Available online at www.sciencedirect.com
SCIENCE DIRECT®

Theoretical Computer Science 358 (2006) 121 – 141

Theoretical Computer Science

www.elsevier.com/locate/tcs

Phase transitions and symmetry breaking in genetic algorithms with crossover

Alex Rogers*, Adam Prügel-Bennett, Nicholas R. Jennings

School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK

Received 1 September 2004; received in revised form 28 October 2005; accepted 24 April 2006

Communicated by C. Reeves

Abstract

In this paper, we consider the role of the crossover operator in genetic algorithms. Specifically, we study optimisation problems that exhibit many local optima and consider how crossover affects the rate at which the population breaks the symmetry of the problem. As an example of such a problem, we consider the subset sum problem. In doing so, we demonstrate a previously unobserved phenomenon, whereby the genetic algorithm with crossover exhibits a critical mutation rate, at which its performance sharply diverges from that of the genetic algorithm without crossover. At this critical mutation rate, the genetic algorithm with crossover exhibits a rapid increase in population diversity. We calculate the details of this phenomenon on a simple instance of the subset sum problem and show that it is a classic phase transition between ordered and disordered populations. Finally, we show that this critical mutation rate corresponds to the transition between the genetic algorithm accelerating or preventing symmetry breaking and that the critical mutation rate represents an optimum in terms of the balance of exploration and exploitation within the algorithm.

Keywords: Symmetry breaking; Phase transition; Crossover; Genetic algorithms

1. Introduction

Despite the popularity of the genetic algorithm as a generic stochastic search algorithm for real-world optimisation problems, the role that crossover plays in these algorithms is still poorly understood. This lack of understanding, means that there is little guidance as to when the genetic algorithm with crossover should be used to good effect, nor how the numerous parameters of the algorithm should be set.

However, recent work has suggested that a critical factor in the performance of population-based optimisation algorithms, is how they break the symmetry of the optimisation problem; hard optimisation problems typically exhibit a search space divided into many regions containing local optima and symmetry breaking is said to occur when the algorithm focuses prematurely on a particular region of that search space [3,6]. This symmetry breaking is typically a stochastic effect due to the finite size of the population and the result is that the algorithm fails to effectively search the entire problem space.

* Corresponding author. Tel.: +44 023 8059 3255; fax: +44 023 8059 2865.
E-mail addresses: acr@ecs.soton.ac.uk (A. Rogers), apb@ecs.soton.ac.uk (A. Prügel-Bennett), nrj@ecs.soton.ac.uk (N.R. Jennings).

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.04.010

This analysis raises some key questions regarding the crossover operator used in the genetic algorithm. Whilst solutions from different regions of the search space may exhibit similar fitness, their chromosomes or genotypes may be very different. In the genetic algorithm without crossover, these different genotypes can happily co-exist within the population. However, the crossover operator acts to combine population members together and thus, when it combines two solutions from different regions of the search space, the offspring produced may be very different from either parent. The effect on the algorithm appears to depend on the fate of these offspring. If the offspring typically have very poor fitness, they will be removed from the population by selection. This creates an indirect selection pressure where it is preferential to have a genotype similar to that of other individuals within the population, and, in turn, symmetry breaking is accelerated. Conversely, if the offspring have similar fitness values to their parents, they are maintained within the population and the diversity of the population prevents symmetry breaking. Thus, in order to understand the role of crossover in the genetic algorithm, it is necessary to understand the effect that crossover has in accelerating or preventing this symmetry breaking.

Thus, as a first step toward this goal, we consider the dynamics of a genetic algorithm solving the subset sum problem. We choose this problem as it presents exactly the type of symmetry described above and rather than being a toy problem, is a real NP-hard optimisation problem [2]. It also has the advantage that it has been the subject of theoretical analysis and it has been shown to be possible to analytically describe the dynamics of a genetic algorithm solving this problem [9]. The results of the analysis are such that we can generalise from this example to other problem instances.

Against this background, we initially demonstrate the novel result that the genetic algorithm with crossover exhibits a critical mutation rate at which the equilibrium state of the algorithm sharply diverges from that of the genetic algorithm without crossover and there is a rapid increase in population diversity. This critical mutation rate is different from the error threshold observed in Eigen's quasi-species model of evolution [1], since it is only apparent within the genetic algorithm with crossover, and whilst the diversity of the population changes, its mean behaviour is unaffected. We show this with experimental results from a standard genetic algorithm subject to single point crossover, mutation and tournament selection solving a real instance of the subset sum problem.

In order to calculate the details of this phenomenon, we consider a simpler instance of the subset sum problem and solve the dynamics of a genetic algorithm on this problem. We show that in the limit of an infinite population and complete crossover, the observed critical mutation rate is a classic thermodynamic phase transition between ordered and disordered populations. We show that this phase transition corresponds to the transition point between the crossover operator either accelerating or preventing spontaneous symmetry breaking within the population of the genetic algorithm.

Finally, we return to the real instance of the subset sum problem to consider the wider implications of these results. We show that in the subset sum problem, this critical mutation rate represents an optimum mutation rate. At this mutation rate, exploration and exploitation forces are in balance and the algorithm effectively searches the entire search space. The novel result gives a key insight into the setting of the various parameters of the genetic algorithm.

The remainder of this paper is organised as follows: Section 2 presents some related work in this area. Section 3 introduces the subset sum problem and describes the experimental results for a genetic algorithm attempting to solve this problem. In Section 4 we solve the dynamics of the genetic algorithm on the subset sum problem and show that the critical mutation rate is a phase transition between ordered and disordered states. Finally in Section 5 we discuss the broader implications of this result and conclude.

2. Related work

Much of the progress in understanding the role of the various operators and parameters of the genetic algorithm has stemmed from work modelling the dynamics of the genetic algorithm on a number of simple problem instances. Whilst the problems investigated by different researchers have generally been similar (e.g. onemax), there are, in general, two approaches to actually perform the modelling. One approach, pioneered by Vose and collaborators, uses a microscopic level description of the population, and models the evolution of the population as a Markov process [14,4]. An alternative approach, models the population with a small number of macroscopic variables and uses techniques from statistical physics to model the evolution of these variables when the population is subject to selection, mutation and crossover [7,12]. In this paper, we adopt the later approach, as by making two key simplifying assumptions to both the problem under consideration and the genetic algorithm used, we are able to solve the dynamics of the genetic algorithm using a single macroscopic variable—the correlation of the population. The ability to actually solve the model, rather than simply numerically calculate the dynamics of the genetic algorithm, allows us to understand the origin of the critical mutation rate that we observe in the experiments.

The existence of this critical mutation rate was first published by one of the authors in the context of biological models of evolution and its implications for the role of sexual reproduction within these models has been explored [10]. Although this specific phenomenon is unknown in the genetic algorithm community, superficially, it appears to be similar to the more common concept of an *error threshold*, introduced in the quasi-species models of evolution of Eigen [1]. These models involve populations evolving subject to just selection and mutation. Specifically, they consider a very specialised problem instance in which there is a single genotype, the ‘master sequence’, which has a fitness advantage over all other genotypes. The error threshold in the Eigen model results from the observation that whilst selection is preferentially reproducing the ‘master sequence’, mutation is acting to disrupt it.

In the computer science literature, Eigen’s model is often called the ‘needle-in-a-haystack’ problem and it has been the subject of some investigation. Typically this work has extended the model and introduced finite populations and crossover [5,16,15]. Whilst these results are interesting and important, the ‘needle-in-a-haystack’ problem is not really representative of real optimisation problems. Critically, there is no gradient information to guide an optimisation algorithm, and thus the error threshold is simply due to mutation disrupting previously found optimal genotypes faster than selection can reproduce them. As such, it provides little insight into the role of the crossover operator in real-world problems. The mechanism by which the phase transition and critical mutation rate arise in our problem instance is very different. It is a general phenomenon arising from the symmetry of the problem space and thus has much wider application for understanding real genetic algorithms and real optimisation problems.

3. The subset sum problem

The subset sum problem posed as an optimisation problem consists of finding a subset from a set of integer numbers whose sum most closely matches a target value. If we consider a set of L integer numbers, $J_1 \ldots J_L$, and we seek to find a subset of these numbers whose sum most closely matches the target, T, the problem is simply described as

$$\arg\min_{X_i} \left| \sum_{i=1}^{L} J_i X_i - T \right| \quad \text{where } X_i \in \{0, 1\}. \tag{1}$$

We can attempt to solve this using a standard genetic algorithm, by considering a population of P potential solutions. Each solution, α, has a chromosome or genotype, consisting of a binary string of L bits, $X_1^\alpha \ldots X_L^\alpha$ and the sum of this solution is given by

$$S^\alpha = \sum_{i=1}^{L} J_i X_i^\alpha. \tag{2}$$

A single generation of the genetic algorithm consists of applying tournament selection, mutation and single point crossover. We initially select a new population of size P, by repeatedly randomly drawing two solutions from the original population, and selecting the one whose sum is closest to the target, T. Following selection, we apply mutation individually to each bit within this new population

$$X_i^\alpha \to 1 - X_i^\alpha \quad \text{with probability } \mu \tag{3}$$

and finally, we perform single point crossover between pairs of solutions within the new population. The new population completely replaces the original one.

