Crossover and Evolutionary Stability in the Prisoner's Dilemma

Xavier Thibert-Plante xavier.thibert-plante@mail.mcgill.ca
Physics Department and Geography Department, Université de Montréal, Montréal, Québec, Canada
Present address: Redpath Museum and Department of Biology, McGill University, Montréal, Québec, Canada

Paul Charbonneau paulchar@astro.umontreal.ca
Physics Department, Université de Montréal, Montréal, Québec, Canada

Abstract
We examine the role played by crossover in a series of genetic algorithm-based evolutionary simulations of the iterated prisoner's dilemma. The simulations are characterized by extended periods of stability, during which evolutionarily meta-stable strategies remain more or less fixed in the population, interrupted by transient, unstable episodes triggered by the appearance of adaptively targeted predators. This leads to a global evolutionary pattern whereby the population shifts from one of a few evolutionarily metastable strategies to another to evade emerging predator strategies. While crossover is not particularly helpful in producing better average scores, it markedly enhances overall evolutionary stability. We show that crossover achieves this by (1) impeding the appearance and spread of targeted predator strategies during stable phases, and (2) greatly reducing the duration of unstable epochs, presumably by efficient recombination of building blocks to rediscover prior metastable strategies. We also speculate that during stable phases, crossover's operation on the persistently heterogeneous gene pool enhances the survival of useful building blocks, thus sustaining long-range temporal correlations in the evolving population. Empirical support for this conjecture is found in the extended tails of probability distribution functions for stable phase lifetimes.

Keywords
Genetic algorithms, crossover, game theory, prisoner's dilemma

1 The iterated prisoner's dilemma

The prisoner's dilemma, a well-known model-problem in game theory, has now become a standard paradigm in studies of the emergence of cooperation among selfish individuals (Axelrod and Hamilton, 1981; Axelrod, 1987; Brembs, 1996). Two non-communicating, competing "players", A and B, must choose to either cooperate (C) or defect (D). They score points according to a payoff matrix (see Table 1) that specifies each player's score according to the four possible combinations of mutual moves. The numerical values for the entries in the payoff matrix have become a kind of standard, following early work in the field (Axelrod and Hamilton, 1981). The "dilemma" arises because the optimal outcome for each player materializes by defecting *provided that* the other player cooperates (DC$\rightarrow$ $[5, 0]$). Mutual defection is bad for both (DD$\rightarrow$ $[1, 1]$) while mutual cooperation yields intermediate scores for both players (CC$\rightarrow$ $[3, 3]$).

Table 1: Payoff matrix for the Iterated Prisoner's Dilemma.

B / A	Cooperate	Defect
Cooperate	3 / 3	5 / 0
Defect	0 / 5	1 / 1

If players meet only once and no prior information is available on the other player's behavior, then the best strategy is to defect. However, mutual cooperation does better under repeated encounters where players "remember" one or more of the opponent's previous moves. This situation defines the iterated prisoner's dilemma game (hereafter IPD). Early research on the IPD showed that a remarkably simple strategy, dubbed Tit-For-Tat (TFT), did quite well under a variety of tournament setups against far more elaborate strategies (see Axelrod and Hamilton (1981)). TFT acts on the basis of the opponent's single previous move. It starts off nice (C on first encounter), retaliates ruthlessly (D if opponent's previous move was D, no matter what anterior moves were), and does not hold a grudge (C if opponent's previous move was C, independently of earlier moves). TFT does quite well playing against itself (CC$\rightarrow$ $[3,3]$ at each move). However, TFT-specific "predators" are readily constructed. For example, Anti-TFT (ATFT) operates like TFT except that it always defects on the first move; the first encounter yields a payoff of 5 for ATFT, giving it an initial scoring edge that TFT cannot overcome in the subsequent suite of mutual defections (DD$\rightarrow$ $[1,1]$). Tit-for-two-tat (TF2T) is a "generous" strategy that grants the opponent a chance to cooperate anew after a first defection, i.e., it requires two D's in a row from the opponent to switch its own moves from C to D. ALLC always cooperates, and represents the ultimate "sucker" strategy in the IPD.

Beginning with the pioneering work of Axelrod (1987) , searches for optimal IPD strategies have been made by *evolving* a population of strategies, ensuring that above-average strategies contribute a greater fraction of the population at the next generational iteration. An interesting question in this context, and one that has been subjected to great scrutiny (Boyd and Lorberbaum, 1987; Nowak and Sigmund, 1992; Nowak and Sigmund, 1993; Fogel, 1993), is whether there are IPD strategies that are evolutionarily stable, i.e., that do well against themselves yet can resist invasion by prey-specific strategies (such as ATFT preying upon TFT). As far as deterministic strategies are concerned, the answer has been shown to be "no" (Boyd and Lorberbaum, 1987), but in the context of stochastic strategies, TFT was found to be a very useful intermediate step towards the emergence of very stable strategies (Nowak and Sigmund, 1992). Mistakes in the implementation of a move, playing cooperation instead of defection or vice-sersa, allow evolutionary stability (Boyd, 1989). Lindgren (1991) found an evolutionary stable strategy with a memory of four while studying strategies with different memory size. The introduction of multiple choices induce diversity in behavior which make the cooperation more robust in noisy environment (Chong and Yao, 2005; Darwen and Yao, 2002).

Genetic algorithms (hereafter GAs) represent one class of population-based evolutionary algorithms that have been used extensively in the study of the IPD. In classical GAs, a crossover operator exchanges substrings between binary strings defining two population members selected for breeding. In the presence of selection pressure, this can lead to an exponential increase in the frequency of substrings that confer their bearers above-average fitness in the population (Holland, 1992). Evolution then becomes primarily a matter of searching for these advantageous "building blocks", and combining them in single individuals in the course of breeding (Holland, 1992; Spears, 1993). Mutation, random changes of single bits in an individual's defining string, is seen as a secondary process, although it is needed to maintain variability across the population, and, in doing so, guards against premature convergence.

It has proven surprisingly difficult to empirically and unambiguously demonstrate the usefulness of crossover in the context of specific search and optimization problems tackled with GAs (Fogel and Atmar, 1990; Spears, 1993; Culberson, 1994; Jones, 1995), or to identify the various roles played by crossover in the evolutionary search process (Spears, 1993; Holland, 2000). Yet, crossover seems ubiquitous in biological genetic systems. Holland (2000) lists four main reasons given to explain this prevalence of crossover, which he ranks in order of importance as follows: (1) recombining building blocks; (2) providing persistent variation to escape adaptive targeting by "predators"; (3) repairing mutational damage; (4) generating large jumps in genetic "parameter space". In principle, all four of these roles should carry over to the GA context.

In this paper, we carry out and investigate a series of genetic algorithm-based evolutionary simulations of the iterated prisoner's dilemma. The simulations offer an evolutionarily dynamical environment in which, among other things, the role of crossover can be examined. Our experimental design is described in §2, and our time series analysis technique in §3. Results are presented and discussed in detail in §4. We then offer some speculative ideas regarding the role of crossover in sustaining long-range temporal correlations in the evolving population (§5). We close in §6 by summarizing our main results and arguments, and their possible implications for evolutionary dynamics at large.

2 Evolutionary simulations of the Iterated Prisoner's Dilemma

2.1 Experimental setup

Following Axelrod (1987) , we use a standard GA to evolve a population of $M = 100$ reactive and deterministic IPD strategies up to $N = 10^7$ generational iterations. Stochastic tournament selection is used to determine which strategies in the current generation contribute offspring to the next. Each individual strategy confronts 10 randomly-selected population members (excluding itself), each time playing 66 game moves. An average score-per-move is then computed for each population member. A subset of strategies having collected higher-than-average scores are then isolated, and breed the next generation amongst themselves. Population replacement takes place by making M/η copies of each of the top-η subset of strategies, pairing them randomly, and applying to each corresponding pair of defining strings the usual GA operators of crossover and mutation. Elitism is enforced, i.e., the best strategy of the current population is always copied intact once into the next generation. With M offspring strategies so constructed, the next tournament then opens anew. All this amounts to rather severe selection pressure; in the absence of recombination and mutation, the takeover time for the best initial strategy is found (empirically) to be about 6 iterations for $\eta = 20$.

