

# Another Investigation on Tournament Selection: modelling and visualisation

Huayang Xie  
Victoria University of  
Wellington  
Wellington, New Zealand  
hxie@mcs.vuw.ac.nz

Mengjie Zhang  
Victoria University of  
Wellington  
Wellington, New Zealand  
mengjie@mcs.vuw.ac.nz

Peter Andreae  
Victoria University of  
Wellington  
Wellington, New Zealand  
pondy@mcs.vuw.ac.nz

## ABSTRACT

Tournament selection has been widely used and studied in evolutionary algorithms. To supplement the study of tournament selection, this paper provides several models describing the probabilities that a program of a particular rank is sampled and is selected in the standard tournament selection in a simple situation and a complex situation. This paper discovers that, with the same tournament size, trends of sampling probability of a program and selection probability distributions of a population are the same regardless of the population size. This paper also models and investigates an alternative tournament selection method which eliminates one of the drawbacks in the standard tournament selection. Finally, this paper proposes a new fitness evaluation saving algorithm via the use of not-sampled individuals, which is a special property of tournament selection.

## Categories and Subject Descriptors

I.6 [Simulation and Modelling]: Model Validation and Analysis

## General Terms

Theory

## Keywords

Tournament Selection, Modelling, Visualisation

## 1. INTRODUCTION

Selection is intended to improve the average quality of the population by giving individuals of higher quality a higher probability of being exploited in the next generation [1]. There is a wide range of selection techniques in Evolutionary Algorithms (EAs), including *fitness-proportional selection*, *ranking selection*, and *tournament selection* [5].

The standard tournament selection method repeatedly randomly samples  $r$  individuals uniformly and with replacement from the current population of size  $S$  and selects the

one with the best fitness [9]. The popularity of tournament selection is growing rapidly, because its selection pressure can be changed by the tournament size for different problem domains in order to tune the convergence of an evolutionary learning algorithm. It is also simple to code, efficient for both non-parallel and parallel architectures [7], and does not require sorting the whole population first (it has the time complexity  $O(rS)$ ).

A drawback of the standard tournament selection is the combination of high between-group selection pressure and random in-group selection. During population convergence, groups of programs have the same or similar fitness values, so that the selection pressure increases between groups resulting in “better” groups dominating the next population and speeding up the convergence. On the other hand, the selection pressure effectively decreases within a group causing the selection to become random.

Although tournament selection has been studied since the 1990s, most papers only study the standard tournament selection in a simple situation where each individual has a unique fitness. There are some models of the probability of an individual being sampled and the probability of an individual of a particular rank being selected. However, there has been no attempt to discover the relationship between population size and tournament size via these probabilities. Further, few papers focus on modelling the probabilities of all individuals being selected in the more complex situation where some individuals share the same fitness value.

Many alternative tournament selection methods have also been developed since the 1990s. However, their effectiveness is mainly demonstrated through experiments. The lack of formal modelling prevents researchers from understanding the working of these alternative tournament selection strategies, and from extending or developing new strategies.

“Not-sampled” is a special characteristic in tournament selection. Poli introduced the use of not-sampled individuals to make savings on the fitness evaluation cost [9]. However, several limitations exist in that work and have not yet been properly addressed.

## 1.1 Goals

This paper aims to provide models and visualisations to supplement the study of tournament selection. Specifically, we investigate: (1) the relationship between population size and tournament size, (2) the probability of an individual of a particular rank being selected in a complex situation, (3) the working of *clustering tournament selection* [13], and (4) a limitation in [9] and a new evaluation saving algorithm.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom  
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

Note that, hereafter, unless otherwise noted, our investigations are described and discussed in the context of Genetic Programming (GP).

## 1.2 Assumption and Definition

Let  $S$  be the size of a population. We follow the standard tournament selection and assume that the standard breeding process is used, that is, one parent produces one offspring after mutation and two parents produce two offspring via crossover. Therefore the total number of tournaments required to generate an entire new population is  $S$ .

We define the *selection probability distribution* of a population to consist of the probabilities of each of the programs in the population being selected at least once in the process of generating the next generation.

## 2. RELATED WORK

### 2.1 Mathematical Analysis

Blickle and Thiele [1] analysed the “fitness distribution” of tournament selection, which provides a way to predict the fitness values after selection, i.e. the fitness values of selected parents, in the context of Genetic Algorithms (GAs). They concluded that, for the same selection intensity, tournament selection has the smallest loss of diversity and the highest selection variance when compared with truncation selection and ranking selection. A fairly comprehensive analysis covering other selection schemes can be found in [2].

Motoki [8] analysed the loss of diversity in a variety of selection schemes, including tournament selection. He showed that in tournament selection, many more individuals are expected to be lost than Blickle and Thiele’s static estimation.

Poli [9] split the tournament selection process into two steps, sampling and selecting. He mentioned that *not sampling* and *not selecting* are two factors that lead to the loss of diversity. The former factor refers to an individual which is never sampled in any tournament. The latter factor refers to a sampled individual which does not win any tournament. He analysed the number of individuals that are not sampled in any tournaments in the process of generating the next generation. Based on the results, he developed a *Backward-chaining* EA (BC-EA) and claims that, if a very small tournament size is used, the algorithm can provide significant saving on fitness evaluations by simply avoiding the creation and evaluation of these *not-sampled* individuals. He obtained about 20% saving with tournament size 2 and about 6% saving with tournament size 3 in his experiments. He also mentioned that when the tournament size is 1 (ie, random selection), the saving will be over 35%.