Now, if we start with an initial random population and allow the genetic algorithm to evolve for a number of generations, we find that whilst the individual population members are continually subject to mutation, selection and crossover, the population as a whole rapidly reaches an equilibrium state. At this equilibrium, the macroscopic properties of the population remain fixed (aside from small temporal fluctuations due to the stochastic nature of the algorithm). The macroscopic properties which we specifically study in this paper are the mean (K_1) and variance (K_2) of sums within the population

$$K_1 = \frac{1}{P}\sum_{\alpha=i}^{P} S^{\alpha}, \tag{4}$$

$$K_2 = \frac{1}{P}\sum_{\alpha=i}^{P} (S^{\alpha})^2 - \left(\frac{1}{P}\sum_{\alpha=i}^{P} S^{\alpha}\right)^2 \tag{5}$$

and the correlation (q) of the population—a measure of the similarity of individual genotypes within the population

$$q = \frac{1}{P(P-1)}\sum_{\alpha\neq\beta}\frac{1}{L}\sum_{i=1}^{L}(2X_i^{\alpha}-1)(2X_i^{\beta}-1). \tag{6}$$

The correlation of a completely random population is zero, whilst a population of identical genotypes has a correlation of one.

In Fig. 1 we show experimental results for this equilibrium, for the genetic algorithm with and without crossover. In this case, we consider a set of 128 natural numbers (i.e. $L = 128$) whose values are drawn uniformly within the range of 0–100 (i.e. $J_i \in [0, 100]$). The value of the target sum, T, is 4800 and this value represents a problem where on average approximately $\frac{3}{4}$ of the bits within the optimal genotype will be a one (the mean value of the numbers within the set is 50 and thus $128 * 50 * \frac{3}{4} = 4800$). Given a fixed mutation rate, the genetic algorithm was run 100 times using a different randomly generated problem instance, and after 1000 generations (when the population had reached equilibrium) the mean, K_1, and variance, K_2, of the sums within the population and the correlation, q, of the genotypes within the population were measured. This process was repeated for different mutation rates in the range of 0 to $2/L$.

If we leave aside crossover, we can understand this equilibrium through the interaction of mutation and selection. Selection favourably reproduces solutions that are closest to the target sum and thus moves the population toward the target sum, increasing the mean and decreasing the variance of the population. In contrast, mutation acts to move the population away from the target sum, toward the original random state of the population, and thus decreases the mean and increases the variance of population. The equilibrium state is reached where these two opposing effects balance.

For the genetic algorithm without crossover, the correlation of the population does not directly affect the dynamics of the population. However, we can understand the resulting equilibrium correlation through the above explanation. When mutation rates are low, the equilibrium correlation is close to one, corresponding to a state in which all the genotypes within the population are similar. As the mutation rate increases, the increasing variance implies that there is more diversity in the population, and the correlation gradually decreases.

Clearly, for small mutation rates, the genetic algorithm with crossover and the genetic algorithm without crossover, reach the same equilibrium and the intuitive understanding presented above explains the equilibrium. However, the genetic algorithm with crossover exhibits a critical mutation rate where the behaviour of the two algorithms sharply diverges. The algorithm shows a rapid increase in the equilibrium variance of the population and a corresponding decrease in equilibrium correlation of the population.

This sudden divergence in the behaviour of the genetic algorithm with and without crossover has not been previously reported. It is clearly different from the standard error threshold since its effect is only seen within the variance and correlation of the population, whilst the mean is unaffected. Also, unlike the standard error threshold, it is dependent on the presence of symmetry within the problem. This symmetry arises, since given any particular initial set of numbers, there are many different subsets (and thus different genotypes) that sum to the same value, and thus have the same fitness. We can demonstrate the effect that this symmetry has by increasing the value of the target sum to its maximum possible value. At this point, the optimum genotype consists of a one in each position (i.e. $X_i = 1$), and clearly there is now only one form of this genotype. In addition, the symmetry of solutions close to the optimum is greatly reduced, since nearly all the bits within the genotype must also be set to one. In Fig. 2 we show experimental results for this problem, using exactly the same procedure as before. Clearly, in this case, there is no critical mutation rate at which the equilibrium correlation and variance of the genetic algorithm with crossover sharply diverges from the genetic algorithm without crossover.

Thus, the transition that we observe in the subset sum problem is dependent on two factors: (i) the presence of a crossover operator that combines genotypes within the population together, and (ii) the existance of symmetry within the problem space by which individuals with the same fitness may have different genotypes. Our goal, in this paper,

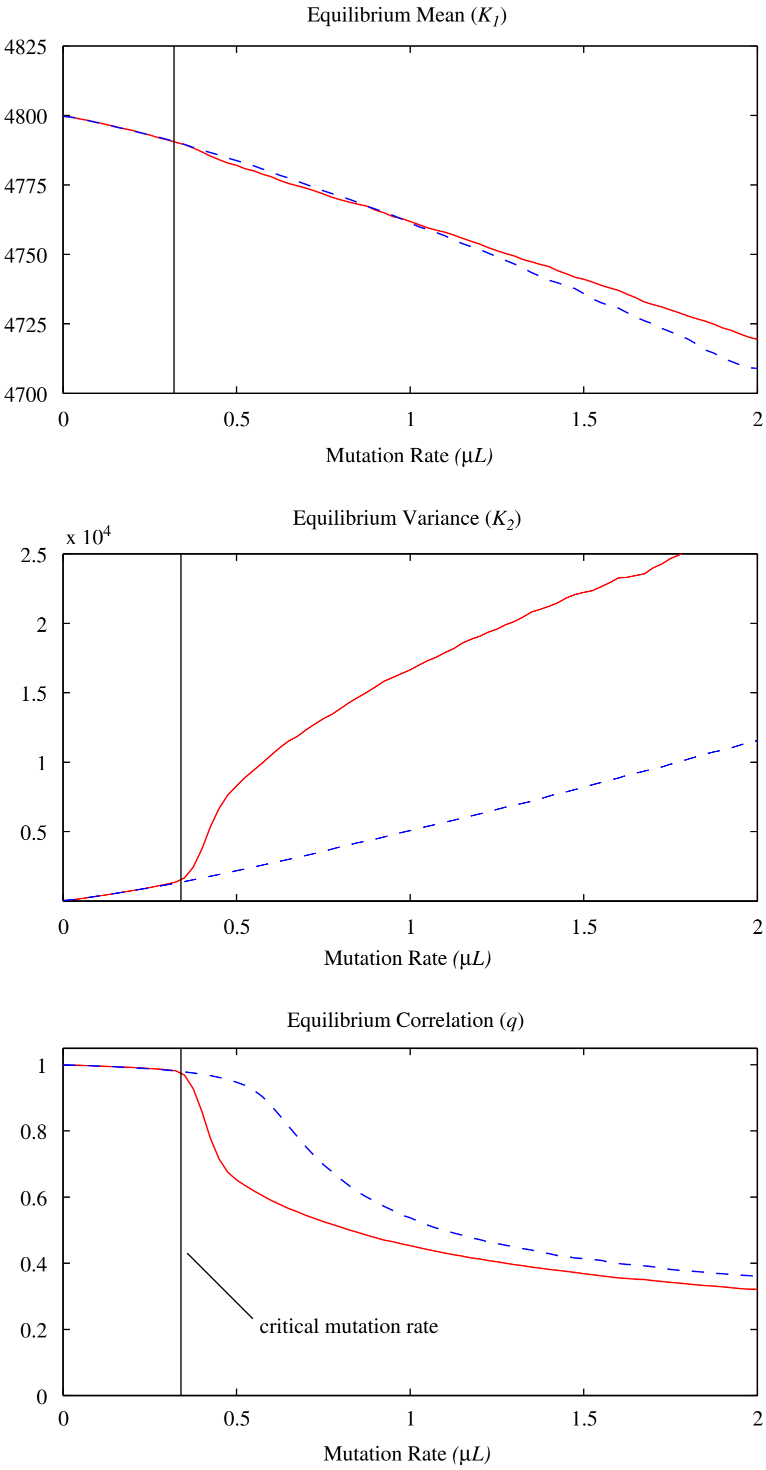

Fig. 1. Experimental results for the mean, variance and correlation of the equilibrium population for a genetic algorithm with (*solid line*) and without (*dashed line*) single point crossover using tournament selection to solve the subset sum problem. Population size $P = 100$, string length $L = 128$, the individual values $J_i \in [0, 100]$ and the target sum $T = 4800$. The measurements are made after 1000 generations and the results are averaged over 100 runs, each with a different randomly generated problem instance.