Strategies are encoded as a 6-level-deep binary tree (see Figure 1 for two 4-level-deep examples). Starting at the top of the tree, a downward path is followed according to the opponent's previous five moves, ending in a "decision", C or D, for the current game move. The strategy is thus defined by the binary values assigned to the tree's 63

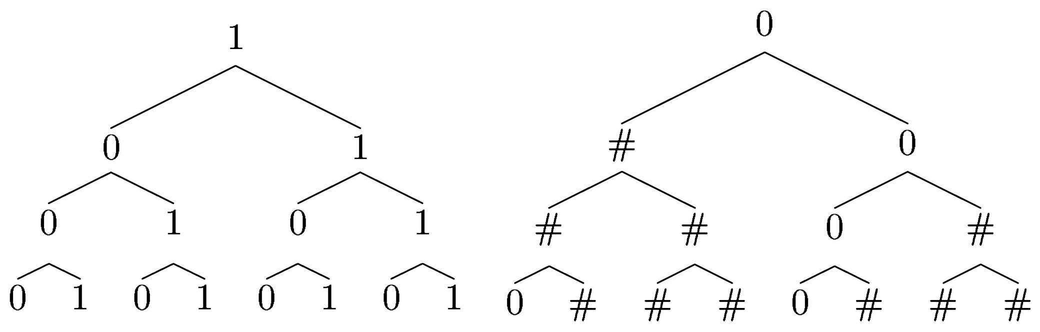

Figure 1: A 4-level version of the 6-level binary tree used to encode strategies, for the TFT (left) and ATFT (right). The coding is such that 1 ≡Cooperation, 0 ≡Defection and "#" is a "don't care" symbol. Starting at the root, one moves right (left) if the opponent last move in memory is C (D); the process is repeated down to the bottom of the tree, and the current move is the value thus arrived at. The two strategies encoded here would, upon meeting, play the sequence [TFT,ATFT]: [1,1], [2,3], [4,6], [8,12], [8,8], [8,8],... amounting to [C,D] on the first move, followed by [D,D] ever after.

nodes (1 ≡C, 0 ≡D). Cooperation (defection) by the opponent on the previous move triggers the next-move decision encoded on the right (left) branch originating from the current node. With these numbered layer by layer starting at the top, a strategy ends up being defined by a 63-bit-long string. Instead of taking only the last row as Axelrod (1987) did, we prefer to have strategies that can make decision without full memory load instead of creating a virtual one for the first moves that never occurred. Encoding the decision as a tree enable actual strategies for the first, and all later moves. In the first 5 moves of a given confrontation, when the information needed to move down the whole tree is not yet available, the scan stops once information on prior moves runs out (level 1 on first move, level 2 on second, etc., up to move five, from which point the full tree depth is utilized). Reading the tree from left to right at each row, starting at the top (root of the tree) and going through all the row until the last one, we end up with the string used in the simulations. Under this representation, TFT is defined by the string `1010101...` (Fig. 1, left), ALLD by `0000000...`, and TF2T by `111011101110...`. The prey-specific ATFT is encoded by "0" at locii 1,3,6,12,24,32,48, and "#" everywhere else, where "#" is a "don't care" symbol, indicating that the corresponding node plays no role against TFT (see Fig. 1, right). Note that only two bit-flip mutations, at locii 1 and 3, are needed to turn a TFT into an ATFT. As seen from the example of TFT and ATFT, not all nodes are used. In this confrontation the nodes represented by "#" in ATFT are not read.

We use uniform one-point crossover (with fixed probability p_c) and uniform bit-flip mutation (fixed probability p_m at each bit). In what follows the value of p_m is set at $1/n$, where $n = 63$ for a 6-level deep binary tree is the number of bits in the defining string. Note that this then implies $p_m \ll p_c$, which is the usual case in most standard GAs.

For ease of reference, Table 2 lists the defining parameters of our "reference" simulation 1. Although most of the discussion to follow focuses on this reference simulation,

Table 2: Parameters of reference simulation

Parameter	Value
Generational iterations N	$2.5{\times}10^6$
Population size (M)	100
binary tree depth	6
string length (n)	63
confrontations per iteration	10
IPD moves per confrontation	66
Breeding subset size (η)	20
crossover probability (p_c)	0.7
mutation probability (p_m)	1/63

and three no-crossover counterparts ($p_c = 0$), we also carried out similar simulations for crossover probabilities in the full range $0 \leq p_c \leq 1$, and mutation probability ranging from one fifth to 12 times the reference value $p_m = 1/63$, which corresponds to mutation affecting on average one bit per offspring string bred. We also ran some simulations at higher and lower selection pressures, by setting the breeding subset parameter η to values of 10 and 50, respectively. The results reported below do not depend sensitively on numerical choices for these parameters, within reasonable bounds.

The simulation starts with purely random strategies, constructed by assigning binary values of "0" or "1" with equal probabilities to every node of the tree. Some simulations were also seeded by artificially inserting a single TFT player in the otherwise random initial population, but this did not lead to any significant differences in the overall behavior across the length of the simulations.

2.2 Representative simulation results

Let $\mathbf{s}(i)$ represent the array containing the average score-per-move (hereafter simply "score") achieved by each strategy within the current population at generational iteration i:

$$\mathbf{s}(i) = \{s_1(i), s_2(i), ..., s_M(i)\} \ , \tag{1}$$

where M is the population size; define now the time series s of the score achieved at each iteration by the current best strategy, i.e.,

$$s(i) = \mathrm{Max}(\,\mathbf{s}(i)\,) \ . \tag{2}$$

The solid line on Figure 2 is a 1200-iteration representative portion of such a time series, extracted from a typical simulation run. The mean score of $s(i)$ across the whole 2.5×10^6 iterations of this simulation is 2.50, with a r.m.s. deviation of 0.19. For perspective, recall that for our adopted IPD payoff matrix, TFT playing against itself yields a score of 3, while ALLD playing against the ultimate "sucker" strategy ALLC scores 5 (with ALLC getting zero).

The first thing to note upon examining the simulations in their totality is that, indeed, there are no strictly evolutionarily stable deterministic strategies for the IPD. Even when run repeatedly from different initial populations, convergence to a final,

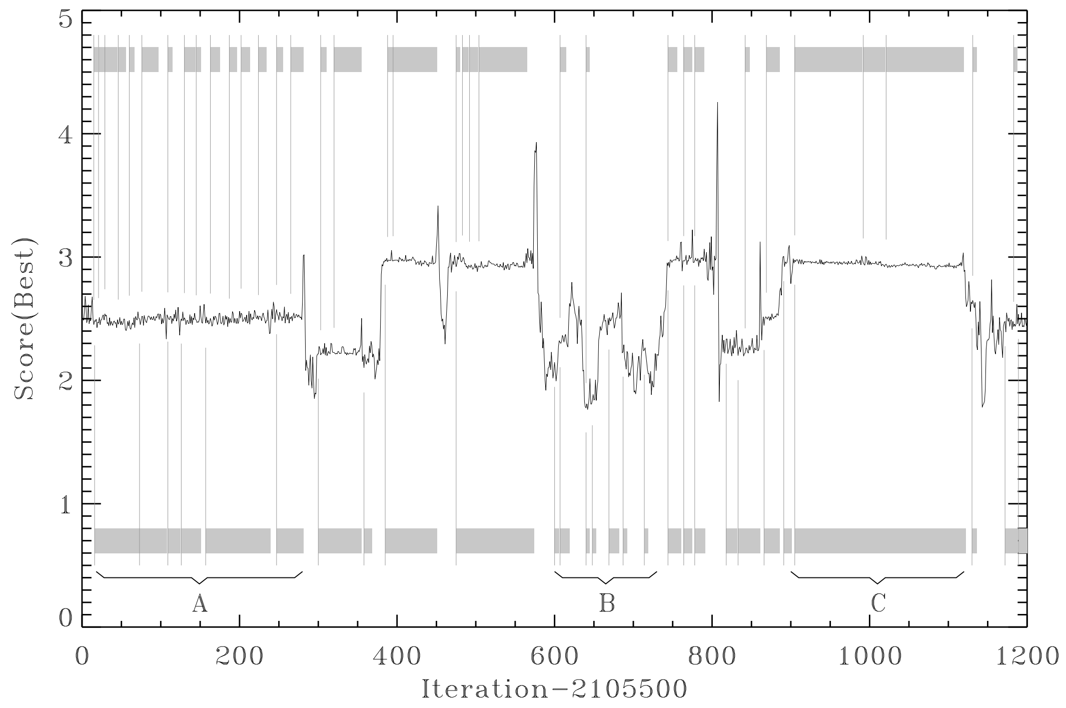

Figure 2: A 1200-iteration representative portion of the time series, for best score at each iteration, 2105500 iterations after the beginning of a typical simulation. The two sets of gray tick marks and horizontal bars indicate respectively the beginning and duration of stable phases, as determined by the recursive running mean technique described in §3, under two settings of the tolerance parameter, $\epsilon = 0.05$ (top) and $\epsilon = 0.1$ (bottom).

ever-stable population-wide strategy never occurs. However, the simulations do reveal extended periods of stability, whether measured in terms of best-score (as on Fig. 2), population-averaged score, or even "genotypic" measures such as Hamming distances. Strictly speaking we should refer to such phases as "metastable", although for the sake of brevity we will usually retain the appellation "stable". The properties and statistics of these stable phases, as well as of the intervening unstable periods, are the focal point of what follows.

The general pattern of long stable phases (subsegments labeled "A" and "C" on Fig. 2), punctuated by disturbances and relatively short-lived unstable phases, resembles what has been dubbed "punctuated equilibrium" in the biological evolutionary literature. In the fossil record, this pattern is increasingly recognized as the norm rather than the exception (Gould and Eldredge, 1993). Lindgren (1991) found a similar dynamic when studying the evolution of the memory length —equivalent here to our binary tree depth— in the prisoner's dilemma. Here it arises when an evolutionarily (meta)stable strategy appears in the population. Such strategies do well when playing IPD against one another, and, provided they can gain a foothold in the existing population, they will lead to evolutionary stability. Because the population as a whole is getting more cooperative, strategies can drop their guard against defection. They may also be susceptible to a new strategy that "discovers" how to exploit whatever failings they might present (Darwen and Yao, 1995). This shows up clearly here, as a prominent spike terminating most stable phases (end of "A", and immediately prior to "B", on Fig. 2). Such predator strategies, however, usually exploit some specific Achilles' heel of an equally specific prey-strategy. They are often not particularly efficient playing IPD against each other, i.e., they are targeted predator adaptation. Because our simulations are carried out with high selection pressure, the formerly stable prey strategies are rapidly decimated. This is usually followed by an unstable period of lower-than-average score, often of significant duration, during which the distribution of strategies within the population —and thus the score measures— vary widely ("B" on Fig. 2). Stability is restored once a metastable strategy emerges again in the population. Note that all stable phases are not created equal; on Fig. 2, the fluctuations about the mean for plateau "A" are distinctly larger than for plateau "C", reflecting the degree to which the dominant strategy is indeed evolutionarily stable. These overall evolutionary patterns are quite characteristic of all simulations performed at varying crossover and mutation probabilities, for different population size and/or groupings, with or without elitism, for different breeding subset size η, etc.