### 2.2 Alternative Implementations

Harik [4] demonstrated some interesting work in the tournament selection in the context of GAs. He introduced a restricted tournament selection method in GAs for two purposes. The first is to preserve and find multiple solutions and the second is to obtain a particular global solution by taking the advantage of the schema found in multiple local solutions.

Filipović et al. [3] investigated a fine grained tournament selection method for a simple plant location problem in GAs. They argued that the standard tournament selection does not allow precise setting of the balance between exploration

and exploitation [1]. In their fine grained tournament selection method, the tournament size is not fixed but close to a preset value.

Luke and Panait [6] developed two modified tournament selection methods in GP. The methods use *buckets* to apply lexicographic parsimony pressure on program selection for problem domains where few individuals have the same fitness. Each individual in the bucket is treated as if it had the same fitness as others in the same bucket. They concluded that the methods maintain the same mean best-fitness-of-run as the Koza-style depth limiting does, but produces equivalent or significantly lower mean tree sizes.

Sokolov and Whitely [11] introduced an unbiased tournament selection scheme that eliminates the loss of diversity due to individuals not being sampled. The algorithm is based on reducing variance in the number of times a particular individual is picked to participate in a tournament. They showed that the algorithm yields better results than the standard tournament selection in a generational GA.

In our previous work [13], we developed a clustering tournament selection method for crossover in GP. A population is grouped into clusters based on some criteria. Clusters are randomly selected for tournaments, and a program is randomly selected from a winning cluster to participate in the recombination process. Therefore the parent programs from different clusters are dissimilar. We showed that avoiding a crossover between programs with similar behaviours can contribute to maintaining population diversity and improving the performance of a GP system. We think an important property of the clustering tournament selection is that it implicitly but effectively eliminates the drawback of high between-group selection pressure in the standard tournament selection.

## 3. MODELLING AND VISUALISING IN A SIMPLE SITUATION

Our investigation starts from a simple situation where the population is wholly diverse. This situation seldom happens in GP for problems with finite number of possible fitness values. If the situation does happen, it will most likely appear at the initial generation as long as constraints are applied to the initialisation process, and the population size is less than the number of possible unique fitness values.

### 3.1 Sampling probability modelling and visualisation

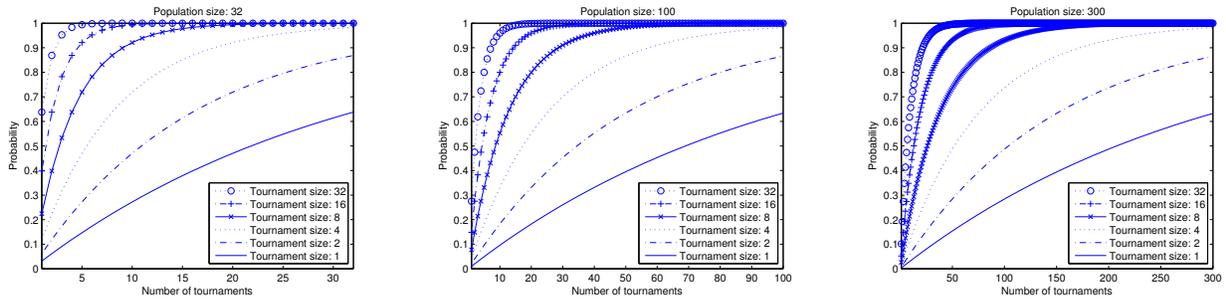
Let  $S$  be the size of a population. Let  $I$  be the event that a program is sampled at least once in a single tournament. There are  $r$  samplings in conducting a tournament. The probability of the event  $I$  is

$$P(I) = 1 - \left(\frac{S-1}{S}\right)^r \quad (1)$$

Let  $I'$  be the event that a program is sampled at least once in  $y$  tournaments. The probability that a program will not be sampled in  $y$  tournaments is

$$(1 - P(I))^y \quad (2)$$

So the probability that a program will be sampled at least once in  $y$  tournaments is the complement of Equation 2.



**Figure 1: Trends of the probability that a program is sampled at least once during the process of generating an entire new population. (Note that the scales on the x-axes differ.)**

That is

$$\begin{aligned}
 P(I^y) &= 1 - (1 - P(I))^y \\
 &= 1 - \left(\frac{S-1}{S}\right)^{ry} \\
 &= 1 - \left(\left(\frac{S-1}{S}\right)^S\right)^{\frac{y}{S}r} \quad (3)
 \end{aligned}$$

Based on Equation 3, it is clear that the sampling probability is independent of program rank and is only a function of population size, tournament size, and number of tournaments.