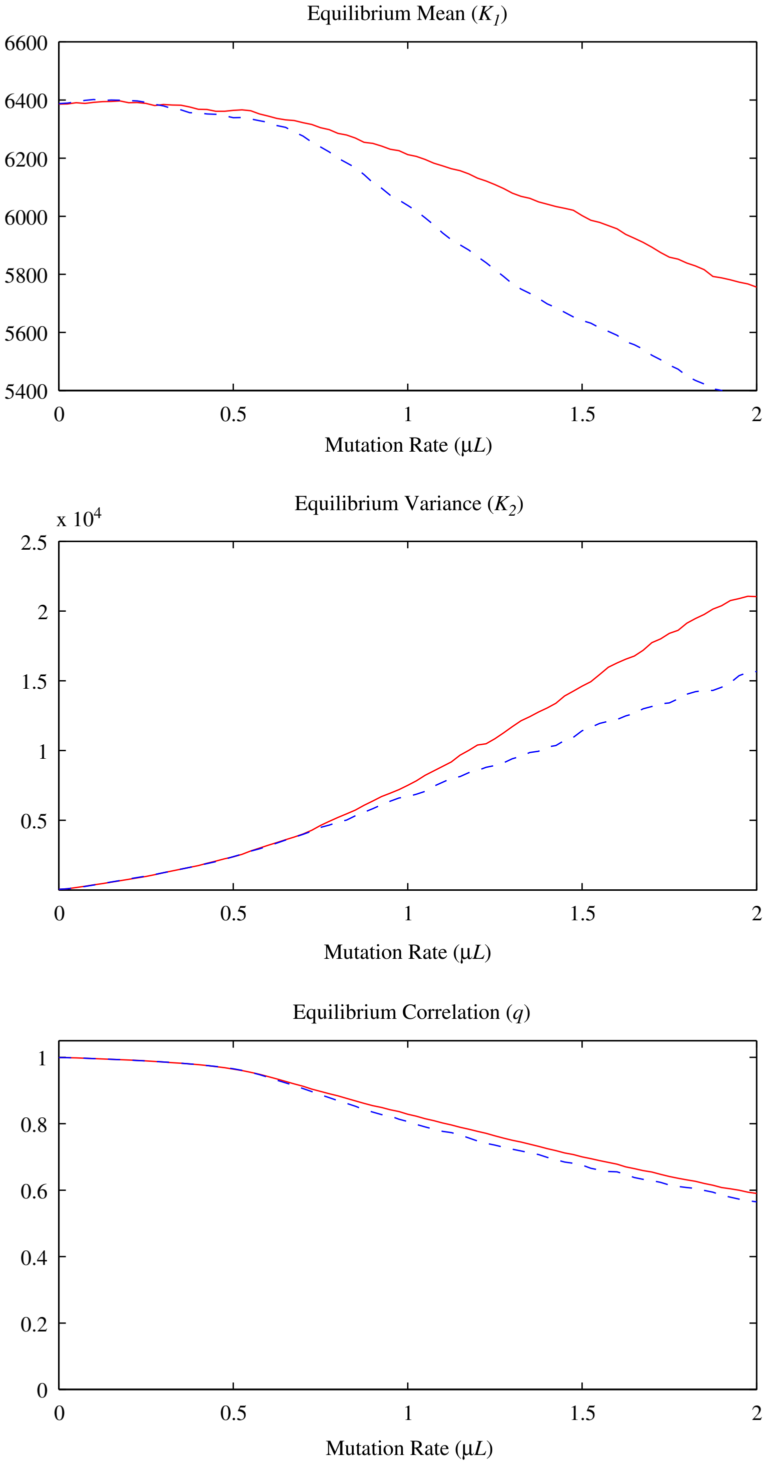

Fig. 2. Experimental results for the mean, variance and correlation of the equilibrium population for a genetic algorithm with (*solid line*) and without (*dashed line*) single point crossover using tournament selection to solve the subset sum problem where the target sum is set to its maximum possible value. Population size $P = 100$, string length $L = 128$ and the individual values $J_i \in [0, 100]$. The measurements are made after 1000 generations and the results are averaged over 100 runs, each with a different randomly generated problem instance.

is to fully understand the causes behind this sudden transition, and also to understand the implications that this has on the way in which the genetic algorithm is searching the problem space.

4. Solving the dynamics

Prügel-Bennett, Shapiro, Rattray and Rogers have modelled the dynamics of genetic algorithms on a range of simple problem instances and we adopt this formalism here. They have shown that the dynamics of the genetic algorithm can be well described by a few macroscopic properties of the population [7,12,8,11]. Specifically, for a genetic algorithm without crossover, the first four cumulants of the population distribution, $K_1 \ldots K_4$, are sufficient to describe the dynamics to a high degree of accuracy. As the effect of the crossover operator is dependent on the similarity of the genotypes being crossed, the dynamics of the genetic algorithm with crossover can also be modelled in this approach by adopting the correlation of the population as an additional macroscopic variable [9].

Whilst this approach numerically predicts the dynamics of the genetic algorithm to a high degree of accuracy, in general it results in a large number of coupled equations which cannot be solved analytically. In this work we go one step further, and by making a key simplifying assumption to both the subset sum problem instance and the genetic algorithm, we derive a single parameter model of the dynamics of the genetic algorithm with crossover. This model uses just the correlation of the population as the order parameter. It allows us to solve the dynamics of the equilibrium state and understand the origin of the critical mutation rate observed in the real-world example shown in Section 3.

Thus, to this end, we consider a simple instance of the subset sum problem, where all $J_i = 1$ and the target sum $T = L/2$. We also consider a genetic algorithm with a large population and a special crossover operator whereby bits are swapped among the entire population, rather than among pairs as is commonly the case in uniform or single point crossover. Such a crossover operator can be implemented directly or through repeated application of a pair-wise crossover operator [13].

Neither assumption changes the character of the genetic algorithm or the problem, however they significantly simplify the analysis. The first assumption ensures that both mutation and selection drive the population toward the target sum and we are not required to explicitly model the mean of the population, K_1. The second assumption means that after applying the crossover operator, there is complete independence between each bit position within the genotype. This independence is known in the biology literature as linkage equilibrium, and allows us to describe the population by a set of L independent probabilities, p_i, each describing the probability of having a one, rather than a zero, at the ith position in the genotype [17]. The definition of correlation given in Eq. (4) can thus be re-expressed as

$$q = \frac{1}{L}\sum_{i=1}^{L} (2p_i - 1)^2. \tag{7}$$

The variance of the population, K_2, can also be expressed in terms of these probabilities

$$K_2 = \sum_{i=1}^{L} p_i(1 - p_i) \tag{8}$$

and thus the variance of the population, after complete crossover, can be described directly in terms of the correlation of the population

$$K_2 = \frac{L}{4}(1 - q). \tag{9}$$

Thus having applied complete crossover, the state of the population is completely described by the correlation alone. As crossover simply mixes the existing population, it does not itself change the correlation of the population, and thus the dynamics of the genetic algorithm are solely dependent on how correlation is affected by selection and mutation.

4.1. Mutation

Calculating the change in correlation due to mutation is relatively straightforward. Mutation introduces new diversity into the population and will thus decrease the correlation. If the probability of mutation is μ, the probability of having a one, rather than a zero, at the ith position after mutation, p_i', is given by

$$p_i' = \mu + (1 - 2\mu)p_i. \tag{10}$$

Thus, using this expression in Eq. (7), gives the correlation after mutation, q', as simply

$$q' = \Gamma^2 q \quad \text{where } \Gamma = 1 - 2\mu. \tag{11}$$

4.2. Selection

Calculating the effect of selection is more complex than mutation. Selection acts on the fitnesses of individuals, and thus, whilst the correlation of the population is well defined, in general we do not know the actual values of p_i (nor the distribution of sums) within the population. However, we can recover this information through maximum entropy inference, and thus, we assume that all population states are equally likely and we then find the population state that occupies the majority of the space of possible populations. In doing so, we find the most likely state of the population, and we note that due to the nature of the problem which we consider here (and specifically the binomial function in the expression for entropy in Eq. (12)), as the population size increases, the probability of finding the population in any of the other states becomes vanishingly small.

4.2.1. Maximum entropy inference

Now, the entropy, S, of any given population state is simply the logarithm of the number of possible ways in which the bits within the population can be arranged and thus

$$S = \ln \prod_{i=1}^{L} \binom{P}{p_i P}. \tag{12}$$

As described above, the most likely state in which we find the population is that which occupies the majority of the space of possible populations and thus has the maximum entropy. As an example, maximising this expression with no constraints gives the result $p_i = \frac{1}{2}$—as expected, a completely disordered population. However, in our case, the values of p_i are constrained by both the mean and the correlation of the population. Thus, by rearranging the expression for correlation in Eq. (7), we have the first constraint

$$\sum_{i=1}^{L} p_i^2 = \frac{L(1+q)}{4} \tag{13}$$

and by assuming that the mean of the population must equal the target value at equilibrium, we have the second constraint

$$\sum_{i=1}^{L} p_i = L/2. \tag{14}$$

Now, maximising the entropy expression, given in Eq. (12), over these constraints yields the expression

$$p_i = \frac{1 \pm \sqrt{q}}{2}, \tag{15}$$

where p_i forms two equal size classes; one with probability $(1 - \sqrt{q})/2$ and one with probability $(1 + \sqrt{q})/2$. A proof of this result is shown in Appendix A.

This is a critical result. In an uncorrelated population where $q = 0$, the population is completely disordered and the probability of having a one at any position within the genotype is $\frac{1}{2}$ (i.e. $p_i = \frac{1}{2}$). As correlation increases, the genes within the genotype separate into two equal-sized classes. In one class the probability of a one is $(1 + \sqrt{q})/2$ and in the other class the probability of a one is $(1 - \sqrt{q})/2$. Which genes belong to which class will depend on the initial conditions and on chance fluctuations caused by the stochastic operators. There is thus a spontaneous symmetry breaking where the genes are assigned to one or other class. If we were to run the genetic algorithm a second time, we are likely to end up with a totally different symmetry breaking.

This symmetry breaking has consequences for which genotypes occur within the population. In the uncorrelated population, all the target genotypes (i.e. those genotypes which sum to the target sum) are equally represented within the population. However, when spontaneous symmetry breaking occurs, one form of genotype is arbitrarily selected, and begins to dominate the other forms. In the completely correlated population where $q = 1$, the population consists solely of P identical copies of this particular genotype. Again, if we were to run the genetic algorithm a second time, we are likely to end up with a totally different final population.

4.2.2. Calculating the effect of selection

Having solved for the values of p_i within the population, the actual distribution of sums can be found, and subsequently the effect of selection calculated. Now, since in our simple instance of a subset sum problem all $J_i = 1$, we do not need to consider the actual value of the sums within the population, but need only consider the probability of a population member having n ones within its genotype. This we denote as $P(S = n)$.

