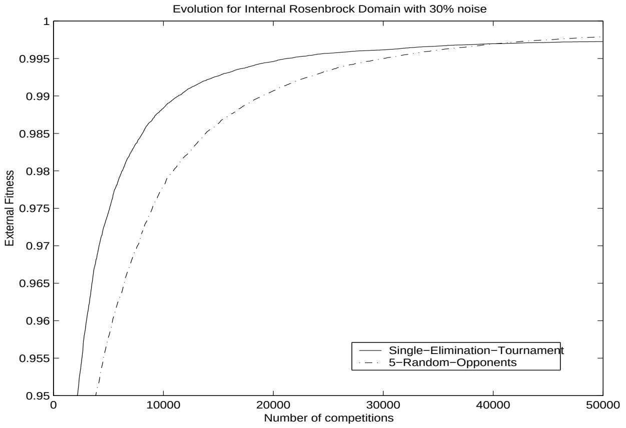

Figure 10: Best-so-far curves for Internal Rosenbrock Domain with 30% noise

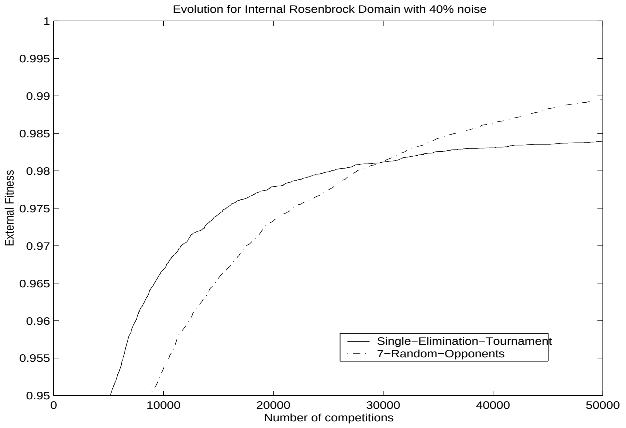

Figure 11: Best-so-far curves for Internal Rosenbrock Domain with 40% noise

---

[1] In a boxplot, the rectangular region covers all values between the first and third quartiles, the stems mark the furthest individual within 1.5 of the quartile ranges, and the center horizontal line indicates the median. Dots show outliers, and × marks the mean.

results in an image registration problem. While we typically found fewer samples were necessary in our coevolutionary approach, ten gave reasonable results in most cases.

**Single-Elimination Tournament Results** The SET results were surprising. When the amount of noise is small, SET performs as good as or better than all other methods presented, even though it has relatively few games per evaluation. As noise is increased to 40% in the Rosenbrock domain, though, SET's performance loses its luster. Why?

Figures 9, 10 and 11 compare the external fitness best-so-far curves of SET and the best performing K-Random Opponents topology, with 0%, 30% and 40% noise respectively. These figures suggest that SET is converging too rapidly: as the field improves, this becomes a hindrance. In Figure 11, ultimately 7-Random Opponents is statistically significantly better (using a t-test at 95%).

It seems that K-Random Opponents *might* be a better choice than SET, particularly if noise is high. The trick, though, is determining what value of $K$ to use. In the absence of any prescience, SET might be the best option.

## 5 CONCLUSIONS AND FUTURE WORK

Our experiments showed that the extremes of the K-Random Opponents method usually lead to worse final results than intermediate (preferably small) values for $K$. Even if games are very expensive, the concern that led to Random-Pairing in [Luke 1998], we still think 5 to 10 games per evaluation is likely to yield a better result. A full Round-Robin tournament appears to be always a bad choice. Our data suggests that the Single-Elimination Tournament may be too aggressive in noisy competitions, leading to premature convergence relative to 5- to 10-Random Opponents. Otherwise it seems to be a good choice.

Though many graphs are similar, nonetheless interesting features stand out. One surprise is the very strong performance of Single-Elimination Tournament in the Nim Version 1 game. This suggests dynamics special to this domain which, on closer investigation, may shed light on SET's performance in general. Does Nim Version 1 promote the duel methodology in a way not found in Nim Version 2, for example? Except for noise, the Internal Rastrigin and Internal Rosenbrock domains are fully transitive: might this explain the deterioration of SET under noise? In future work we hope to examine the dynamics of such topologies in these and other domains more closely.