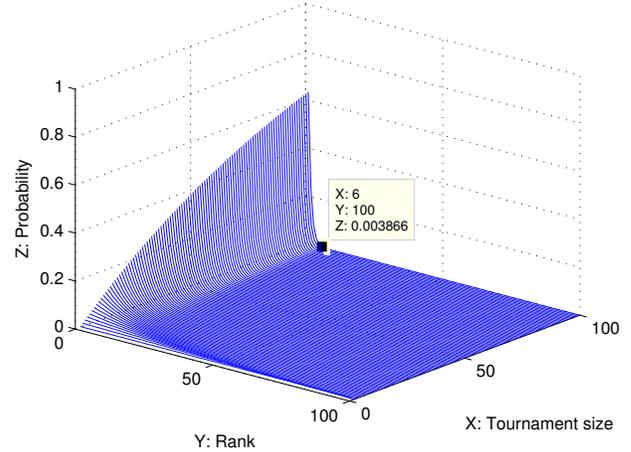
Figure 1 illustrates the probability trends of a program being sampled using six tournament sizes (from 1 to 32 with ratio 2) in populations with three different sizes (32, 100, and 300) when the number of tournaments increases up to the corresponding population size.

It is clear that for a given population size, increasing the tournament size or the number of tournaments raises the probability of an arbitrary program being sampled. The number of tournaments required to obtain a sampling probability very close to 1.0 is inversely proportional to the tournament size and is about  $\frac{4S}{r}$ . For instance, in a population of size 100, the sampling probability is about 0.982 after 100 tournaments when the tournament size is 4, while it becomes 0.985 after only 13 tournaments when the tournament size is 32.

It is also clear that for a fixed tournament size, the sampling probability of a program in a single tournament varies with population size. However, interestingly, the trends of sampling probabilities of a program along the increments of the number of tournaments are very similar in different sized populations. This is because  $\left(\frac{S-1}{S}\right)^S$  is close to a constant for large  $S$ , so sampling probability depends on the ratio of the number of tournaments to the population size. In other words, it depends on the fraction of population generated for the next generation. Therefore, with the same tournament size, sampling probability for large populations can be estimated reliably from experiments on smaller populations at the stage of generating the same fraction of population.

### 3.2 Selection probability distribution: modelling and visualisation

Suppose a population is ranked by fitness values and the best program is ranked 1st. Let  $W_j$  be the event that the  $j$ th ranked program is selected by a tournament. The probability of this event is given by the probability that all  $r$  programs in a tournament are of rank  $j$  or worse, minus the



**Figure 2: Selection probabilities of programs with different ranks in a *single* tournament using different tournament sizes.**

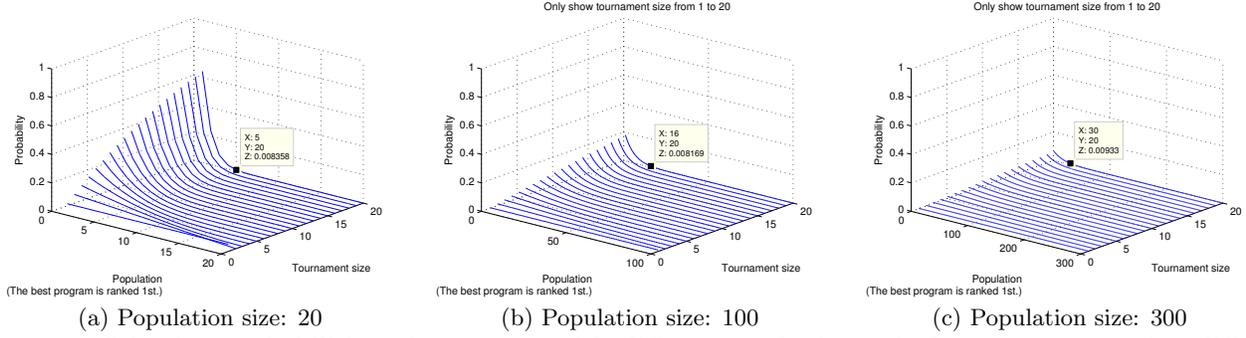
probability that all  $r$  programs in a tournament are of rank  $j + 1$  or worse:

$$P(W_j) = \frac{(1 + S - j)^r - (S - j)^r}{S^r} \quad (4)$$

Let  $W'_j$  be the event that the  $j$ th ranked program is selected at least once in  $y$  tournaments. The probability of this event is

$$\begin{aligned}
 P(W'_j) &= 1 - (1 - P(W_j))^y \\
 &= 1 - \left(1 - \frac{(1 + S - j)^r - (S - j)^r}{S^r}\right)^y \quad (5)
 \end{aligned}$$

Figure 2 illustrates the selection probabilities of programs with different ranks in a *single* tournament, where the population size is 100, and the tournament size ranges from 1 to 100. It shows that the selection probability of the 1st program increases significantly as the tournament size increases, while the selection probabilities of low ranked programs decrease: the selection is biased more and more heavily towards high ranked programs as the selection pressure increases. When the tournament size is the same as the population size, which represents an extremely high selection pressure applied, only the best five programs have a selection probability greater than 1% (see the data label in the figure).



**Figure 3: Selection probabilities of programs with different ranks in a *single* tournament using different tournament sizes up to 20.**

Figure 3 zooms in on the small tournament size portion of this graph, and also shows the same portion for two other population sizes.

For the same tournament size, the selection probabilities of programs in a population with large size are overall flatter than that in a population with small size. It means that selection pressure decreases when population size increases.

For the same population size, the selection probability of the high ranked program decreases when the tournament size decreases. But the reductions become less significant when population size increases. Therefore, in a *single* tournament, the selection probability of each program in a population is clearly affected by the population size and the tournament size.