Now, selection simply changes the prevalence within the population of individuals with each of the $L + 1$ possible values of n. Thus the effect of selection is simply to apply a weighting, $W(S = n)$, such that the value of $P(S = n)$ after selection, $P'(S = n)$, is simply

$$P'(S = n) = W(S = n)P(S = n). \tag{16}$$

However, what we really need to calculate is the value of p_i after selection, p'_i. To do so, we must consider the probability that any particular individual with n ones in its genotype also has a one at the ith position, $P(S = n \;\&\; X_i = 1)$, and simply sum this expression over all values of n, weighted by the appropriate weighting, $W(S = n)$, to give

$$p'_i = \sum_{n=0}^{L} W(S = n)P(S = n \;\&\; X_i = 1). \tag{17}$$

Now, the values of $P(S = n)$, $W(S = n)$ and $P(S = n \;\&\; X_i = 1)$ can be calculated given our knowledge of p_i derived from the maximum entropy inference (i.e. that it forms two equal size classes; one with probability $(1 - \sqrt{q})/2$ and one with probability $(1 + \sqrt{q})/2$). In Appendix B we present the full details of performing these calculations. Finally, having calculated, p'_i, we can simply use Eq. (7), to calculate the correlation after selection

$$q' = (2p'_i - 1)^2. \tag{18}$$

Thus, to summarise, given the initial correlation of the population, we are able to calculate the values of p_i within the population using maximum entropy inference. Using this result, we can calculate the distribution of sums within the population and thus calculate the value of p_i after selection. Finally, we can use this result to derive the correlation after selection. In general, this process results in complex expressions that we solve numerically, however, the case when $L = 2$ is special and results in a simple analytical expression (see Appendix C for more details).

4.3. Equilibrium

Now, having calculated the effect of both selection and mutation on the correlation of the population, the equilibrium state is found when the increase in correlation due to selection is balanced by the decrease in correlation due to mutation (thus leaving correlation unchanged after both operators). Solving this equilibrium graphically clearly illustrates the existence of a classic phase transition between ordered and disordered states. In particular, Fig. 3 shows the correlation after both selection and mutation (calculated as described here), plotted against the initial correlation, at four different mutation rates. For small mutation rates, there is clearly a stable equilibrium solution near to $q = 1$. However, as the mutation rate increases, a critical value is exceeded where there are no equilibrium solutions other than $q = 0$, the completely disordered state (i.e. the correlation curve does not intersect the line $x = y$ other than at the point $x = 0$).

Fig. 4 shows this equilibrium correlation for a range of values of L from 2 to 32. Note that, in general, since the curves that describe the correlation after both selection and mutation (shown in Fig. 3) are not convex, the correlation is discontinuous at the critical mutation rate (i.e. a first order phase transition). In addition, note that the position of the critical mutation rate varies linearly with L (i.e. it is fixed at $0.32 \times L$). The exception to this is the case when $L = 2$. As discussed earlier, the expression for the correlation after selection has a simple analytical form and results in a convex curve (see Appendix C for full details of this expression and the calculation of the equilibrium correlation). Thus, in the

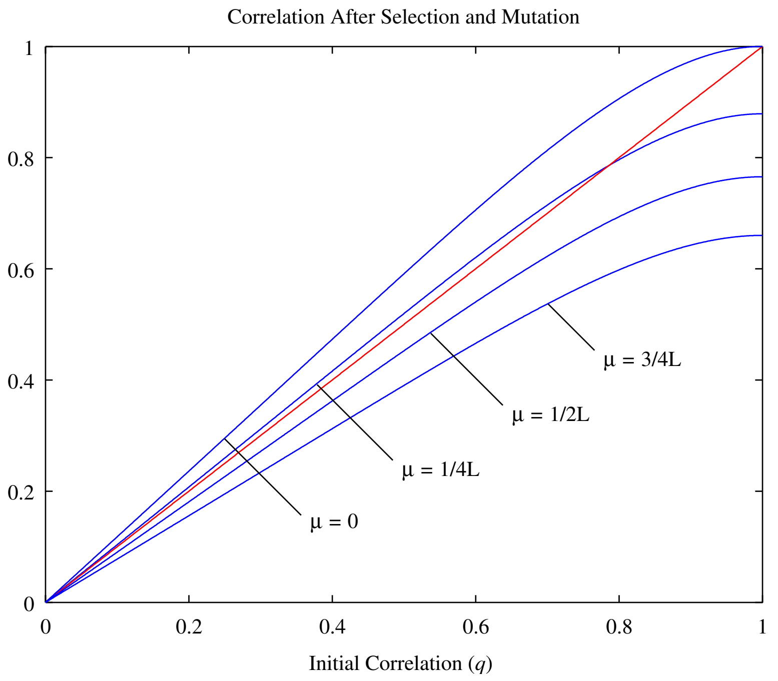

Fig. 3. Calculated results showing the graphical solution for equilibrium correlation at four different mutation rates $\gamma = 0$, $1/4L$, $1/2L$ and $3/4L$. For this example, we choose a small variable problem ($L = 8$) as the curvature of the curves is more pronounced, making the figure clearer to see.

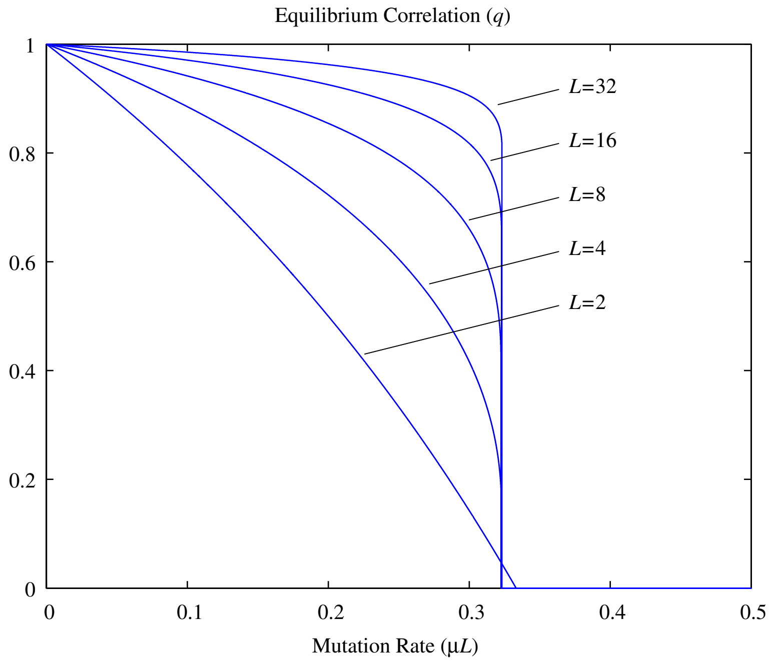

Fig. 4. Calculated results showing the equilibrium correlation for the genetic algorithm with crossover with binary string lengths over a range from $L = 2$ to 32.

case that $L = 2$, the correlation is continuous across the critical mutation rate, but the rate of change is discontinuous (i.e. a second order phase transition).

Now, when using tournament selection, the selection strength can be described by a parameter, s, that determines the probability with which the fitter individual of the two is selected (otherwise a random choice is made). In Fig. 5

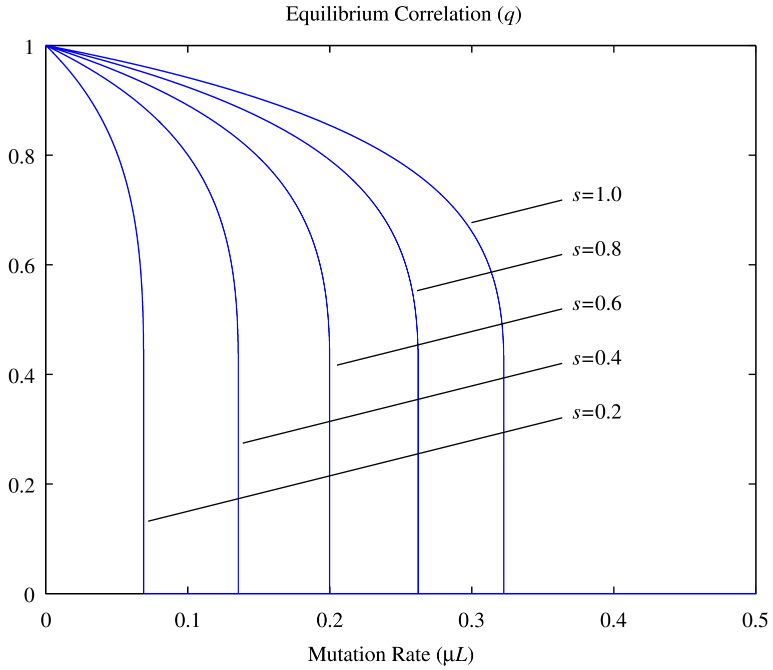

Fig. 5. Calculated results showing the equilibrium correlation for the genetic algorithm with crossover for selection strengths from $s = 0.2$ to 1.0 (in all cases $L = 8$).

the equilibrium correlation is shown for a range of different selection strengths (and $L = 8$). As shown, as selection strength decreases, the critical mutation rate occurs at a lower mutation rate (this dependence is shown analytically for the special case when $L = 2$, in Appendix C). However, the form of the phase transition, and the discontinuity in correlation when $L > 2$, is unchanged.