Shorter simulations with different binary tree depth and number of game moves per confrontation were also carried out, as listed on Table 3. The Table gives the fraction of simulation time spent in stable phases, for our reference simulation 1 (see Table 2), as well as the three variants to be introduced shortly below. Across all simulations, only those with binary tree depth of at least 5 levels (memory of 4 moves) yield a punctuated equilibrium-like evolutionary dynamic, where a high fraction of simulation time is spent in stable phases, in agreement with the results of Lindgren (1991) . We use long enough confrontation (66 game moves) so that the initial transient during which the whole tree depth is not yet used to reach a decision does not overly influence the strategy's final score on a given confrontation. For long enough confrontations, augmenting the breeding subset size ($\eta = 10, 20, 50$) increases the fraction of time spent in stable phases when punctuated equilibrium-like dynamics is observed (not shown).

Simulations were done on a SunFire V880 computer running UltraSPARC III processors with $900MHz$ clock. Representative simulations were taking on the order of 100 hours to run and analyze on a single processor.

3 Extracting strategy lifetimes

Throughout the foregoing analysis, we adopt a "phenotypic" viewpoint, in that we analyze the simulation results in terms of "observable" quantities, primarily time series of various score measures such as on Fig. 2. A central requirement of such an analysis here is to automatically (and reliably) identify phases of stable scores on such time series. Among the many possible plateau-identification schemes, we use the following two.

We first consider an algorithm based on recursive running means. From some specified starting point j in the (discrete) time series of best scores $s(i)$, we compute a running average

$$A(j,k) = \frac{1}{k-j+1}\sum_{i=j}^{k} s(i) \qquad (3)$$

Table 3: Effect of binary tree depth and game moves per confrontation

Binary tree depth	Moves per confrontation	Fraction stable			
		sim 1 (●)	sim 2 (△)	sim 3 (⋄)	sim 4(∗)
2	20	0.006474	0.006306	0.001560	0.0036640 ($\hat{d} = 0.18717$)
2	40	0.005432	0.005184	0.001260	0.0028400 ($\hat{d} = 0.18885$)
2	70	0.004682	0.004590	0.001196	0.0023940 ($\hat{d} = 0.19058$)
2	80	0.005140	0.005042	0.001050	0.0026840 ($\hat{d} = 0.18971$)
3	20	0.027126	0.045366	0.005874	0.0054720 ($\hat{d} = 0.15007$)
3	40	0.015782	0.040284	0.004460	0.0045920 ($\hat{d} = 0.128921$)
3	70	0.012924	0.035392	0.004044	0.0045680 ($\hat{d} = 0.125676$)
3	80	0.011934	0.037426	0.004006	0.0042020 ($\hat{d} = 0.126191$)
4	20	0.175720	0.150316	0.048360	0.014354 ($\hat{d} = 0.106015$)
4	40	0.108454	0.170786	0.039976	0.013356 ($\hat{d} = 0.095579$)
4	70	0.070144	0.100586	0.030772	0.012304 ($\hat{d} = 0.095047$)
4	80	0.066980	0.087046	0.029448	0.010646 ($\hat{d} = 0.095531$)
5	20	0.309232	0.285546	0.188784	0.038856 ($\hat{d} = 0.079495$)
5	40	0.453592	0.414696	0.295642	0.048150 ($\hat{d} = 0.079495$)
5	70	0.474070	0.450648	0.318606	0.051406 ($\hat{d} = 0.078484$)
5	80	0.463210	0.450440	0.319714	0.051690 ($\hat{d} = 0.078620$)

where the averaging ends once an iteration k ($> j$) has been found such that $|A(j,k) - s(j)|$ rises above some preset tolerance ϵ (set to 0.05 in what follows, unless otherwise noted). The process is repeated recursively starting now at iteration $j+1$ to compute the running mean, but keeping the same "origin" at $s(j)$ for the stopping criterion. The smallest k value (k^*, say) found through the recursive iteration becomes the end of the plateau, which is then assigned a lifetime $\tau = k^* - j$. The search for the next plateau then begins anew by resetting the origin at $k^* + 1$.

We also made use of a derivative-based algorithm, whereby the time series is first smoothed using a 21-iteration-wide boxcar averaging filter. This filter, assign the average value of the 20 surrounding values, ten before, ten after and the current to the current position. With the filtering we loose 20 values: the first and last ten. A plateau is deemed to end, and the next one begin, whenever the absolute value of the first derivative of the smoothed time series exceeds some preset tolerance ϵ' (usually $= 0.003$).

This task of reliably and automatically identifying stable phases turns out to be surprisingly delicate. Some of the pitfalls are readily seen upon comparing the two plateau-identification runs displayed along the top and bottom parts of Fig. 2. The gray vertical lines mark the onset of stable phases, and the horizontal gray bars indicate their corresponding lifetime. The two sequences were obtained using the recursive running-mean method described above, with the top sequence corresponding to our adopted tolerance value $\epsilon = 0.05$, and the bottom sequence to twice that tolerance. Not surprisingly, the smaller tolerance breaks up noisy plateau ("A" on Fig. 2) in a larger number of shorter stable phases than greater tolerance does. The "eye" perhaps would prefer to see a long but noisy stable phase extending over the complete "A" interval. On this basis one might be tempted to give preference to higher tolerance levels. Consider however the unstable phase "B"; the low-tolerance run finds only two short stable phases there, while the higher tolerance run identifies seven. The former now appears preferable. Focus now on the low-noise interval "C". The high tolerance run identifies it as a single stable phase, which seems eminently sensible. Yet closer examination reveals a small but significant drop in the mean score, taking place approximately halfway through the interval. The low tolerance run correctly detects this, although it also inserts a short intermediate phase following a short noisy episode.

We have carried out all analyses of the simulation time series with both a low and high tolerance value, as well as with the derivative-based method described above. Unless explicitly stated otherwise, all results discussed in what follows use the recursive running mean method with tolerance value $\epsilon = 0.05$, as on the top part of Fig. 2. These in fact correlate very well with the stable phases identified using the derivative-based scheme with $\epsilon' = 0.003$. Fortunately, our results and interpretations turn out to be robust with respect to the choice of plateau identification technique and associated internal parameter settings (within reasonable bounds of course).

Although the takeover time associated with our tournament selection procedure is quite short (anywhere from 5 to 10 iterations, depending on the adopted value of the breeding subset size η), it still represents a lower bound of what can be meaningfully dubbed a "stable phase". Consequently, unless noted otherwise, we will exclude from the foregoing analysis stable phases of duration inferior to five generational iterations.

4 Simulation results: crossover and stability

Of particular interest in what follows is the role played by crossover in the evolutionary dynamics of the IPD. The most straightforward way to assess the usefulness of crossover is to compare various measures in a simulation with crossover, and a second simulation with crossover turned off but otherwise identical. Note however that for a given mutation probability p_m, including crossover increases the probability of disruption at breeding (elitism notwithstanding). For the 63-bit strings used here, the disruption probability is $d_1 = 1 - (1 - p_m)^{63}(1 - p_c)$ with crossover included, and $d_2 = 1 - (1 - p_m)^{63}$ without (Holland, 1992) . The simulation with crossover must thus be compared also with a no-crossover simulation having a different mutation rate p_m^* such that $d_1 = d_2$. For our reference simulation with $p_m = 1/63$ and $p_c = 0.7$, this requires $p_m^* = 0.0345$. In the presence of natural selection, the disruption probability is altered by the convergence toward a smaller set of strategies than a purely random set. We calculate the empirical disruption probability ($\hat{d}$), that is the average proportion of different bits between the parents and the offspring. In the absence of crossover the empirical disruption probability is the same as the mutation probability. In the presence of crossover, the empirical disruption probability is the average Hamming distance between the parents divided by the string length (n). The simulation with crossover is also compared to a simulation with the same empirical disruption probability ($\hat{d} = 0.078273$). $\hat{d}$ is constant throughout each simulation and does not differ between stable and unstable phases, so we only need one simulation with empirical disruption probability to the simulation with crossover (Table 4, Line 4, 5 and 6). Consequently, we will compare results across four distinct "baseline" simulations, the defining parameters of which are listed in Table 4 below. The first is the reference simulation, the second is its no-crossover counterpart, the third is a no-crossover simulation with $p_m = 0.0345$ and the fourth is a no-crossover simulation with $p_m = 0.078273$, as per the above analysis. In this way we hope to measure the effects of crossover apart from its disruptive action at breeding. Other simulation control parameters have values as listed in Table 2, which also lists a suite of measures extracted from these three simulations, which are used in the foregoing discussion to assess the role of crossover.