Figure 4 illustrates the selection probability distributions for two different population sizes (100 and 300) and three different tournament sizes (4, 8, and 16). In all cases, the fitness values of the programs are assumed to be unique.

For a given population size, the flat region (probabilities close to 0) in a selection probability distribution becomes larger as tournament size increases, meaning that more and more low ranked programs are thrown away in the selection process due to the increased selection pressure.

With the same tournament size, although the selection probability of a program with a given rank varies in different sized populations, the selection probability distributions for different population sizes remain the *same shape* and are dependent primarily on the fraction of population at the next generation that has been generated. This indicates that it is not necessarily to investigate an effective tournament size for every different population size. As for the sampling probability, the results of empirical studies with small populations on the effects of different tournament sizes should also be applicable to larger population sizes.

#### 4. MODELLING AND VISUALISING IN A COMPLEX SITUATION

Our further investigation focuses on a more complex situation where some programs have the same fitness value. In this case, programs with the same fitness value are given the same rank.

As sampling is independent of program ranking, the sampling probability model in a complex situation is the same as that in a simple situation. However, modelling the selection probability of a program ranked  $j$ th is more difficult because the probability will be affected by the number of programs with the same rank, the probability of any one of

these programs being sampled, and the probabilities of any programs with worse fitness values being sampled.

Let the set of programs be  $S$ , and let the set of programs with the rank  $j$  be  $S_j$ . Let  $Q$  be the number of distinct fitness values. Let  $W_j$  be the event that a particular program in  $S_j$  is selected in a tournament involving  $r$  samples. Let  $W'_j$  be the event that a certain program in  $S_j$  is selected at least once in  $y$  tournaments.

The probability that all the programs sampled for a tournament have a fitness value between  $j$  and  $Q$  (ie, are from  $S_j \dots S_Q$ ) is given by

$$\left( \frac{\sum_{i=j}^Q |S_i|}{|S|} \right)^r$$

Let  $T_j$  be the event that the highest ranked program in a tournament is from  $S_j$ . Therefore, the probability that the selected program will have rank  $j$  is

$$P(T_j) = \left( \frac{\sum_{i=j}^Q |S_i|}{|S|} \right)^r - \left( \frac{\sum_{i=j+1}^Q |S_i|}{|S|} \right)^r \quad (6)$$

As each element of  $S_j$  has equal probability of being selected in a tournament, for any given program  $p \in S_j$ , the probability that  $p$  will be selected in a tournament is

$$P(W_p) = \frac{\left( \frac{\sum_{i=j}^Q |S_i|}{|S|} \right)^r - \left( \frac{\sum_{i=j+1}^Q |S_i|}{|S|} \right)^r}{|S_j|} \quad (7)$$

Therefore the probability that  $p$  is selected at least once in  $y$  tournaments is

$$P(W'_p) = 1 - (1 - P(W_p))^y \quad (8)$$

$$= 1 - \left( 1 - \frac{\left( \frac{\sum_{i=j}^Q |S_i|}{|S|} \right)^r - \left( \frac{\sum_{i=j+1}^Q |S_i|}{|S|} \right)^r}{|S_j|} \right)^y \quad (9)$$

A wholly diverse population can be seen as each program having a unique fitness for each rank  $j$ . In other words  $|S_j| = 1$ . Therefore Equations 4 and 5 can be obtained by replacing each  $|S_j|$  in Equations 7 and 8 by 1 and simplifying, respectively.

Two simulations were conducted to visualise the selection probability distributions in the complex situation. The first one is a restricted case, where only one (arbitrary) fitness