Having found the equilibrium correlation, the equilibrium variance can be simply found through the relationship described in Eq. (9). Fig. 6 shows the theoretical equilibrium variance and correlation plotted against experimental results for genetic algorithms with and without crossover. The theoretical results for genetic algorithm with crossover are calculated as described here. For the genetic algorithm, without crossover, the equilibrium state can be found by the direct effect of selection and mutation on the population distribution (see Appendix D for full details). We have assumed a large population and thus there is a small discrepancy between the calculated values and experimental results for the genetic algorithm with crossover, close to the critical mutation rate, due to finite population effects. However, the theoretical results clearly accurately describe the experimental results. The change in population variance is very marked in this example, due to the large population and the complete crossover operator. However, we see the same rapid increase in population variance and a divergence between the behaviour of the two algorithms. We know that the increase in population variance is driven by a rapid decrease in the correlation of the population, and thus the diversity of the population increases rapidly at the critical mutation rate.

4.4. Discussion

Solving for the equilibrium correlation of the population, clearly demonstrates the existence of a classic phase transition. However, it does not tell the whole story. Implicit in the calculation of the effect that selection has on correlation, is the fact that the phase transition and the divergence of behaviour of the two algorithms results from the problem space allowing solutions with different genotypes to exhibit the same fitness value (i.e. in the case of the subset sum problem, any particular sum can be expressed by a number of different subsets). It is this feature that differentiates this phase transition from the standard error threshold. In fact, this condition is extremely common in optimisation problems. For example, in

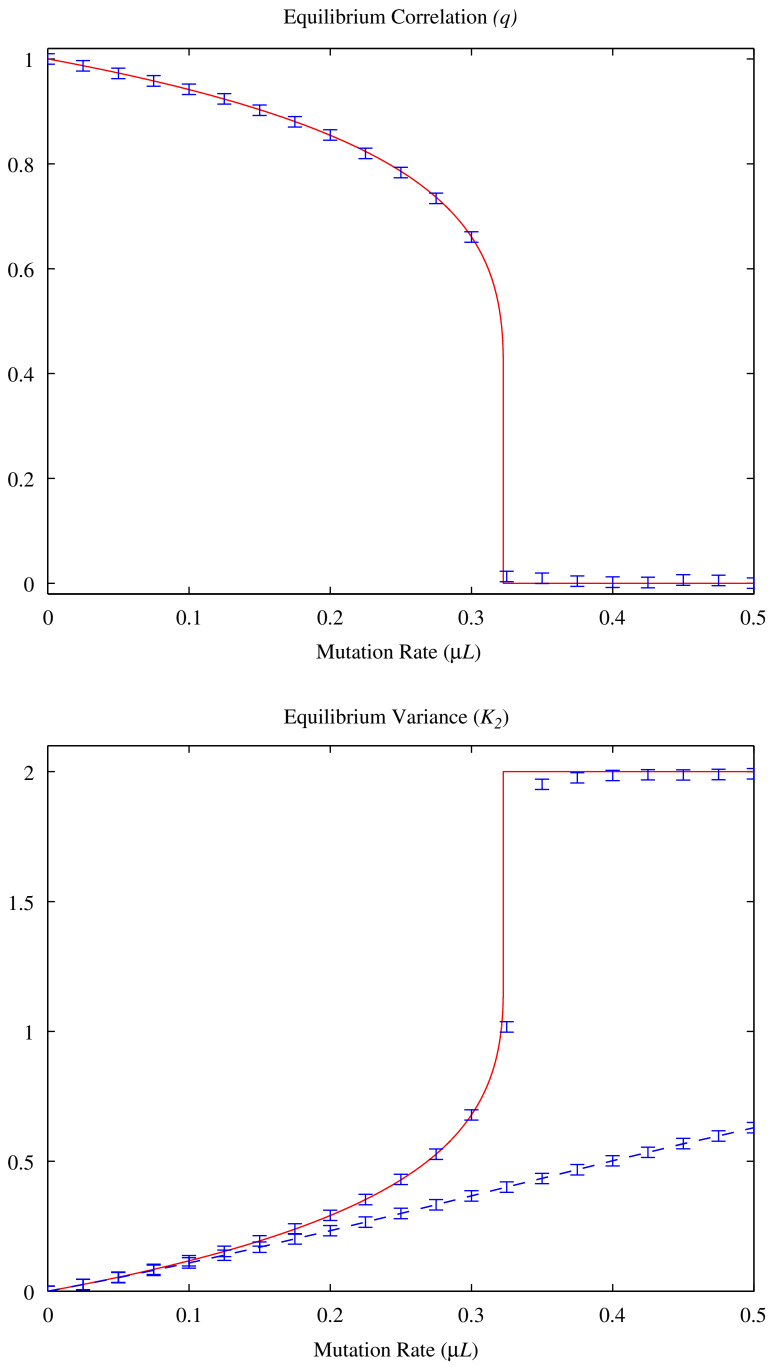

Fig. 6. Comparison of calculated and experimental results for the equilibrium correlation and variance of the genetic algorithm with (*solid line*) and without (*dashed line*) crossover. Population size $P = 10,000$, the length of the binary string $L = 8$, and experimental results are averaged over 100 runs.

graph-colouring and MAXSAT problems, many different solutions can give rise to the same number of colour clashes or false clauses.

The effect of this symmetry is that crossover introduces a form of indirect selection. If we consider two solutions whose sums are close to the target sum, and we apply any form of crossover operator to these two, the resulting offspring are more likely to have sums close to their parents (and thus close to the target sum), if the two parents have similar genotypes. Thus, in the genetic algorithm with crossover, there is an additional indirect selection pressure which favours genotypes which are similar to other genotypes within the population. The result is that the population is driven toward a correlated population where one genotype dominates all others, and the symmetry of the problem is broken. The choice of which genotype is present in the final equilibrium population is arbitrary and depends on the initial condition of the population and random effects due to the stochastic nature of the selection operator.

However, this condition only holds whilst the mutation rate is low. New mutations are continually being introduced into the solutions and the crossover operator is acting to mix those mutations among the different solutions within the population. When the population is highly correlated, with a mean value close to the target sum, the most likely effect of this mutation and crossover, is to generate solutions further from the target value. These solutions are subsequently removed from the population by selection and equilibrium is restored. As the mutation rate increases however, there is an increasing likelihood that rather than generating a solution further from the target sum, crossover and mutation will result in a symmetric genotype equally close to the target value. Selection cannot remove these mutations from the population and in the subsequent generation, crossover acts on these mutations to increase the variance of the population further. At this point, the process repeats and the variance of the population runs away. The result is the completely uncorrelated population predicted by the calculations and observed in the experiments.

5. Conclusions

The phase transition described in the previous section, indicates that the crossover operator has a significant effect on the dynamics of the genetic algorithm. The dynamics, in turn, affect the way in which the algorithm searches the problem space. The critical mutation rate represents a threshold at which crossover is either accelerating symmetry breaking by applying an indirect selection which favours similar genotypes within the population, or preventing symmetry breaking by rapidly mixing new mutations within the population. The analysis in Section 4 shows that this is not a gradual transition, but is actually a discontinuous phase transition.

Whilst the analysis has dealt with a specific simple instance of the subset sum problem. It is clear that the same phenomenon is present in the experimental results presented in Section 3. We see the same rapid reduction in correlation and a corresponding increase in population variance. However, due to the small population size, the less effective crossover operator and the fact that the target sum is located away from the centre of problem space, the size of the variance increase is much reduced.

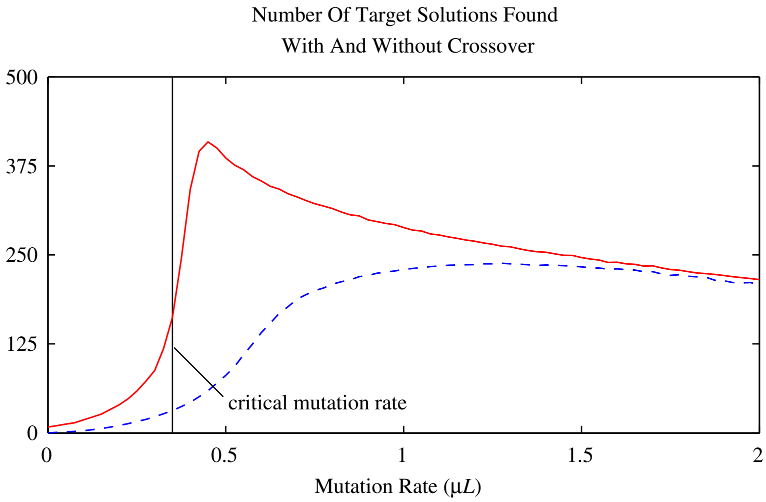

Fig. 7. Experimental results for the number of target solutions found after 1000 generation of a genetic algorithm with (*solid line*) and without (*dashed line*) single point crossover. Population size $P = 100$, string length $L = 128$, the individual values $J_i \in [0, 100]$ and the target sum $T = 4800$. The results are averaged over 100 runs, each with a different randomly generated problem instance.