Table 4: Defining parameters and characteristics of four baseline simulations

Line	Measure	Symbol	Def. eq.	1 (●)	2 (△)	3 (⋄)	4 ($*$)
1	Crossover probability	p_c	–	0.7	0	0	0
2	Mutation probability	p_m	–	0.0159	0.0159	0.0345	0.0783
3	Disruption probability	d	–	0.89	0.64	0.89	0.99
4	Empirical disruption probability	$\langle \hat{d} \rangle$	§4	0.0783	0.0159	0.0345	0.0783
5	Average empirical disruption probability in stable phases	$\langle \hat{d} \rangle_S$	§4	0.0790 ± 0.0146	0.0159 ± 0.0016	0.0345 ± 0.0023	0.0782 ± 0.0034
6	Average empirical disruption probability in unstable phases	$\langle \hat{d} \rangle_U$	§4	0.0776 ± 0.0153	0.0159 ± 0.0016	0.0345 ± 0.0023	0.0783 ± 0.0034
7	Number of stable phases ($\tau \geq 5$)	K	–	102992	92351	82892	22313
8	Run-averaged best score	$\langle s \rangle$	(4)	2.62 ± 0.32	2.69 ± 0.35	2.64 ± 0.31	2.40 ± 0.16
9	Run-averaged mean score	$\langle \overline{s} \rangle$	(5)	2.41 ± 0.34	2.46 ± 0.37	2.36 ± 0.35	2.01 ± 0.16
10	Fraction stable	ϕ	(6)	0.469	0.450	0.318	0.053
11	Average duration of stable phases	$\langle \tau \rangle$	(7)	11.4	12.2	9.6	5.9
12	Average duration of cooperative stable phases	$\langle \tau \rangle_C$	§4.3	16.1	15.6	11.6	7.3
13	Average duration of others stable phases	$\langle \tau \rangle_O$	§4.3	8.8	8.2	6.9	5.9
14	Average duration of unstable phases	$\langle \Delta\tau \rangle$	(7)	15.3	17.9	23.4	108.0
15	Averaged best score, stable phases	$\langle s \rangle_{\mathrm{S}}$	–	2.72 ± 0.24	2.80 ± 0.23	2.80 ± 0.21	2.46 ± 0.11
16	Average best score, unstable phases	$\langle s \rangle_{\mathrm{U}}$	–	2.53 ± 0.36	2.59 ± 0.40	2.55 ± 0.33	2.40 ± 0.16
17	Averaged mean score, stable phases	$\langle \overline{s} \rangle_{\mathrm{S}}$	–	2.60 ± 0.27	2.70 ± 0.25	2.64 ± 0.25	2.14 ± 0.15
18	Average mean score, unstable phases	$\langle \overline{s} \rangle_{\mathrm{U}}$	–	2.22 ± 0.31	2.25 ± 0.33	2.22 ± 0.30	2.00 ± 0.16
19	Run-averaged score difference	$\langle \Delta s \rangle$	(10,7)	0.18	0.21	0.25	0.36
20	Run-averaged r.m.s. deviation	$\langle \sigma \rangle$	(10)	0.130	0.136	0.146	0.184
21	Average score difference in stable phases	$\langle \Delta s \rangle_{\mathrm{S}}$	(14)	0.083	0.075	0.119	0.284
22	Average r.m.s. deviation in stable phases	$\langle \sigma \rangle_{\mathrm{S}}$	(14)	0.110	0.109	0.116	0.172
23	Averaged score difference in unstable phases	$\langle \Delta s \rangle_{\mathrm{U}}$	(15)	0.280	0.326	0.309	0.366
24	Average r.m.s. deviation in unstable phases	$\langle \sigma \rangle_{\mathrm{U}}$	(15)	0.148	0.160	0.161	0.185
25	Average change in r.m.s. deviation ($\tau \geq 50$)	$\langle \delta\sigma \rangle_{\mathrm{S}}$	(17)	-0.050	-0.062	-0.060	N/A
26	Average r.m.s. deviation ($\tau \geq 50$)	$\langle \sigma \rangle_{\mathrm{S}}$	(17,14)	0.110	0.101	0.093	N/A

Note that in Table 4 and later on in our subsequent discussion, standard deviations on population-averaged and/or run-averaged mean values are only listed when the corresponding statistical distribution of the quantity being averaged is at least roughly symmetrical about the mean; this is in general true for raw score measures (discussed in §4.1), but not for lifetime distributions (§4.2) or heterogeneity measures (§4.4).

We also ran the same simulations with noise in the interpretation of the strategies. Strategies were rarely making mistake when reading the last node of their path in the binary tree (1%). The main results did not change.

4.1 Score measures

Is crossover useful in evolving high-scoring deterministic strategies in the context of the IPD? Axelrod (1987) originally answered "yes" to this question , but his simulations were only run over 50 generations. In all likelihood, what he observed is the better initial exploration of parameter space carried out through the use of crossover, so that good strategies emerge faster. This is indeed a very useful thing, and is in line with Holland's building block hypothesis. Because our simulations were carried over a great many more generational iterations, we are in a position to answer a somewhat different set of questions.

In the long run, is crossover helping to produce and sustain higher scores than in its absence? Consider first the run-averaged best score:

$$\langle s \rangle = \frac{1}{N} \sum_{i=1}^{N} s(i) \; , \tag{4}$$

where N ($= 2.5 \times 10^6$ here) is the number of generational iterations. Line 8 of Table 4 lists values for this quantity, along with r.m.s. deviations about the average, in our four baseline simulations. Crossover does not seem to have any statistically significant effect here. A related but distinct score measure is the simulation-averaged mean score for the *entire* population, rather than the best performing member at each generational iteration:

$$\langle \overline{s} \rangle = \frac{1}{N} \sum_{i=1}^{N} \overline{s}(i) = \frac{1}{MN} \sum_{i=1}^{N} \sum_{m=1}^{M} \mathbf{s}_m(i) \; . \tag{5}$$

In the function optimization context, some authors have noted that while crossover does not seem to improve significantly the maximum fitness, it does lead to a significant increase in the population-averaged fitness (see Spears (1993) , and discussion therein). For the four simulations of Table 4, these run-averaged mean scores are again all well within each other's r.m.s. deviations (see Line 9 on Table 4). This indicates that in the present context, crossover has indeed very little overall influence on global score measures.

When these score measures are computed for stable and unstable phases separately, one clear pattern does emerge. Both the run-averaged best score and the run-averaged, population-averaged mean score are significantly higher in stable phases than in the intervening unstable epochs, a trend present in all four baseline simulations (cf. Lines $15 - 16$ and $17 - 18$ in Table 4). It pays to be stable, for the best strategy as well as for the population as a whole. While this suggests distinct evolutionary dynamics in stable and unstable phases, once again crossover does not seem to have a significant direct effect, as all score measures and associated standard deviations are similar in all four simulations except for the average mean score in the stable phase of simulation 4 that is marginally lower than the three other simulations (cf. Line 17 in Table 4).

Extreme values of disruption probabilities (cf. Line 3 in Table 4) correspond to extreme values of performance measures related to score in Table 4 (cf. Lines $8-9$ and $15-18$). These entries show that the lowest disruption probability (Simulation 2) yields the highest scores, while the highest disruption probability (Simulation 4) produces the lowest scores. Although this observation is not a rule, as can be seen from intermediate disruption probability (Simulation 1 and 3) and standard deviations, it resembles the pattern observed by Ishibuchi and Namikawa (2005) on the effect of mutation and crossover on the percentage of mutual cooperation, a surrogate measure for score.

4.2 Stability measures

Is crossover perhaps favoring evolutionary stability? To answer this question we first consider the fraction ϕ of the whole simulation spent in stable phases, as identified above:

$$\phi = \frac{1}{N}\sum_{k=1}^{K}\tau_k. \tag{6}$$

Here an interesting difference emerges. Restricting the analysis to stable phases having $\tau \geq 5$, one finds that in simulation 1, 47% of the simulation is spent in stable phases. This fraction falls slightly to 45% when crossover is turned off (simulation 2), the fall continues to 32% in simulation 3 which has identical disruption probability at breeding than simulation 1, and drops to 5% in simulation 4 where the empirical distribution probability its equivalent to simulation 1 (see Line 10 in Table 4).

Broadly speaking, two explanations can be put forth here. Either crossover increases the duration of stable phases, or it decreases the duration of unstable phases (or both). To disentangle these possibilities, we begin by constructing the probability distribution function (PDF) of stable phase lifetimes, $f(\tau)$, i.e., the fraction of lifetimes that have values lying between τ and $\tau + \mathrm{d}\tau$. This is quite straightforward once a list of lifetimes τ_k $(k = 1, ..., K)$ has been extracted from a given time series, as described in §3. The results of such an exercise are shown on Figure 3, for the four baseline simulations of Table 4. The distributions have been constructed for logarithmically constant binsizes, to reduce the statistical noise level in the distributions' tails, where they most differ.