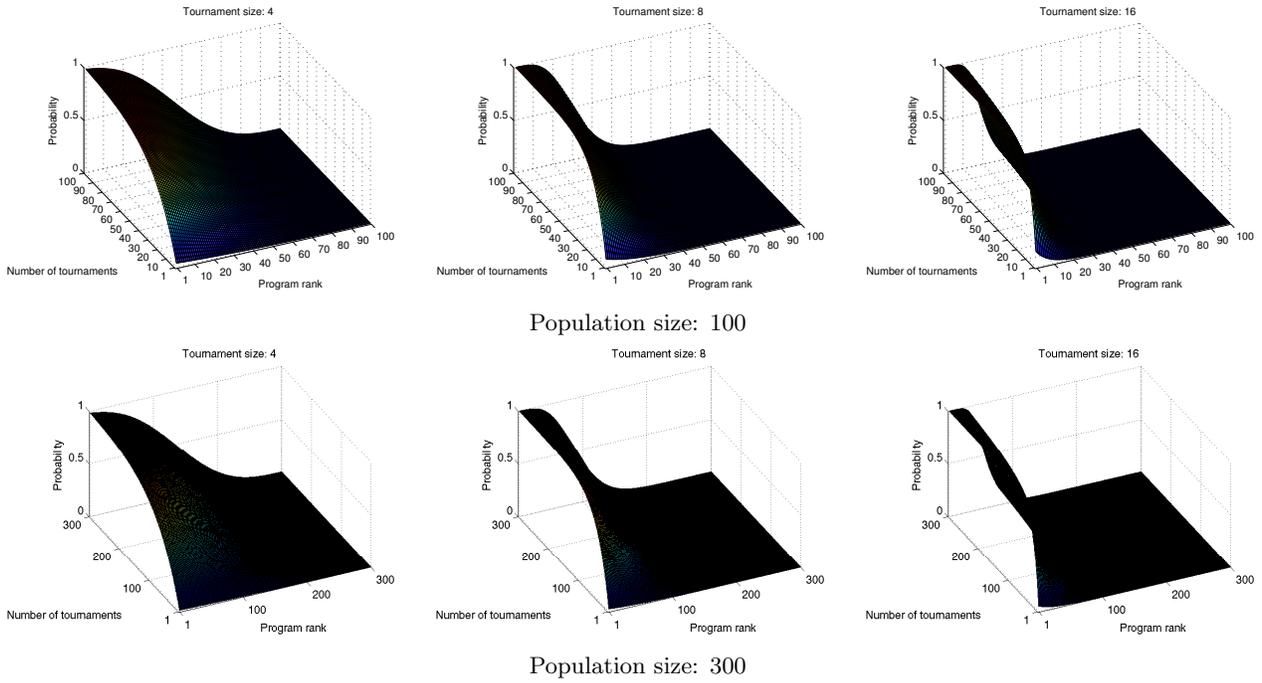


Figure 4: Selection probability distributions in a simple situation.

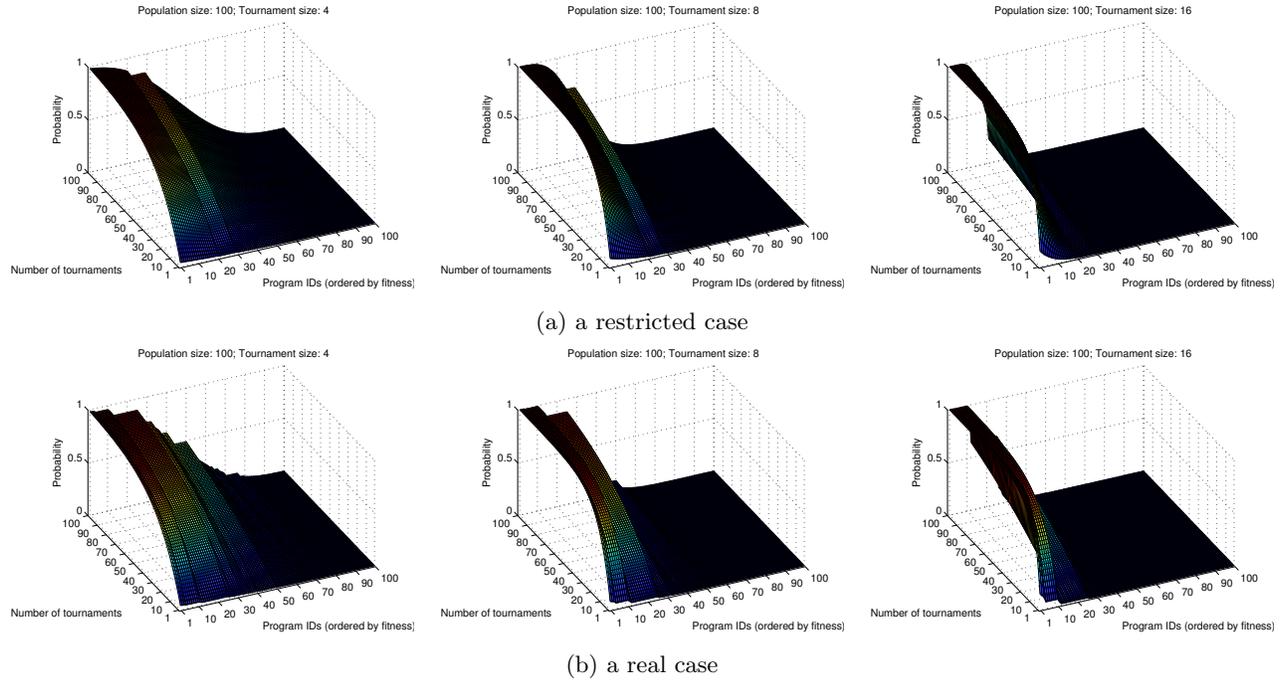


Figure 5: Selection probability distributions in a complex situation.

value is associated with several programs; each of the other fitness values is still mapped to a unique program. The selection probability distribution in this restricted case gives an indication of how the distribution will be changed when a population has some programs with the same fitness value. The change is a stair-like probability distribution breaking the smooth surface (see Figure 5 (a)).

The second simulation uses data taken from a randomly chosen generation in a real GP run on a symbolic regression problem (see Figure 5 (b)).