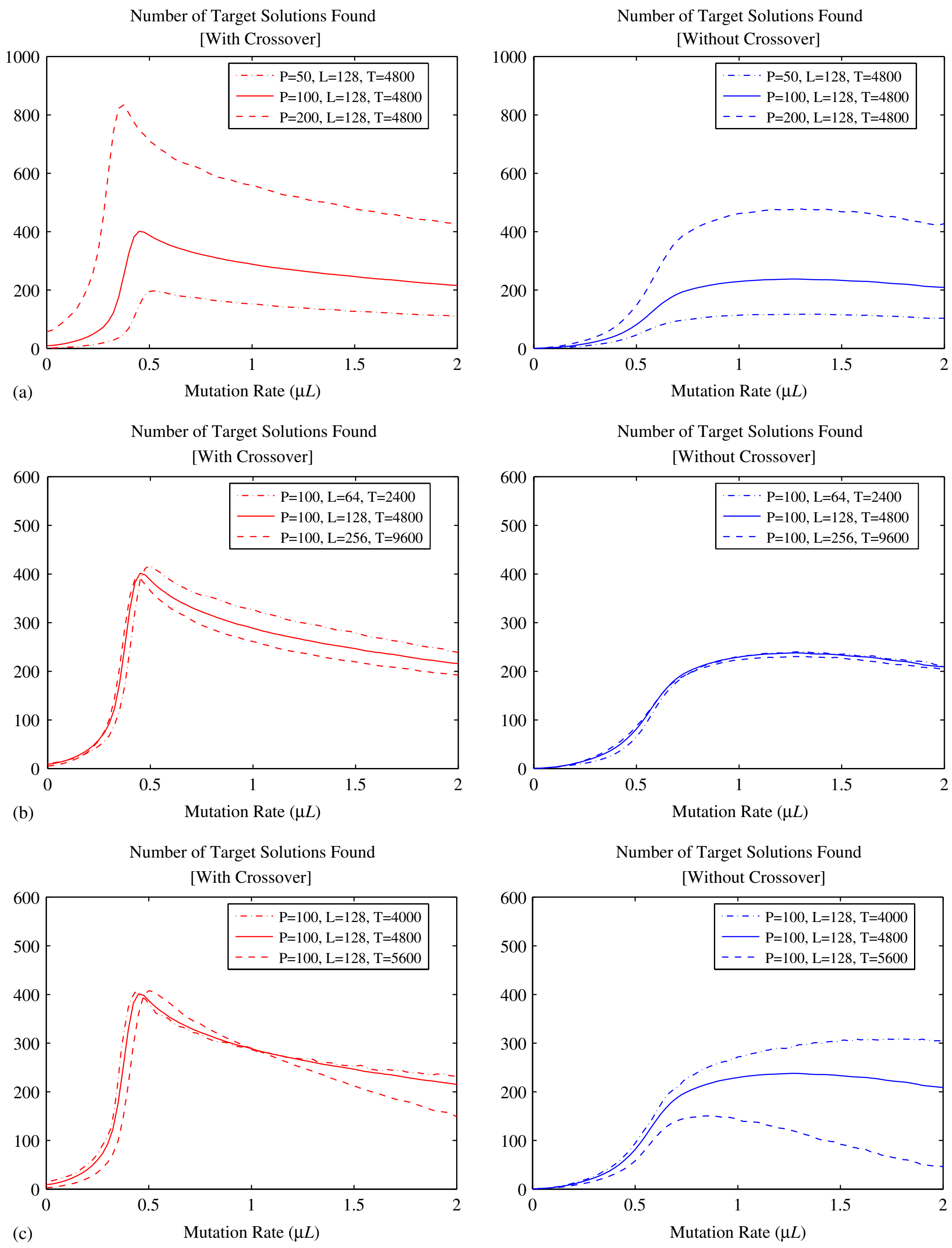

Fig. 8. Experimental results for the number of target solutions found after 1000 generation of a genetic algorithm with and without single point crossover. The default setting is population size $P = 100$, string length $L = 128$, and target sum $T = 4800$. In (a) we vary the population size and compare values of $P = 50$, 100 and 200, in (b) we vary the string length and compare values of $L = 64$, 128 and 256 (also adjusting the value of the target sum to keep its position relative to the range of all possible sums constant) and, in (c) we vary just the value of the target sum and compare values of $T = 4000$, 4800 and 5600. In all cases the individual values $J_i \in [0, 100]$ and the results are averaged over 100 runs, each with a different randomly generated problem instance.

Having made this observation, it is natural to consider how the genetic algorithm is searching the problem space. To do so, we consider the number of different solutions for the target sum that the genetic algorithm has found after a fixed number of generations (i.e. there is no single solution to the subset sum problem as the target sum may be expressed by many different subsets and we investigate how many different solutions that the algorithm has found). This measure reflects the balance of exploitation and exploration within the algorithm. A genetic algorithm which over exploits previous solutions, will rapidly converge to the first target solution found and will thus find very few further solutions. Conversely, a genetic algorithm that explores too much, will not utilise solutions which have already been found to be close to the target sum and will thus also find few target solution.

Fig. 7 shows experimental results for this measure, when applied to the original subset sum problem presented in Section 3 (i.e. the population size $P = 100$, the string length $L = 128$, the individual values $J_i \in [0, 100]$ and the target sum $T = 4800$). The figure clearly show that the genetic algorithm with crossover outperforms the genetic algorithm without crossover. Most importantly, it shows that an optimum in terms of the balance between exploitation and exploration, is achieved close to the critical mutation rate. In comparison with Fig. 1, the peak in the number of target solutions found coincides with the point at which the variance and the correlation are changing most rapidly (we have labelled the critical mutation rate as the point at which the behaviour of the genetic algorithm with crossover first diverges from that of the genetic algorithm without crossover). At this point, the genetic algorithm is able to exploit previously found solutions but also explores the full search space of the problem.

In order to demonstrate that this effect is not dependent on the other parameters of the problem, in Fig. 8 we show the same analysis but in each case we systematically vary one parameter. Thus, in Fig. 8a we use three different population sizes (i.e. $P = 50$, 100 and 200). In Fig. 8b we use three different string lengths (i.e. $L = 64$, 128 and 256) whilst we also simultaneously adjusting the target sum value to $T = 2400$, 4800 and 9600, respectively (thus we adjust the length of the string, but attempt to maintain the same degree of problem hardness by keeping the relative position of the target sum, within the range of all possible sums, constant). Finally, in Fig. 8c, we use three different values of target sum (i.e. $T = 4000$, 4800 and 5600, representing problems in which on average $\frac{5}{8}$, $\frac{3}{4}$ and $\frac{7}{8}$ of the bits within the optimum genotype must be set to one). In each case, despite the changing parameters of the problem, the peak in the number of target solutions found remains close to the location of the critical mutation rate observed earlier. The only small departure from this occurs with decreasing population size, where, as we expect, the increase in finite population effects moves the critical mutation rate to higher values.

Whilst the results presented here are based on the analysis of the subset sum problem, the phenomenon which we observe is very general; all that is required is a genetic algorithm with some form of crossover operator and a problem with symmetry such that many possible genotypes map to a single fitness value. Now, such symmetry is common in optimisation problems, and for example, occurs in both graph-colouring and MAXSAT problems. In addition, both these problems have been the subject of extensive research that has related the hardness of typical problem instances with the particular problem parameters settings (i.e. the average connectivity of the graph for graph colouring and the ratio of clauses to variables in the MAXSAT problem). Thus, as future work we would like to explore how both the hardness and the symmetry of the problem affect the behaviour and existence of the phase transition that we have observed here. Our goal in this work is to better understand how the genetic algorithm is searching these problems, and thus gain insights into how they should best be used on other real-world optimisation problems.

Appendix A. Maximum entropy inference

As discussed in Section 4.2, in order to find the probability distribution of the population, we must know the actual distribution of p_i—the probability of having a one rather than a zero at the ith position in the genotypes—within the population. To do so, we use maximum entropy inference to find the most likely distribution. The entropy, S, is simply the logarithm of the number of possible ways in which the bits within the population can be arranged and thus

$$S = \ln \prod_{i=1}^{L} \binom{P}{p_i P}. \tag{A.1}$$

This term must be maximised, under the constraint imposed by the mean value of the population

$$\sum_{i=1}^{L} p_i = \frac{L}{2} \tag{A.2}$$

and the correlation of the population

$$\sum_{i=1}^{L} p_i^2 = \frac{L(1+q)}{4}. \tag{A.3}$$

In Section 4.2, we presented the result that maximising the entropy over these constraints yields the expression

$$p_i = \frac{1 \pm \sqrt{q}}{2}, \tag{A.4}$$

where p_i forms two equal size classes; one with probability $(1-\sqrt{q})/2$ and one with probability $(1+\sqrt{q})/2$.

Proof. We initially define

$$f(p_i) = \ln \binom{P}{p_i P} \tag{A.5}$$

and thus we seek to maximise

$$S = \sum_{i=1}^{L} f(p_i). \tag{A.6}$$

To reduce the symmetry in the problem, we define a new variable

$$x_i = \left(p_i - \tfrac{1}{2}\right)^2 \tag{A.7}$$

and define the function

$$g(x_i) = \ln \binom{P}{\left(\sqrt{x}_i + \frac{1}{2}\right) P}. \tag{A.8}$$

From these two expressions, it follows that if $p_i > \frac{1}{2}$, then $\sqrt{x}_i = p_i - \frac{1}{2}$ and thus $g(x_i) = \ln \binom{P}{Pp_i} = f(p_i)$. Similarly, if $p_i < \frac{1}{2}$ then $\sqrt{x}_i = \frac{1}{2} - p_i$ and $g(x_i) = \ln \binom{P}{(1-p_i)P} = \ln \binom{P}{p_i P} = f(p_i)$. Now from the two constraints, we have

$$\begin{aligned} \frac{1}{L}\sum_{i=1}^{L} x_i &= \frac{1}{L}\sum_{i=1}^{L} \left(p_i - \frac{1}{2}\right)^2 \\ &= \frac{1}{L}\sum_{i=1}^{L} \left(p_i^2 - p_i + \frac{1}{4}\right) \\ &= \frac{q}{4}. \end{aligned} \tag{A.9}$$