The mean durations of stable phases are computed by integrating their corresponding PDF:

$$\langle\tau\rangle = \int f(\tau)\tau\mathrm{d}\tau \qquad \left(\equiv \frac{1}{K}\sum_{k=1}^{K}\tau_k\right), \tag{7}$$

and are indicated by symbol-coded vertical line segments on Fig. 3. At equal disruption probability (Simulation 1 versus 3), stable phases last about 20% longer on average when crossover is present. At equivalent empirical disruption probability (Simulation 1 versus 4), stable phases are almost twice longer when crossover is present. This is a significant difference, but insufficient to explain all the aforementioned differences in time spent in stable phases. One must then conclude that less time is spent in *unstable* phases when crossover is present. This is readily verified upon computing the PDFs

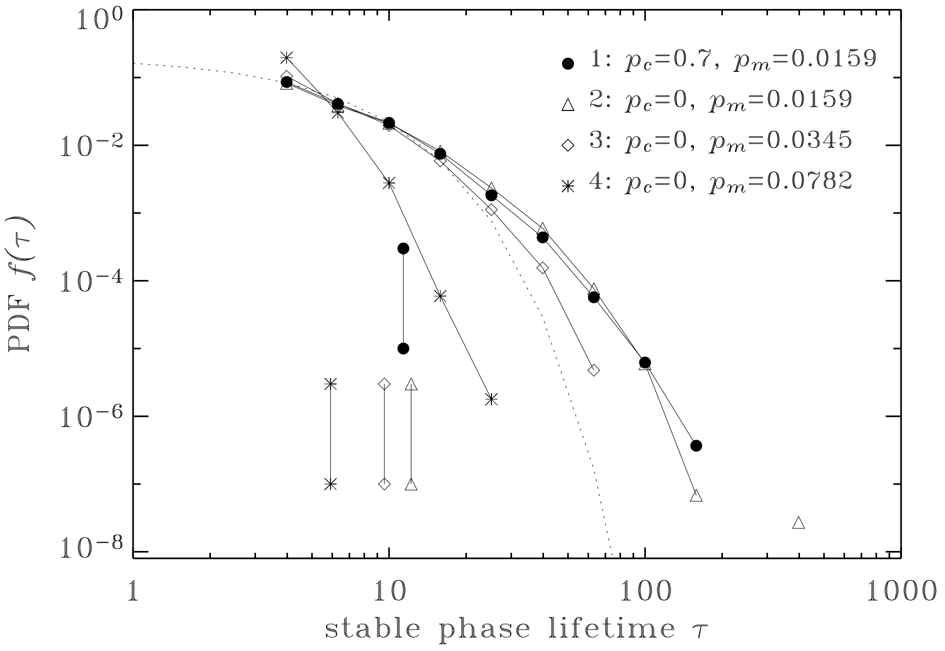

Figure 3: Probability distribution functions $f(\tau)$ for the lifetimes (τ) of stable phases, in the four baseline simulations of Table 4 (•, △, ⋄, ∗ for simulations 1, 2, 3, 4, respectively). The dotted line is an exponential PDF, i.e., $f(\tau) \propto \exp(-\tau/\tau_0)$, fitted to the first three data points of simulation 1. The horizontal position of the symbol-coded vertical line segments indicate the mean values of the corresponding PDFs.

$f(\Delta\tau)$ of unstable phases durations $\Delta\tau$. These are simply defined as the number of iterations between the end of one stable phase and the beginning of the next:

$$\Delta\tau_k = j_{k+1} - (j_k + \tau_k) \ , \qquad k = 1, ..., K-1 \ . \tag{8}$$

where j_k is the iteration marking the beginning of stable phase k (of duration τ_k). Figure 4 shows the PDFs for this quantity in our three usual simulations, in the same format as for the PDFs of stable phases lifetimes presented on Fig. 3. Mean values $\langle\Delta\tau\rangle$, computed via eq. (7) with $\Delta\tau$ replacing τ, are listed on Line 14 of Table 4. Note how, at equal disruption probability (simulations 1 vs 3), crossover at $p_c = 0.7$ nearly halves the mean duration of unstable phases, and still reduces it significantly ($\sim 15\,\%$) at fixed mutation probability (simulations 1 vs 2). At equal empirical disruption probability (simulations 1 vs 4), the unstable phases more than seven times longer than with crossover. Even if the distributions of simulations 1, 2 and 3 look similar to the eye on this log-log plot, they do differ significantly. The p-value of a Kolmogorov-Smirnov test associated with all combinations of comparison are lower than 10^{-6}, so the hypothesis that the distributions come from the same function can be rejected. To outline the difference of the distributions functions of unstable phase lifetime, the Q-Q plot is used to amplify the difference on the tail of the distributions, it is the representation of the

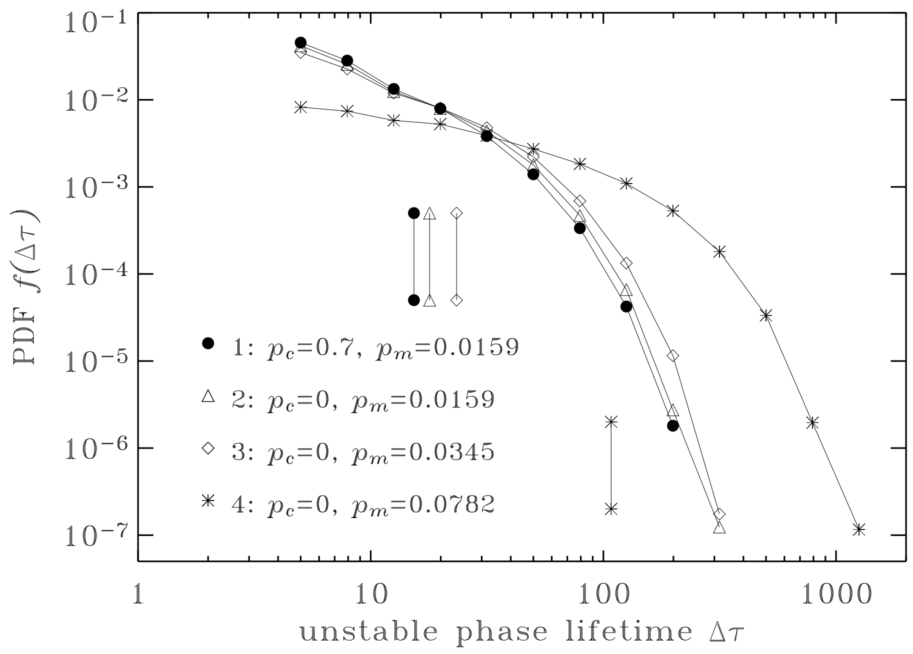

Figure 4: PDF of unstable phase durations, for our four baseline simulations. The horizontal position of the symbol-coded vertical lines segment indicate the mean duration of unstable phases for each of the three PDFs. This mean value for simulation 2 (without crossover) is 20% higher than for simulation 1 with $p_c = 0.7$ (18 iterations, versus 15 when crossover is included), and a little over 50% larger at equal disruption probability in simulation 3 ($\diamond$). The mean value in simulation 4 ($*$) is more than 700% larger than in simulation 1 (108 iterations versus 15).

quantiles of the first distribution against the quantiles of the second distribution (Law and Kelton, 2000) (Fig. 5). There is a difference between all tree simulations, since none of the Q-Q plot are following the dashed line representing two distributions from the same function.

One might conclude that this is a direct consequence of crossover's enhanced exploratory capability through building block assembly. After the appearance of an adaptively targeted predator, the formerly stable population is rapidly decimated, which is followed by the equally rapid demise of the targeted predators, since these are typically not particularly good at playing IPD against one another. Recovery to a new stable phase must happen through the discovery (or rediscovery) of another evolutionarily (meta)stable strategy. Crossover can in principle accelerate this recovery process, for example by allowing different types of "jumps" through strategy space (one of crossover's secondary role, according to Holland, cf. §1), and/or recombining building blocks, if some are present in the problem space (crossover's most important role, again according to Holland).

Examination of time series such as on Fig. 2 reveals that many stable phases seem

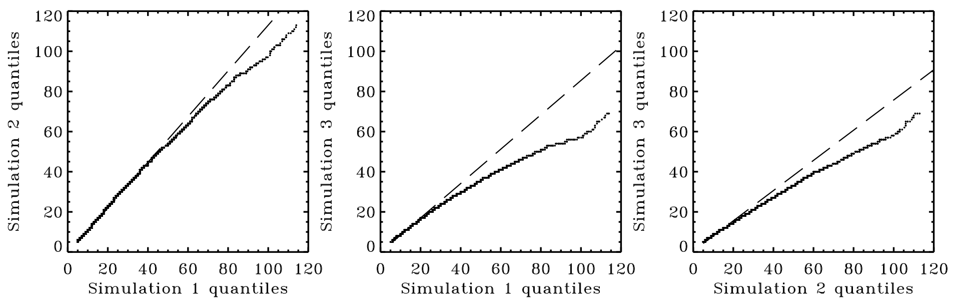

Figure 5: QQ-plot compares the quantiles of two distribution. If both distributions are the same, all points fall on the dashed line. Here we compared the distributions of unstable phase lifetime. All distributions have significant differences on their tails and failed a Kolmogorov-Smirnov test with a p-value less than 10^{-6}.

to have similar mean scores (such as A and C on Fig. 2). First define a plateau-averaged best score $\langle s \rangle_k$ over the duration τ_k of the k^{th} stable phase beginning at iteration number j_k:

$$\langle s \rangle_k = \frac{1}{\tau_k} \sum_{i=j}^{j_k+\tau_k-1} s(i) \ , \qquad k = 1, ..., K \tag{9}$$

where, as before $s(i) \equiv \max(\mathbf{s}(i))$ is the time series of best score (cf. Fig. 2). Figure 6 shows PDFs of this quantity, for our three baseline simulations. Remarkably, all four PDFs are peaking at 2.5, which is the mean score of a series of alternate cooperation-defection out of synchrony ($\frac{T+S}{2} = 2.5$). Another peak is common for all simulations just below 3.0 (the mutual cooperation reward), although simulation 4 barely shows this peak. Instead, simulation 4 has peaks at 2.5 and 2.25 the mean of two strategies that cooperate half the time without any consideration for the previous moves ($0.25S + 0.25P + 0.25R + 0.25T = 2.25$). We will refer to those scores as the combination of strategies that generate them, even if more complicated sequences can create the same outcome. Other peaks can be found below 2.4 and at 2.6, but they do not represent any particular set of strategies. In every simulation most of the stable phases are either close to mutual cooperation or alternate cooperation-defection out of synchrony.