When comparing the corresponding probability distributions in Figure 5 with those in Figure 4, we observe that:

1. with the same tournament size, the corresponding outlines of the selection probability distributions remain unchanged;
2. the group of programs with the same fitness have the same probability of being selected, showing that when a single top ranked group covers most of the population, the whole selection process becomes random;

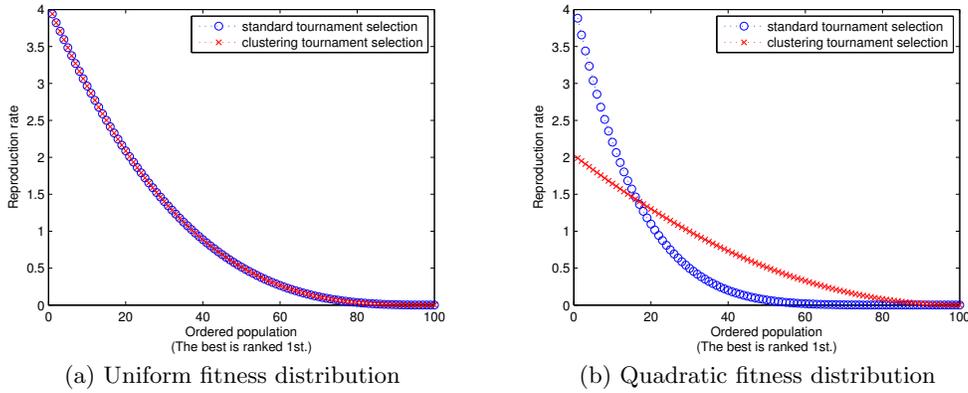


Figure 6: Comparison of the reproduction rate.

3. as a corollary of the previous observation, the width of a stair-like probability distribution change is proportional to the number of these programs;
4. the depth of a stair-like probability distribution change is affected by the tournament size and the number of tournaments. The larger the tournament size, the deeper the change becomes (except programs falling in the flat region). The change is also amplified when the number of tournaments increases.

## 5. MODELLING CLUSTERING TOURNAMENT SELECTION

One of our previous work for dealing with the high between-group selection pressure drawback is an alternative tournament selection method called *clustering tournament selection* [13]. This section provides a model to investigate how it can well maintain the population diversity.

The clustering tournament selection assumes a population is grouped by fitness values and each cluster is a single sampling candidate and is equivalent to a set of programs  $S_j$  mentioned in Section 4. Therefore, the number of sampling candidates is reduced from the population size to the number of clusters.

Using the same notation as in Section 4  $P(T_j)$  (for a clustering tournament selection) can be obtained from Equation 6 by replacing  $|S|$  with  $Q$ , and  $|S_j|$  with 1, and simplifying:

$$P(T_j) = \frac{(1 + Q - j)^r - (Q - j)^r}{Q^r} \quad (10)$$

Therefore, we obtain

$$P(W_j) = \frac{(1 + Q - j)^r - (Q - j)^r}{Q^r \times |S_j|} \quad (11)$$

where  $P(W_j)$  is the probability of selecting the  $j$ th cluster.

*Reproduction rate* is given a different definition in [1] from the common meaning in GP: it is defined as the ratio of the number of individuals with a certain fitness value in the mating pool after the whole selection process to the number before the selection. A reasonable selection method should favour good individuals by assigning them a reproduction rate greater than 1 and penalise bad individual by a ratio less than 1. This definition of reproduction rate can be interpreted as the frequency that individuals with a certain fitness value are used to produce offspring. Therefore, reproduction rate can be used to measure selection pressure

across population. We follow the formula in [1] and calculate the reproduction rate to compare the standard tournament selection with the clustering tournament selection.

We simulate two populations with 100 possible fitness values. One has a uniform fitness distribution: every unique fitness value maps to the same number of individuals. The other has a quadratic fitness distribution biased towards good fitness values so that good fitness values map to more individuals.

Figure 6 illustrates the reproduction rates using the standard tournament selection and the clustering tournament selections. It is clear that if a population has a uniform fitness distribution, the clustering tournament selection will have the same behaviour as the standard tournament selection. But in a population with a quadratic fitness distribution, where more individuals share the good fitness values, the clustering tournament selection significantly reduces the reproduction rate of high fitness programs, while increasing the rate of middle ranked programs. Therefore, the clustering tournament selection can reduce the chance that groups of high ranked programs dominate the next generation and it has better ability to maintain population diversity than the standard one.

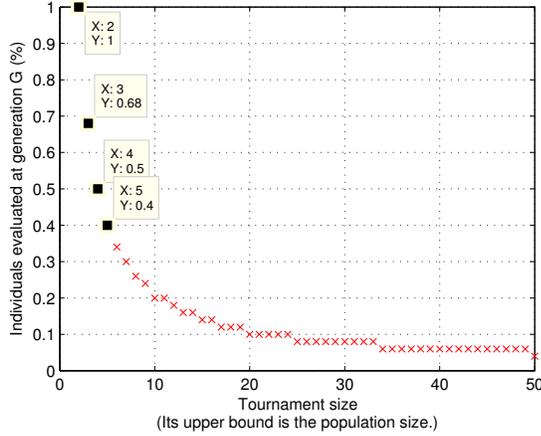
## 6. COMPUTATIONAL SAVINGS ON NOT-SAMPLED INDIVIDUALS

As mentioned in Section 2, BC-EA [9] is an interesting algorithm for obtaining computational savings by simply avoiding the creation and evaluation of not-sampled individuals<sup>1</sup>. However, there are several factors that may limit the effectiveness of the algorithm. One of them is the number ( $m$ ) of individuals that need to be evaluated at generation  $G$  in order to find an acceptable solution. If an acceptable solution is the first evaluated individual ( $m = 1$ ) in the population at generation  $G$ , significant computational saving can be obtained. But if  $m$  is a large number, requiring more samplings from generation  $G - 1$ , and consequently more evaluations at generation  $G - 1$ , then the effectiveness of the algorithm will be reduced.