Since $x_i \geqslant 0$ and $\sum x_i$ is a constant, it follows from the assumption that $g(x)$ is a concave function (see below for a proof of this condition) and Jensen's inequality that

$$\frac{1}{L}\sum_{i=1}^{L} g(x_i) \leqslant g\left(\frac{1}{L}\sum_{i=1}^{L} x_i\right), \tag{A.10}$$

where equality hold when all x_i are the same i.e. $x_i = x = q/4$. But we have shown $g(x_i) = f(p_i)$ so

$$S \leqslant Lg\left(\frac{q}{4}\right). \tag{A.11}$$

Thus, the maximum value of S occurs when the equality holds (i.e. when all the values of x_i are the same). This occurs when $x_i = \left(p_i - \frac{1}{2}\right)^2 = q/4$, and thus

$$p_i = \frac{1 \pm \sqrt{q}}{2}. \tag{A.12}$$

To comply with the constraint given in Eq. (A.2), the genotype must separate into two equal-sized classes; one where the probability is given by $(1-\sqrt{q})/2$ and one where it is $(1+\sqrt{q})/2$. □

Proposition 1. *The function* $g(x) = \ln \binom{P}{(\sqrt{x}+\frac{1}{2})P}$ *is a concave function.*

Proof. To prove $g(x)$ is concave we use the Weierstrass formula for the Gamma function which can be written as

$$\ln \Gamma(z+1) = -\gamma z + \sum_{k=1}^{\infty}\left[\frac{z}{k} - \ln\left(1+\frac{z}{k}\right)\right]. \tag{A.13}$$

The logarithm of the binomial coefficient can be written

$$\begin{aligned}\ln\binom{N}{n} &= \ln\Gamma(N+1) - \ln\Gamma(n+1) - \ln\Gamma(N-n+1)\\ &= \sum_{k=1}^{\infty}\left[\ln\left(1+\frac{n}{k}\right) + \ln\left(1+\frac{N-n}{k}\right) - \ln\left(1+\frac{N}{k}\right)\right].\end{aligned} \tag{A.14}$$

We have $g(x) = \ln\binom{P}{P(\sqrt{x}+\frac{1}{2})}$ and thus

$$g(x) = \sum_{k=1}^{\infty}\left[\ln\left(1+\frac{P\left(\frac{1}{2}+\sqrt{x}\right)}{k}\right) + \ln\left(1+\frac{P\left(\frac{1}{2}-\sqrt{x}\right)}{k}\right) - \ln\left(1+\frac{P}{k}\right)\right]. \tag{A.15}$$

Differentiating this gives

$$g'(x) = \sum_{k=1}^{\infty}\frac{-P^2}{k^2+kP+P^2/4-P^2x}. \tag{A.16}$$

Differentiating again gives

$$g''(x) = \sum_{k=1}^{\infty}\frac{-P^4}{(k^2+kP+P^2/4-P^2x)^2} \tag{A.17}$$

which is convergent for all $x > 0$. Since every term in the sum is strictly negative, we can conclude that $g''(x) < 0$ and thus $g(x)$ is concave as required. □

Appendix B. Calculating the effect of selection on correlation

As described in Section 4, we use maximum entropy inference to calculate the values of p_i within the population, and then calculate the effect of selection by considering the fitness of the individuals within the population. More specifically, we consider the probability of a population member having n ones within its genotype to be $P(S=n)$, and we calculate the weighting, $W(S=n)$, which is applied through selection.

Now, with tournament selection, this weighting is complex since it is dependent on the comparison of two individuals drawn from the population. In our simple subset sum instance, the target value is $L/2$, and thus the first individual is selected if it is equally close or closer to $L/2$ than the second individual, and the second individual is selected if it is closer to $L/2$ than the first. Thus, if we denote $P'(S=n)$ as the probability that $S=n$ after selection, and we initially consider the case that $n < L/2$, we can calculate $P'(S=n)$ by considering the probability of these two events occurring

$$\begin{aligned}P'(S=n) = P(S=n)&\left[\sum_{m=0}^{n}P(S=m) + \sum_{m=L-n}^{L}P(S=m)\right]\\ +P(S=n)&\left[\sum_{m=0}^{n-1}P(S=m) + \sum_{m=L-n+1}^{L}P(S=m)\right].\end{aligned} \tag{B.1}$$

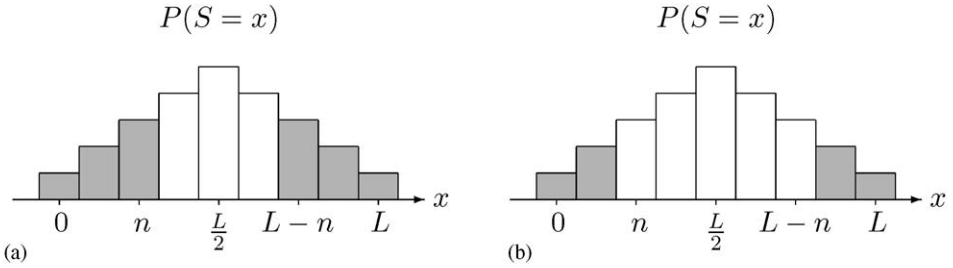

Fig. B.1. Diagram indicating the distribution of individuals with fitnesses (a) equal or less than those with $S = n$, and (b) less than those with $S = n$.

These two summations over $P(S = n)$ are shown as the shaded areas in Fig. B.1 and they can be re-expressed in order to simplify the expression

$$P'(S = n) = P(S = n)\left[1 - \sum_{m=n+1}^{L-n-1} P(S = m)\right] + P(S = n)\left[1 - \sum_{m=n}^{L-n} P(S = m)\right]. \tag{B.2}$$

Then, by expressing $P'(S = n)$ as the product of the initial value of $P(S = n)$ and the weighting factor $W(S = n)$, we can combine the two bracketed expressions and give the weighting factor as

$$W(S = n) = \left[2 - \sum_{m=n}^{L-n} P(S = m) - \sum_{m=n+1}^{L-n-1} P(S = m)\right]. \tag{B.3}$$

Finally, noting that this expression must be symmetrical about $L/2$, and thus $W(L/2 - k) = W(L/2 + k)$, we can derive the final result

$$W(S = n) = \begin{cases} 2 - \sum_{m=n}^{L-n} P(S = m) - \sum_{m=n+1}^{L-n-1} P(S = m), & n < L/2, \\ 2 - P(S = n), & n = L/2, \\ 2 - \sum_{m=L-n}^{n} P(S = m) - \sum_{m=L-n+1}^{n-1} P(S = m), & n > L/2. \end{cases} \tag{B.4}$$

In addition, it is common to implement a selection strength parameter, s, that varies between 0 and 1. In this case, when the two individuals are compared, the fitter individual is selected with probability s, otherwise a random choice is made. We can thus express the weighting in terms of this parameter

$$W(S = n) = (1 - s) + \begin{cases} s\left[2 - \sum_{m=n}^{L-n} P(S = m) - \sum_{m=n+1}^{L-n-1} P(S = m)\right], & n < L/2, \\ s[2 - P(S = n)], & n = L/2, \\ s\left[2 - \sum_{m=L-n}^{n} P(S = m) - \sum_{m=L-n+1}^{n-1} P(S = m)\right], & n > L/2 \end{cases} \tag{B.5}$$

noting that when $s = 1$ we have the same result as before, and when $s = 0$, then $W(S = n) = 1$ and the population is unchanged by selection.

We are now left with the calculation of the two probabilities $P(S = n)$ and $P(S = n \,\&\, X_i = 1)$. These can be derived by considering the genotype of length L divided into two halves, each of length of $L/2$. As discussed in Section 4.2 and shown in detail in Appendix A, at equilibrium $p_i = (1 \pm \sqrt{q})/2$. Thus, in one half of the genotype

$P(X_i = 1) = (1 + \sqrt{q})/2$ and in the other $P(X_i = 1) = (1 - \sqrt{q})/2$. Thus, if $S = n$ and there are m ones in one half of the genotype, there must be $n - m$ ones in the other half. Thus, the probability $P(S = n)$ is given by summing over all possible values of m and calculating the probability of each event. The result is

$$P(S=n) = \begin{cases} \displaystyle\sum_{m=0}^{n} \binom{\frac{L}{2}}{m} \left(\frac{1+\sqrt{q}}{2}\right)^{m} \left(1 - \frac{1+\sqrt{q}}{2}\right)^{L/2-m} \\ \displaystyle\quad \times \binom{L/2}{n-m} \left(\frac{1-\sqrt{q}}{2}\right)^{n-m} \left(1 - \frac{1-\sqrt{q}}{2}\right)^{L/2-n+m}, & n \leqslant \frac{L}{2}, \\ \displaystyle\sum_{m=n-L/2}^{L/2} \binom{\frac{L}{2}}{m} \left(\frac{1+\sqrt{q}}{2}\right)^{m} \left(1 - \frac{1+\sqrt{q}}{2}\right)^{L/2-m} \\ \displaystyle\quad \times \binom{\frac{L}{2}}{n-m} \left(\frac{1-\sqrt{q}}{2}\right)^{n-m} \left(1 - \frac{1-\sqrt{q}}{2}\right)^{L/2-n+m}, & n > \frac{L}{2} \end{cases}$$

which simplifies to

$$P(S=n) = \sum_{m=\max(0,n-L/2)}^{\min(L/2,n)} \binom{\frac{L}{2}}{m} \binom{\frac{L}{2}}{n-m} \left(\frac{1+\sqrt{q}}{2}\right)^{L/2-n+2m} \left(\frac{1-\sqrt{q}}{2}\right)^{L/2+n-2m}. \tag{B.6}$$