Interestingly, there are no stable phases above 3.0. In other words, the exploitation of cooperators cannot be stable. The predator, can exploit cooperators, but won't last more than 4 iterations, which is less than the takeover time. As soon as a predator invades the population, no stable phase is possible. Stable phases are only present for best score between 2 and 3, so mutual punishment or exploitation of cooperators cannot be stable (Fig. 6). From figure 6, we see two major families of outcomes: an alternate sequence of cooperation-defection (C-D) out of phase (2.5) and mutual cooperation (3.0). Simulation 1 spent most of its stable phases in the alternate sequence of C-D outcomes. Simulation 2 and 3 spent more of their stable phases in the mutual cooperation regime. Between the two major families, there is a gap below 2.8 where no stable phases exist. This breaking point separate mutually cooperative outcomes from the others. Strategies jump between the families after a perturbation occurs.

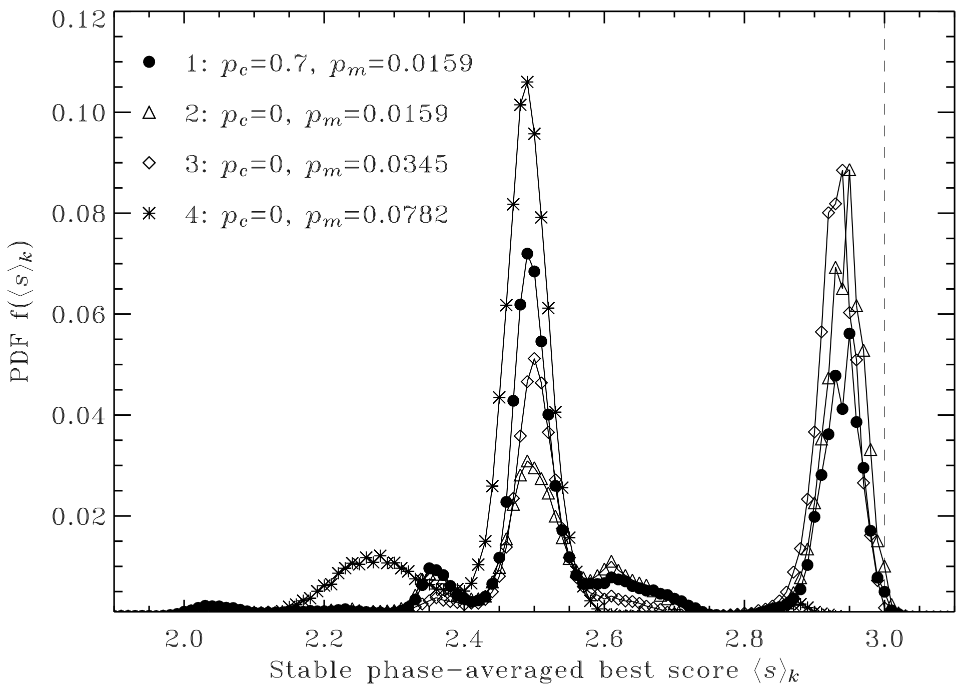

Figure 6: PDF of best score averaged over individual stable phases (see eq. 9), for the four baseline simulations. The dashed vertical line indicates the expected score for TFT playing IPD against another TFT.

4.3 Joint score-lifetime distributions

The averaged best scores in stable phases listed in Table 4 (Line 15) cannot be extracted from the PDFs plotted on Fig. 6 simply because stable phases do not have the same duration τ_k. These discrepancies, however, are an indicator of correlations between lifetimes and scores averaged over stable phases. Figure 7 is a scatter plot of the pairs $(\langle s \rangle_k, \tau_k)$ for all stable phases. Note how long-lifetime stable phases ($\tau_k \geq 10$, say) tend to clump in fairly distinct families, each with distinct score-lifetime correlations. This indicates that there exist a *finite* number of evolutionarily metastable strategies, and that it tends to be preferentially one or the other of these strategies that keep being discovered by the evolutionary process, only to be wiped out again once strategy-specific predators emerge, and reappear at some subsequent time once those predators have driven themselves to extinction by obliterating their prey.

This suggests that the evolving population achieves evasion from targeted adaptive predation by rapidly evolving from one metastable strategy to another as prey-specific predators appear in the simulation. Appearance of such predators is thus the primary trigger behind the large-scale evolutionary patterns observed in the simulation (recall, cf. Fig. 2, that most long-lifetime stable phases end up with a marked upward "spike" in the best-mean-score, indicative of the appearance of a strategy that does very well against other members of the current population). The fact that unstable phases are of significantly shorter duration when crossover is included in the simula-

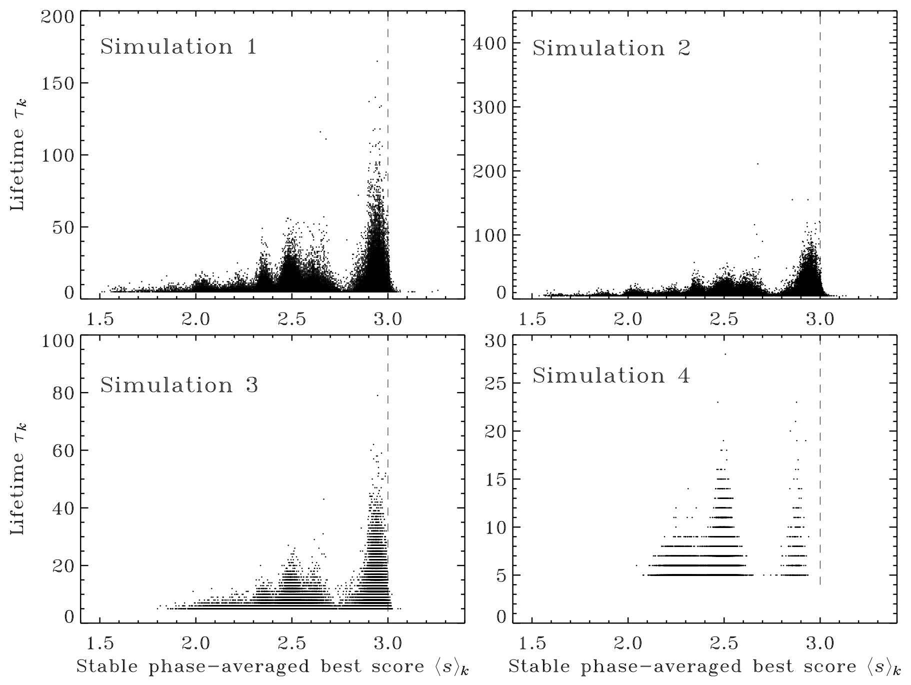

Figure 7: Distribution of stable phase characteristics plotted in the averaged best score+lifetime plane for the four baseline simulations with noise. The dashed vertical line again indicates the expected score for TFT playing IPD against another TFT. The highly inhomogeneous grouping of points is quite striking.

tion (cf. Fig. 4 and Table 4) further suggests that recovery to a new stable phase is more efficiently carried out by recombining appropriate building blocks having survived through the predatory phase and now scattered across the population, than by sequences of bit-flip mutations.

Even if the stable phases of the first three simulations are almost equally split between 2.5 and 3.0 points, the duration of the stable phases has a positive bias for mutual cooperation (Fig. 7). The mutual cooperation range in this study is 3.0 ± 0.2 and the others are from $0.0 - 2.8$ and $3.2 - 5.0$. In every simulation, even simulation 4 where the number of stable phases is small compared with the three others, the mean duration of mutual cooperation stable phases achieve longer duration (Line $12 - 13$, Table 4). At equal disruption probability and empirical disruption probability, crossover is inflating the duration of stable phases for every type of stable phases: mutual cooperation and other stable phases.

Crossover is thus enhancing evolutionary stability in two ways: first, it accelerates the recovery of stability after the spread and subsequent demise of adaptively targeted predators, by reassembling building blocks still present in the population. Second, during evolutionary stable phases it also impedes the appearance and spread of adaptively targeted predators. While the former role is readily understood in terms of building block assembly, the latter is not. Moreover, for crossover to assemble building blocks in a productive manner, a variety of these must be present in the population, which, in turn, requires genotypic diversity. We now turn to this issue.

4.4 Population heterogeneity and evolutionary stability

Recombination by one-point crossover only has a net effect if it operates on two parent strings that differ by at least two bits. In the present simulations, crossover was found to have an important effect on the duration of unstable phases, and a smaller yet still significant effect on the duration of stable phases. This implies that a significant level of variability must still exist in the population even at the end of a long stable phase.