## Acknowledgments

The authors would like to thank Ken De Jong, Paul Wiegand and Jeff Bassett for helpful comments and suggestions, and Vlad Staicu for his considerable help in running the experiments. We would also like to thank our reviewers for their helpful comments.

## References

Angeline, P. and Pollack, J. (1993). Competitive environments evolve better solutions for complex tasks. In Forest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA)*, pages 264–270, San Mateo, CA. Morgan Kaufmann.

Axelrod (1987). The evolution of strategies in the iterated prisoner's dilemma. In Davis, L., editor, *Genetic Algorithms and Simulated Annealing*. Morgan Kaufmann.

Cervone, G., Michalski, R., Kaufman, K., and Panait, L. (2000). Combining machine learning with evolutionary computation: Recent results on LEM. In *Proceedings of the Fifth International Workshop on Multistrategy Learning*, pages 41–58.

Cliff, D. and Miller, G. F. (1995). Tracking the red queen: Measurements of adaptive progress in co–evolutionary sumulations. In *Proceedings of the Third European Conference on Artificial Life*, pages 200–218. Springer–Verlag.

De Jong, K. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor, MI.

Eriksson, R. and Olsson, B. (1997). Cooperative coevolution in inventory control optimisation. In Smith, G., Steele, N., and Albrecht, R., editors, *Proceedings of the Third International Conference on Artificial Neural Networks and Genetic Algorithms*, University of East Anglia, Norwich, UK. Springer.

Fogel, D. (2001). *Blondie24: Playing at the Edge of Artificial Intelligence*. Morgan Kaufmann.

Grefenstette, J. J. and Fitzpatrick, J. M. (1985). Genetic search with approximate function evaluations. In *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 37–46.

Hillis, D. (1991). Co-evolving parasites improve simulated evolution as an optimization procedure. *Artificial Life II, SFI Studies in the Sciences of Complexity*, 10:313–324.

Koza, J. (1992). *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. MIT Press.

Lubberts, A. and Miikkulainen, R. (2001). Co-evolving a Go-playing neural network. In *Coevolution: Turning Adaptive Algorithms upon Themselves, (Birds-on-a-Feather Workshop, Genetic and Evolutionary Computation Conference)*.

Luke, S. (1998). Genetic programming produced competitive soccer softbot teams for RoboCup97. In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Riolo, R., editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 214–222, University of Wisconsin, Madison, Wisconsin, USA. Morgan Kaufmann.

Luke, S. (2001). ECJ 7: An evolutionary computation research system in Java. Available at http://www.cs.umd.edu/projects/plus/ec/ecj/.

Pollack, J. and Blair, A. (1998). Coevolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240.

Pollack, J., Blair, A., and Land, M. (1997). Coevolution of a backgammon player. In *Artificial Life V*. MIT Press.

Potter, M. and De Jong, K. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29.

Reynolds, C. (1994). Competition, coevolution and the game of tag. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV, Proceedings of the fourth International Workshop on the Synthesis and Simulation of Living Systems.*, pages 59–69. MIT Press.

Rosin, C. and Belew, R. (1995). Methods for competitive co-evolution: Finding opponents worth beating. In Eshelman, L., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA)*, pages 373–380. Morgan Kaufmann.

Rosin, C. and Belew, R. (1996). New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29.

Sims, K. (1994). Evolving 3D morphology and behavior by competition. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV, Proceedings of the fourth International Workshop on the Synthesis and Simulation of Living Systems.*, pages 28–39. MIT Press.

Smith, R. and Gray, B. (1993). Co-adaptive genetic algorithms: An example in othello strategy. Technical Report TCGA 94002, University of Alabama, Department of Engineering Science and Mechanics.

Wiegand, R. P., Liles, W., and De Jong, K. (2001). An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In Spector, L., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2001*, pages 1235–1242. Morgan Kaufmann.