### 6.1 Modelling the threshold of $m$

Let  $S$  be the population size and let  $r$  be the tournament size. Then the total number of samplings from generation

<sup>1</sup>We assume readers have had enough information about the algorithm.



**Figure 7: Threshold at generation  $G$  requires the evaluation of the whole population at generation  $G - 1$ .**

$G - 1$  is  $mr$ . Let  $u$  be the number of *unique* individuals in the  $mr$  samplings. Clearly, when  $m$  increases,  $u$  also increases. When  $m$  reaches a certain threshold, such that  $u = S$ , then the algorithm will not bring us any saving at all. Therefore, an important question is that what is the threshold of  $m$ .

For a given  $m$ ,  $u$  can vary from 1 to  $\max(u)$ , where

$$\max(u) = \begin{cases} S & , m \times r > S \\ m \times r & , m \times r \leq S \end{cases} \quad (12)$$

Let  $U^*(S, r, m, u)$  denote the number of different sampling events from generation  $G - 1$

$$U^*(S, r, m, u) = \begin{cases} S & , u = 1 \\ \binom{mr}{u} \binom{S}{u} u! & , 1 < u \leq \max(u) \end{cases} \quad (13)$$

Let  $U$  be the event that there are  $u$  unique individuals sampled from generation  $G - 1$  for a given  $m$  at generation  $G$ . The probability of this event is

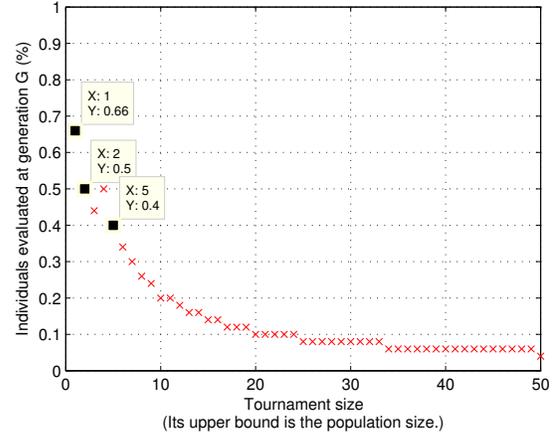
$$P(U) = \frac{U^*(S, r, m, u)}{\sum_{i=1}^{\max(u)} U^*(S, r, m, i)} \quad (14)$$

When  $u = x$ , such that  $P(U)$  returns the highest probability, it is most likely that there are  $x$  unique individuals sampled in  $mr$  samplings. Therefore, for  $m$  increased from 1 to  $S$ , we select  $u$  such that  $P(U)$  is the highest. If  $u = S$ , then the value of  $m$  will be the threshold. In other words, the value will be the maximum number of individuals that can be evaluated at generation  $G$  in order to have evaluation savings at generation  $G - 1$ .

## 6.2 Visualising the threshold of $m$

Figure 7 shows the number of individuals evaluated at generation  $G$  that requires the evaluation of the whole population at generation  $G - 1$  using different tournament size in a population of size 50.

The case of tournament size 1 is not shown in the figure. It means that no matter how many individuals at generation  $G$  will be evaluated, we will never need to evaluate the whole population at generation  $G - 1$ . The figure shows that, when the tournament size is 2, if the last individual in generation  $G$  is evaluated, the whole population at generation  $G - 1$  will be evaluated. When the tournament size is 3, we can evaluate up to 68% of the population at generation  $G$  in order



**Figure 8: Threshold at generation  $G$  requires the evaluation of the whole population excluding not-sampled individuals at generation  $G - 1$ .**

to have evaluation savings on generation  $G - 1$ . From the results, it seems that BC-EA is promising. However, as the whole population at generation  $G - 1$  has to be fully evaluated, there would be no further saving on generation  $G - 2$ , generation  $G - 3$ , ..., and generation 0, unless tournament size 1 is used. It is clear that, if tournament size 1 is used, the evolutionary algorithm will effectively act the same as a random (beam) search algorithm and may suffer long search time and low problem solving quality. In this case, it is not worth having such saving if we have to sacrifice the search time and the problem solving quality.

According to Equation 3, we compute that, with a population of size 50, the sampling probability of an individual after 50 tournaments is about 0.64, 0.87, and 0.95 when the tournament size is 1, 2, and 3 respectively. Let  $NS$  be the expected number of not-being sampled individuals. Then about 36%, 13% and 5% of the population is  $NS$  in the process of generating the next generation. Since  $NS$  should not be evaluated, we certainly should take these numbers into consideration to re-calculate the threshold. In other words, we would like to know what is the threshold of  $m$ , such that  $u = S - NS$ .

Figure 8 illustrates the number of individuals evaluated at generation  $G$  that requires the evaluation of  $S - NS$  individuals at generation  $G - 1$  using different tournament sizes in a population of size 50. The figure shows that when  $NS$  is considered, the threshold decreased significantly for tournament size 1, 2, and 3. Note that due to the number of not-sampled individuals is very small for tournament sizes greater than 3, there is no difference between Figures 7 and 8 from tournament size 4.