Population heterogeneity can be measured in a number of ways in our simulations. In keeping with the spirit of working only on "phenotypic" characteristics (rather than scrutinizing the population at the genotypic level), we measure heterogeneity in terms of two score-related quantities. The first is simply the score difference Δs between the best and median individual at each generational iteration:

$$\Delta s(i) = \mathrm{Max}[\,\mathbf{s}(i)\,] - \mathrm{Med}[\,\mathbf{s}(i)\,]\ . \tag{10}$$

The second is the r.m.s. deviation σ about the mean population score at each iteration:

$$\sigma(i) = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(s_m(i) - \overline{s}(i))^2}. \tag{11}$$

These quantities can then be numerically averaged over the whole simulation ($\langle\Delta s\rangle$, $\langle\sigma\rangle$), over individual stable phases ($\langle\Delta s\rangle_k$, $\langle\sigma\rangle_k$), or over all stable or unstable phases ($\langle\Delta s\rangle_{\mathrm{S}}$, $\langle\Delta s\rangle_{\mathrm{U}}$, etc), e.g.:

$$\Big[\langle\Delta s\rangle, \langle\sigma\rangle\Big] = \frac{1}{N}\sum_{i=1}^{N}\Big[\Delta s(i), \sigma(i)\Big]\ , \tag{12}$$

$$\Big[\langle\Delta s\rangle_k, \langle\sigma\rangle_k\Big] = \frac{1}{\tau_k}\sum_{i=j_k}^{j_k+\tau_k-1}\Big[\Delta s(i), \sigma(i)\Big]\ , \tag{13}$$

$$\Big[\langle\Delta s\rangle_{\mathrm{S}}, \langle\sigma\rangle_{\mathrm{S}}\Big] = \frac{1}{\phi N}\sum_{k=1}^{K}\sum_{i=j_k}^{j_k+\tau_k-1}\Big[\Delta s(i), \sigma(i)\Big]\ , \tag{14}$$

$$\Big[\langle\Delta s\rangle_{\mathrm{U}}, \langle\sigma\rangle_{\mathrm{U}}\Big] = \frac{1}{(1-\phi)N}\sum_{k=1}^{K-1}\sum_{i=j_k+\tau_k}^{j_{k+1}-1}\Big[\Delta s(i), \sigma(i)\Big]\ , \tag{15}$$

where the same notation as in eq. (9) is used to identify the beginning (j_k) and duration (τ_k) of the K stable phases (Note again that eq. (14) is *not* the same as computing the average of the K values for $\langle\Delta s\rangle_k$, which would give equal weight to all stable phases independently of their lengths).

Numerical values for some of these quantities are listed in Table 4, Lines $19-24$. At equal disruption probability (simulations 1 vs 3 and 1 vs 4), the most significant systematic variation is found for the averaged score difference $\langle \Delta s \rangle$ (Lines $19, 21$, and 23 in Table 4), where crossover is found to reduce population heterogeneity by some tens of percent, the drop being most pronounced when only stable phases are considered ($\sim 30\%$; cf. Line 21). Examination of the PDFs for the stable phase score differences $\langle \Delta s \rangle_k$ (not shown) reveals differences in details, yet all four simulations indeed show similar distributions of score differences. Globally, crossover is only having a minor effect on population heterogeneity.

How is heterogeneity varying *in the course* of a given stable phase? To address this question we consider sequences of time series of r.m.s. deviations $\sigma(i)$ about the population-wide mean scores $\overline{s}(i)$. Figure 8 shows a sequence of time series for the quantity

$$\delta\sigma(i - j_k) = \sigma(i - j_k) - \sigma(j_k) \ , \ \ i = j_k, ..., j_k + \tau_k \ , \tag{16}$$

i.e., the change in σ during stable phases, with the zero point reset each time at the beginning of each of the stable phases. The corresponding time series, plotted on Fig. 8, are extracted from simulation 3 and are now restricted to stable phases of duration $\tau \geq 50$ iterations to avoid overcrowding. A 10-iteration-wide boxcar filter was also applied to each time series, since the original series defined by eq. (16) turned out to be rather noisy.

At first glance there appears to be little trend in the evolution of the r.m.s. deviation σ in the course of stable phases; closer examination reveals, however, that significantly more time series end up below than above zero (112 versus 38 here). One can also compute an average beginning-to-end difference:

$$\langle \delta\sigma \rangle_{\rm S} = \frac{1}{K} \sum_{k=1}^{K} \Big(\sigma(j_k + \tau_k) - \sigma(j_k) \Big) \ , \tag{17}$$

which turns out to be systematically negative, but remains a small fraction of the mean r.m.s. deviation, in all first three baseline simulations (see Lines 25 and 26 in Table 4). In simulation 4 there are no stable phases of 50 iterations or more, we used the notation N/A for non applicability of the measures for this specific simulation. Turning crossover on or off at fixed mutation probability (simulations 1 vs 2) has little influence here. Interestingly, in these subsets of long stable phases the run including crossover (simulation 1) shows the highest averaged r.m.s. deviation, although it is unclear whether the difference, of about 10%, is statistically significant.

What is quite striking as well as significant is that even in stable phases lasting a few hundred iterations, population heterogeneity remains at a more or less constant level despite an expected takeover time of less than 10 iterations. The longest stable phase observed was in simulation 2 and lasted 410 iterations; yet in this instance the population-wide average deviation ended up slightly larger than at the beginning of that stable phase ($\langle \delta\sigma \rangle_k = +0.02$).

This general trend of more-or-less constant population-wide r.m.s. deviation also characterizes the other three baseline simulations, and thus represents a clear indication that sustained heterogeneity is an essential component of the evolutionary dynamics in these simulations. Support for this conclusion is found in simulations carried out with higher selection pressure ($\eta = 10$), which show even slightly *higher* levels of population heterogeneity, at least by the two measures used here.

Such persistent heterogeneity is remarkable in a number of ways. Recall that our population reproduction scheme begins by isolating the $\eta = 20$ best performing strate-

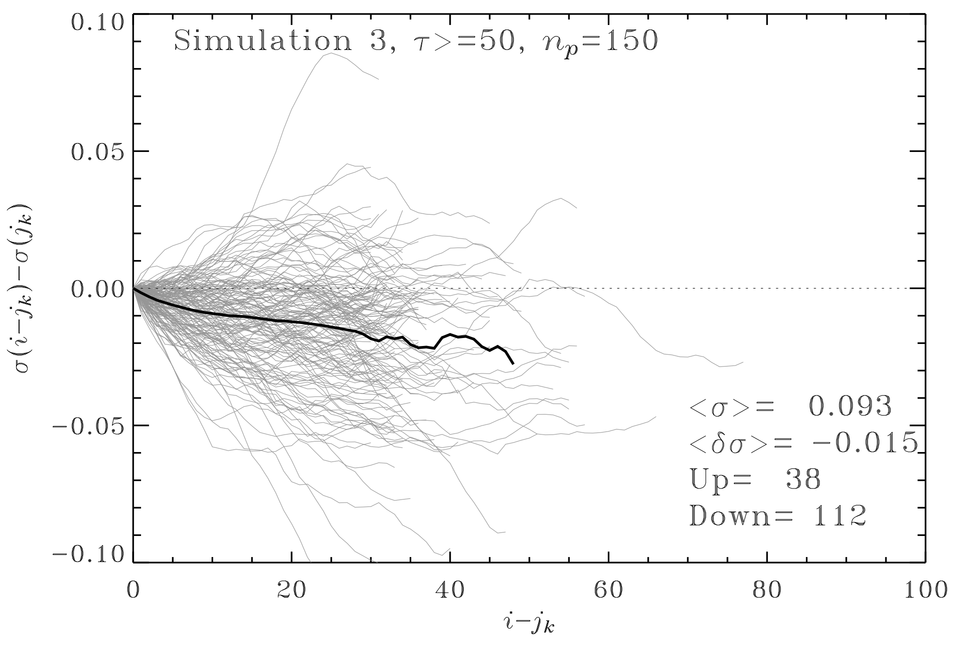

Figure 8: Time series of r.m.s. deviation changes during stable phases of duration $\tau \geq 50$ iterations in simulation 3. Each time series was smoothed via a 10-iteration-wide boxcar filter to reduce the noise to visually manageable levels. The thick line is the time series of average change out to 80 iterations, indicating a weak but systematic trend towards decrease in r.m.s. deviations. The mean r.m.s. deviation for this subset of 150 stable phases is 0.093, so that the average change from beginning to end of each plateau (eq. 17) represents a fairly small ($\sim 18\,\%$) decrease in population heterogeneity, even in this subset of very long stable phases.

gies at a given iteration (out of a population size $M = 100$), and breeds the next generation only from this subset. This represents very strong selection pressure. Because our GA operates at fairly low mutation probability, diversity can only be maintained if the "top-twenty" breeding subset is itself diversified, which in turn implies that multiple, distinct mutually cooperating strategies coexist through stable phases.

From an ecological point of view, the simulations can be regarded as equivalent to a complex "ecosystem" characterized by interacting "communities" of mutually cooperating strategies. In such a diversified, evolutionary (meta)stable population, targeted adaptive predators can perhaps decimate their prey strategy, but being prey-specific they have greater difficulty destabilizing the whole population. Even when they do, there remains a greater chance that useful building blocks are still passed on to subsequent generations through the end of the predatory phase, and may be available for crossover to recombine, and help "resurrect" former stable strategies. Heterogeneity thus enhances evolutionary stability in two ways: (1) It makes the population as a whole more resistant to targeted adaptive predators, and (2) it enhances the survival of useful building blocks during stable phases as well as across unstable epochs. The greater overall stability of simulations with crossover indicates that it can take advantage of this heterogeneity in a manner that (by all appearances) cannot be duplicated by one-point bit-flip mutations alone.