The second probability, $P(S = n \;\&\; X_i = 1)$, is simply derived by fixing one position of the genotype to be a one rather than a zero. Now, for there to be a total of n ones in the genotype, there must be an additional $m - 1$ ones in that half and, as before, $n - m$ in the other half. This results in the expression

$$P(S=n \;\&\; X_i=1) = \begin{cases} \displaystyle\left(\frac{1+\sqrt{q}}{2}\right) \sum_{m=1}^{n} \binom{\frac{L}{2}-1}{m-1} \left(\frac{1+\sqrt{q}}{2}\right)^{m-1} \left(1 - \frac{1+\sqrt{q}}{2}\right)^{L/2-m} \\ \displaystyle\quad \times \binom{\frac{L}{2}}{n-m} \left(\frac{1-\sqrt{q}}{2}\right)^{n-m} \left(1 - \frac{1-\sqrt{q}}{2}\right)^{L/2-n+m}, & n \leqslant \frac{L}{2}, \\ \displaystyle\left(\frac{1+\sqrt{q}}{2}\right) \sum_{m=n-L/2}^{L/2} \binom{\frac{L}{2}-1}{m-1} \left(\frac{1+\sqrt{q}}{2}\right)^{m-1} \left(1 - \frac{1+\sqrt{q}}{2}\right)^{L/2-m} \\ \displaystyle\quad \times \binom{\frac{L}{2}}{n-m} \left(\frac{1-\sqrt{q}}{2}\right)^{n-m} \left(1 - \frac{1-\sqrt{q}}{2}\right)^{L/2-n+m}, & n > \frac{L}{2} \end{cases}$$

which simplifies to

$$P(S=n \;\&\; X_i=1) = \sum_{m=\max(1,n-L/2)}^{\min(L/2,n)} \frac{2m}{L} \binom{\frac{L}{2}}{m} \binom{\frac{L}{2}}{n-m} \left(\frac{1+\sqrt{q}}{2}\right)^{L/2-n+2m} \left(\frac{1-\sqrt{q}}{2}\right)^{L/2+n-2m}. \tag{B.7}$$

Appendix C. Special case when $L = 2$

Now, when $L = 2$ the equilibrium has a simple analytical form. Thus, using Eq. (B.6) gives

$$\begin{aligned} P(S=0) &= \frac{1-q}{4}, \\ P(S=1) &= \frac{1+q}{2}, \\ P(S=2) &= \frac{1-q}{4}. \end{aligned} \tag{C.1}$$

Likewise, using Eq. (B.7) gives

$$\begin{aligned} &P(S=0\ \&\ X_1=1)=0,\\ &P(S=1\ \&\ X_1=1)=\left(\frac{1+\sqrt{q}}{2}\right)^2,\\ &P(S=2\ \&\ X_1=1)=\frac{1-q}{4}. \end{aligned} \tag{C.2}$$

Finally, using the results in Eq. (C.1) in Eq. (B.4) gives the weightings

$$\begin{aligned} &W(S=0)=(1-s)+s\left(\frac{1-q}{2}\right),\\ &W(S=1)=(1-s)+s\left(\frac{3-q}{2}\right),\\ &W(S=2)=(1-s)+s\left(\frac{1-q}{2}\right), \end{aligned} \tag{C.3}$$

where s is the selection strength. Applying these expression in Eq. (17) and then Eq. (18), yields a simple expression for the correlation after selection

$$q'=\frac{q}{4}\left(sq-s-2\right)^2. \tag{C.4}$$

Now, the equilibrium correlation is found by equating the correlation after both selection and mutation with the initial correlation

$$q=\frac{q}{4}\left(1-2\gamma\right)^2\left(sq-s-2\right)^2 \tag{C.5}$$

and solving to give

$$q=1+\frac{2}{s}\left(1-\frac{1}{1-2\gamma}\right). \tag{C.6}$$

Thus, when $L=2$, as the mutation rate increases, the equilibrium correlation decreases continuously to zero (with a discontinuity in the rate of decrease when $q=0$). The critical mutation rate which the correlation equals zero, γ^*, is given by

$$\gamma^*=\frac{s}{2\left(2+s\right)}. \tag{C.7}$$

Appendix D. Calculating the equilibrium state of the genetic algorithm without crossover

In order to calculate the equilibrium state of the genetic algorithm without crossover, we again consider the probability distribution of solutions within the population and directly calculate the effect that selection and mutation have on this distribution. As in our simple instance of a subset sum problem, all $J_i=1$, we need not consider the actual sum of a population member but only the number of ones within its genotype. Thus, we consider the probability of a population member having n ones within its genotype to be $P(S=n)$, and the effect of selection, is then to simply to weight some values of n over others. Thus the probability distribution after selection, $P'(S=n)$, is given by

$$P'(S=n)=W(S=n)P(S=n), \tag{D.1}$$

where $W(S=n)$ is the weighting of the selection scheme, described in Appendix B and shown in Eq. (B.4).

To calculate the effect of mutation, we initially consider the probability of a population member having m ones within its genotype, $P(S=m)$. We then sum over all the possible mutations that would result in this individual ending with n ones within its genotype. The probability distribution after mutation, $P'(S=n)$, is thus given by

$$P'(S=n)=\sum_{m=0}^{L}P(S=m)\sum_{l=\max(0,m-n)}^{\min(m,L-n)}\binom{m}{l}\binom{L-m}{n-m+l}\mu^{n-m+2l}(1-\mu)^{L+m-n-2l}, \tag{D.2}$$

where μ is the mutation rate.

These two expressions are too complex, to enable us to solve directly for the equilibrium probability distribution. Rather, we calculate the equilibrium distribution by numerically iterating these equations for a sufficient number of generations. Once an equilibrium distribution has been found, the mean (K_1) and variance (K_2) are simply calculated by

$$K_1 = \sum_{n=0}^{L} nP(S = n), \tag{D.3}$$

$$K_2 = \sum_{n=0}^{L} n^2 P(S = n) - \left(\sum_{n=0}^{L} nP(S = n) \right)^2 . \tag{D.4}$$

The results of this calculation are compared against experimental results in Fig. 6. Our calculation has assumed that selection and mutation are deterministic (i.e. we have assumed a large population and ignored any stochastic effects). When the population size is small, we expect some deviation from this result due to finite population effects. However, the agreement displayed in the comparison with experimental results is very good.

References

[1] M. Eigen, Molekulare selbstorganisation und evolution, Naturwissenschaften 58 (1971) 465.

[2] M. Garey, D. Johnson, Computers and Intractability—A Guide to the Theory of NP-Completeness, W.H. Freeman, San Francisco, 1979.

[3] R. Monassonm, R. Zecchina, S. Kirkpatrick, B. Selmans, L. Troyansky, Determining computational complexity from characteristic 'phase transitions', Nature 400 (1999) 133–137.

[4] A. Nix, M.D. Vose, Modeling genetic algorithms with Markov chains, Ann. Math. Artificial Intelligence 5 (1992) 79–88.

[5] G. Ochoa, I. Harvey, Recombination and error thresholds in finite populations, in: W. Banzhaf, C. Reeves (Eds.), Foundations of Genetic Algorithms, Vol. 5, Morgan Kaufmann, San Francisco, 1999, pp. 245–264.

[6] A. Prügel-Bennett, Symmetry breaking in population-based optimization, IEEE Trans. Evolutionary Comput. 8 (1) (2004) 63–79.

[7] A. Prügel-Bennett, J.L. Shapiro, An analysis of genetic algorithms using statistical mechanics, Phys. Rev. Lett. 72 (9) (1994) 1305–1309.

[8] A. Prügel-Bennett, J.L. Shapiro, The dynamics of a genetic algorithm for simple random Ising systems, Physica D 104 (1997) 75–114.

[9] M. Rattray, The dynamics of a genetic algorithm under stabilizing selection, Complex Systems 9 (3) (1995) 213–234.

[10] A. Rogers, Phase transitions in evolving populations, Phys. Rev. Lett. 90 (15) (2003) 158103.

[11] A. Rogers, A. Prügel-Bennett, The dynamics of a genetic algorithm on a model hard optimization problem, Complex Systems 11 (6) (2000) 437–464.

[12] J.L. Shapiro, A. Prügel-Bennett, M. Rattray, A statistical mechanical formulation of the dynamics of genetic algorithms, in: T.C. Fogarty (Ed.), Lecture Notes in Computer Science, Vol. 865, Springer, Berlin, 1994, pp. 17–27.

[13] G. Syswerda, Simulated crossover in genetic algorithms, in: L.D. Whitley (Ed.), Foundations of Genetic Algorithms, Vol. 2, Morgan Kaufmann, San Francisco, 1993, pp. 239–255.

[14] M.D. Vose, The Simple Genetic Algorithm: Foundations and Theory, MIT Press, Cambridge, MA, 1999.

[15] A. Wright, J. Rowe, J.R. Neil, Analysis of the simple genetic algorithm on the single peak and double peak landscapes, in: Proc. of the 2002 Congress on Evolutionary Computation, IEEE, 2002, pp. 214–219.

[16] A. Wright, J. Rowe, R. Poli, C. Stephens, Bistability in a gene pool GA with mutation, in: K. De Jong, R. Poli, J. Rowe (Eds.), Foundations of Genetic Algorithms, Vol. 7, Morgan Kaufmann, San Francisco, 2003, pp. 63–80.

[17] S. Wright, Evolution and the Genetics of Populations, Chicago University Press, 1968.