According to Figure 8, the threshold is about 66% of the population when the tournament size is 1. It means if an optimal solution is found within the first 66% of the population at generation  $G$ , we will have about 36% saving on the evaluation of generation  $G - 1$ . As the proportion of the population that needs to be evaluated at generation  $G - 1$  is 64%, which is less than 66%, we expect constant savings at all preceding generations, even with little growth. But for each of other tournament sizes, when  $m$  reaches the threshold, the savings at preceding generations will quickly drop because more and more individuals beyond the threshold have to be evaluated at each of the preceding generations.

### 6.3 A new fitness evaluation saving algorithm

Is there a way to avoid the limitations of  $m$  and tournament size but still obtain constant saving on the fitness evaluation at each generation? We propose a simple algorithm called *Evaluated-just-in-time* (Ejit) and expect it can guarantee the savings as long as there exist not-sampled individuals. Briefly, Ejit works in the following way:

1. create (all) programs at generation  $G - 1$  but *do not* evaluate them,
2. sample programs at generation  $G - 1$  for tournaments,
3. evaluate the sampled programs if they have not been evaluated, then select the winners as the parents of programs at generation  $G$ .

Although Ejit has to create every individual in a population, the creation time is in fact very little comparing with the time spent on fitness evaluations. Savings on creating not-sampled individuals are negligible thus it is fine to leave it out. Comparing with BC-EA, Ejit is expected to provide the following additional features: (1) constant savings by avoiding evaluations of not-sampled individuals; (2) no limitation on the maximum number of individuals that are evaluated at generation  $G$ ; (3) no need to limit the tournament size to be just 1; (4) no additional memory required; and (5) no need to put effort on choosing  $G$  in order to solve a problem.

### 7. CONCLUSIONS AND FUTURE WORK

We have provided several models of describing the probabilities that a program of a particular rank is sampled and is selected in the standard tournament selection process in a simple situation and a complex situation.

Under the assumption that the standard breeding process is used, we have demonstrated that with the same tournament size, the trends of sampling probability of a program and the selection probability distributions of a population are the same regardless of the population size. This observation suggests that the ratio of a tournament size and a population size may not be an issue in empirical studies on the effects of different tournament sizes in an evolutionary algorithm. Results from one reasonable population size should be applicable to other population sizes.

We have also demonstrated that with the same tournament size, the overall outline of a selection probability distribution in a complex situation is similar to that in a simple situation. This observation should reduce some complexities in further research of modellings in a complex situation.

We have also modelled the clustering tournament selection [13] to investigate its advantages in terms of maintaining population diversity.

We have further modelled and investigated a limitation of BC-EA [9]. In addition, we have proposed Ejit algorithm in order to have constant computation savings by avoiding the evaluation of not-sampled individuals.

Although this study is conducted in GP, the results are expected to be applicable to other evolutionary algorithms.

There are other different implementations at the fundamental level, including returning more than one individual per tournament [12] and sampling without replacement. We will further model and analyse existing alternative implementations in order to develop new selection strategies.

### Acknowledgment

We would like to thank Simon Doherty of School of Mathematics, Statistics and Computer Science at Victoria University of Wellington for his useful discussions, and the anonymous reviewers for valuable suggestions that helped to improve this paper.

### 8. REFERENCES

- [1] T. Blicke and L. Thiele. A mathematical analysis of tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 9–16. Morgan Kaufmann, 1995.
- [2] T. Blicke and L. Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, 1997.
- [3] V. Filipović, J. Kratica, D. Tošić, and I. Ljubić. Fine grained tournament selection for the simple plant location problem. In *5th Online World Conference on Soft Computing Methods in Industrial Applications*, pages 152–158, 2000.
- [4] G. R. Harik. Finding multimodal solutions using restricted tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31, San Francisco, CA, 1995. Morgan Kaufmann.
- [5] J. R. Koza. *Genetic Programming — On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, 1992.
- [6] S. Luke and L. Panait. Lexicographic parsimony pressure. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 829–836. Morgan Kauffman, 2002.
- [7] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. Technical Report 95006, University of Illinois at Urbana-Champaign, July 1995.
- [8] T. Motoki. Calculating the expected loss of diversity of selection schemes. *Evolutionary Computation*, 10(4):397–422, 2002.
- [9] R. Poli. Tournament selection, iterated coupon-collection problem, and backward-chaining evolutionary algorithms. In *Foundations of Genetic Algorithms*, volume 3469 of *Lecture Notes in Computer Science*, pages 132–155. Springer-Verlag, 2005.
- [10] R. Poli and W. B. Langdon. Backward-chaining genetic programming. In H.-G. Beyer et al, editors, *Proceedings of the 2005 conference on Genetic and evolutionary computation GECCO 2005*, volume 2, pages 1777–1778, Washington DC, USA, 25-29 June 2005. ACM Press.
- [11] A. Sokolov and D. Whitley. Unbiased tournament selection. In *Proceedings of the 2005 conference on Genetic and Evolutionary Computation*, pages 1131–1138. ACM Press, 2005.
- [12] T. Weinbrenner. GPC++ - Genetic Programming C++ Class Library, 1997.
- [13] H. Xie, M. Zhang, and P. Andreae. Population clustering in genetic programming. In *Proceedings of the 9th European Conference, EuroGP 2006*, volume 3905 of *LNCS*, pages 190–201. Springer-Verlag, 2006.