5 A conjecture: long-range correlations and evolutionary memory

Consider a hypothetical, simple and memoryless evolutionary procedure driven by a stationary random process. For example, one could construct a time series of numbers r_k extracted from a normal (Gaussian) distribution, and declare that the system remains stable until r_k exceeds some externally-set threshold. Under these conditions, it is readily shown that the resulting PDF of stable phase lifetimes is exponential, i.e., $f(\tau) \propto \exp(-\tau/\tau_0)$. This is the distribution plotted as a dotted line back on Fig. 3, with τ_0 adjusted to fit the first three bins of the reference simulation.

The PDFs of stable phase lifetimes plotted on Fig. 3 are clearly non-exponential, and, in particular, exhibit a "fat tail" at large lifetimes. This is indicative of some "memory", or, more accurately, temporal correlations, in the evolutionary process. This is in fact not surprising at all, in view of the descent-with-modification nature of the GA: successive generations are correlated, and here we can actually expect the correlation to be quite high given the high selection pressure embodied in our tournament selection procedure.

These correlations are slowly but inexorably destroyed by mutation. Referring back to Fig. 3, we notice how the tails of the PDFs for simulations 1 and 2 (mutation probability $p_m = 0.0159$) fall more slowly and extends farther than for the PDF of simulation 3 ($p_m = 0.0345$). Crossover, on the other hand, redistributes building blocks and therefore does not destroy temporal correlations *in the population as a whole*, at least as far as building blocks of short defining lengths are concerned. Evidence for this claim can be found in the essentially identical tails of simulations 1 and 2, both having the same mutation probability but crossover turned off in the second. Yet simulation 1 has a markedly higher probability of disruption at breeding. While individual population members differ more on average from their parents in simulation 1 than in 2, at the level of building blocks (or, loosely speaking, the "gene pool"), both simulations collectively show the same collective average change in the population from one generational iteration to the next.

On this basis, we conjecture that crossover is sustaining genotypic correlations in the evolving population more efficiently than mutation, while allowing comparable adaptive evolutionary change from one generation to the next. In a static evolutionary environment (such as function optimization), this would amount to lesser diversity, and might well prove deleterious. In the evolutionarily dynamical environment offered by the IPD, such correlations are advantageous, as they allow the efficient re-assembly of metastable strategies from surviving building blocks scattered in the population following destabilization by adaptively targeted predators.

6 Concluding remarks

The conjecture put forth in this paper, on the basis of an evolutionary study of the Iterated Prisoner's Dilemma, is that crossover recombination sustains long-range temporal correlations in an evolving population more efficiently than purely mutation-driven adaptive descent-with-modification, which then leads to overall greater evolutionary stability.

Long timescale evolutionary patterns qualitatively resembling "punctuated equilibrium" arise in the simulations as the population rapidly evolves from one of a few possible metastable strategies to another, in response to the appearance of adaptively targeted predator strategies. Crossover plays an important role in these transient phases, where recombination of building blocks apparently favors an efficient search for a new metastable strategy while predators drive themselves to extinction: In the absence of crossover, unstable phases last 50% and 700% longer than when crossover is included (for $p_c = 0.7$ and at equal breeding disruption probability and empirical disruption probability respectively). This is particularly remarkable in view of the very high selection pressure characterizing our adopted tournament selection algorithm, characterized by a takeover time of only 6 generational iterations. Crossover is not just assembling building blocks; it is evidently finding and assembling them very efficiently.

Persistent population heterogeneity, in the form of coexisting families of cooperating strategies is a vital factor in impeding the spread of adaptively targeted predators throughout stable phases. This makes it more difficult for an adaptive predator to wipe out the whole population through the use of an exploitive predatory strategy targeted to a specific prey-strategy (such as ATFT exploiting TFT's "start off nice" feature). The simulations discussed here indicate that crossover-mediated recombination can exploit this heterogeneity and also ensures that building blocks defining mutually cooperating strategies survive longer in the population, making it more likely that such strategies are rapidly rediscovered after evolutionary stability has been lost. We have argued that this survival of building blocks dispersed in the population is the origin of the long temporal correlations evidenced by the extended tails in the PDFs of stable phase lifetimes.

Whether these results can be generalized to dynamical evolutionary environments other than the IPD remains to be demonstrated. Recent studies of the geological fossil record have revealed extended tails —often in the form of more-or-less convincing power laws— in the lifetimes of various species and taxa (see, e.g., Sepkoski (1993), and references therein). This has sometimes been taken as evidence that biological evolution may be a self-organized critical phenomenon (Solé and Manrubia, 1996; Jan et al., 1999), although other non-critical, simple and plausible power-law generating schemes have also been proposed in the evolutionary context (Newman, 1996). The simulation reported herein illustrates yet another means of producing extended tails in lifetime distributions, through the long-range correlations sustained by crossover.

Acknowledgment

We wish to thank Tim Brown, Thierry Emonet, and Lael Parrott for useful discussions, and anonymous referees for constructive criticism and valuable suggestions on the former version of this paper. This work was supported by the Natural Sciences and Engineering Research Council, the Canadian Foundation for Innovation, and the Canada Research Chair program.

References

Axelrod R. and Hamilton W. D. (1981). The evolution of cooperation. *Science*, 211:1390–1396.

Axelrod R. (1987). The Evolution of Strategies in the Iterated Prisoner's Dilemma, In Davis L., editor, *Genetic Algorithms and Simulated Annealing*, pages 32–41, Morgan Kaufmann, Los Altos, California.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, New York.

Boyd, R. and Lorberbaum, J. P. (1987). No pure strategy is evolutionarily stable in the repeated prisoner's dilemma game. *Nature*, 327:58–59.

Boyd, R. (1989). mistakes Allow Evolutionary Stability in the Repeated Prisoner's Dilemma Game. *J. Theor. Biol.*, 327:58–59.

Brembs, B. (1996). Chaos, cheating and cooperation: potential solutions to the Prisoner's Dilemma. *OIKOS*, 76:14–24.

Chong, S. Y. and Yao, X. (2005). Behavioral Diversity, Choices, and Noise in the Iterated Prisoner's Dilemma. *IEEE Transactions on Evolutionary Computation*, 9:540–551

Culberson, J. C. (1994). Mutation-crossover isomorphisms and the construction of discriminating functions. *Evol. Comp.*, 2:279–311.

Darwen, P. and Yao, X. (1995). On evolving robust strategies for iterated prisoner's dilemma, In Progress in Evolutionary Computation, Lecture Notes in Artificial Intelligence, Vol. 956, Springer-Verlag, Heidelberg, Germany, pp.276–292.

Darwen, P. and Yao, X. (2002). Co-Evolution in Iterated Prisoner's Dilemma with Intermediate Levels of Cooperation: Application to Missile Defense. *International Journal of Computational Intelligence and Applications*, 2:83–107.

Fogel, D. B. and Atmar, J. W. (1990). Comparing genetic operators with Gaussian mutation in simulated evolutionary processes using linear systems. *Biol. Cybern.*, 63:111–114.

Fogel, D. B. (1993). Evolving behaviors in the iterated prisoner's dilemma. *Evol. Comp.*, 1:77–97.

Gould, S. J. and Eldredge, N. (1993). Punctuated equilibrium comes of age. *Nature*, 366:223–227.

Holland, J. H. (1992). *Adaptation in natural and Artificial Systems*, MIT Press, Cambridge, Massachusetts.

Holland, J. H. (2000). Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions. *Evol. Comp.*, 8:373–391.

Ishibuchi, H. and Namikawa, N. (2005) Evolution of Iterated Prisoner's Dilemma game strategies in structured demes under random pairing in game playing. *IEEE Transactions on Evolutionary Computation*, 9:552–561.

Jan, N., Moseley, L., Ray T., and Stauffer, D. (1999). Is the fossil record indicative of a critical system? *Adv. Complex Syst.*, 2:137–141.

Jones, T. (1995). Crossover, macromutation, and population-based search. In Eshelman, L. J., editor, *Proceedings of the 6th international conference on genetic algorithms*, pages 73–80 Morgan Kaufmann, California, United States.

Law, A. M. and Kelton, W. D. (2000). *Simulation Modeling and Analysis, 3rd edition*, McGraw-Hill, New York, United States.

Lindgren, K. (1991). Evolutionary Phenomena in Simple Dynamics. In Langton, C. G., Taylor, C., Farmer, J. D. and Rasmussen, S., editor, *Artificial Life II*, pages 295–312, Addison-Wesley, New Mexico, United States.

Newman, M. E. J. (1996). Evidence for self-organized criticality in evolution *Physica D*, 107:293–296.

Nowak, M. A. and Sigmund, K. (1992). Tit for tat in heterogeneous populations. *Nature*, 355:250–253.

Nowak, M. A. and Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364:56–58.

Sepkoski, J. J. (1993). Ten years in the library: new data confirm paleontological patterns. *Paleobiology*, 19:43–51.

Solé, R. V. and Manrubia, S. C. (1996). Extinction and self-organized criticality in a model of large-scale evolution. *Phys. Rev. E*, 54:R42–R45.

Spears, W. M. (1993). Crossover or Mutation?. In Whitley L. D., editor, *Foundations of Genetic Algorithms 2*, pages 221–237, Morgan Kaufmann, San Mateo, California.