

Impacts of sampling strategies in tournament selection for genetic programming

Huayang Xie · Mengjie Zhang

Published online: 16 September 2011
© Springer-Verlag 2011

Abstract Tournament selection is one of the most commonly used parent selection schemes in genetic programming (GP). While it has a number of advantages over other selection schemes, it still has some issues that need to be thoroughly investigated. Two of the issues are associated with the sampling process from the population into the tournament. The first one is the so-called “multi-sampled” issue, where some individuals in the population are picked up (sampled) many times to form a tournament. The second one is the “not-sampled” issue, meaning that some individuals are never picked up when forming tournaments. In order to develop a more effective selection scheme for GP, it is necessary to understand the actual impacts of these issues in standard tournament selection. This paper investigates the behaviour of different sampling replacement strategies through mathematical modelling, simulations and empirical experiments. The results show that different sampling replacement strategies have little impact on selection pressure and cannot effectively tune the selection pressure in dynamic evolution. In order to conduct effective parent selection in GP, research focuses should be on developing automatic and dynamic selection pressure tuning methods instead of alternative sampling replacement

strategies. Although GP is used in the empirical experiments, the findings revealed in this paper are expected to be applicable to other evolutionary algorithms.

Keywords Tournament selection · Selection pressure · Genetic programming

1 Introduction

Genetic programming (GP) (Koza 1992), one of the metaheuristic search methods in evolutionary algorithms (EAs) (Eiben and Smith 2003), is based on the Darwinian natural selection theory. Its special characteristics make it an attractive learning or search algorithm for many real-world problems, including signal filters (Andreae et al. 2008; Brameier et al. 2001), circuit designing (de Sa and Mesquita 2008; Koza et al. 1999; Popp et al. 1998), image recognition (Agnelli et al. 2002; Akyol et al. 2007; Vanyi 2005), symbolic regression (Castillo et al. 2006; Schmidt and Lipson 2007; Smits et al. 2005), financial prediction (Lee 2006; Li and Tsang 2000; Zhang et al. 2004), and classification (Espejo et al. 2010; Hong and Cho 2004; Zhang et al. 2003, 2006).

Selection is an important aspect in EAs. Although “survival of the fittest” has driven EAs since the 1950s and many selection methods have been developed, how to effectively select parents still remains an important open issue.

Commonly used parent selection schemes in EAs include fitness proportionate selection (Holland 1975), ranking selection (Grefenstette and Baker 1989), and tournament selection (Brindle 1981). To determine which parent selection scheme is suitable for a particular paradigm, three factors need to be considered. The first factor is

H. Xie (✉) · M. Zhang
School of Engineering and Computer Science, Victoria
University of Wellington, Wellington, New Zealand
e-mail: hxie@ecs.vuw.ac.nz

M. Zhang
e-mail: mengjie@ecs.vuw.ac.nz

H. Xie
Department of Information and Computer Science, Anhui
Polytechnic University, Anhui, People's Republic of China

whether the selection pressure¹ of a selection scheme can be changed easily since it directly affects the convergence of learning. The second is whether a selection scheme supports parallel architectures since a parallel architecture is very useful for speeding up learning paradigms that are computationally intensive. The third factor is whether the time complexity of a selection scheme is low since the running cost of the selection scheme can be amplified by the number of individuals involved.

Tournament selection randomly draws/samples k individuals with or without replacement from the current population of size N into a tournament of size k and selects the one with the best fitness from the tournament. In general, selection pressure in tournament selection can be easily changed by using different tournament sizes; the larger the tournament size, the higher the selection pressure. Drawing individuals with replacement into a tournament makes the population remain unchanged, which in turn allows tournament selection to easily support parallel architectures. Selecting the winner involves simply finding the best out of k individuals; thus the time complexity of a *single* tournament is $O(k)$. Furthermore, in general, since the standard breeding process in GP produces one offspring by applying mutation to one parent and produces two offspring by applying crossover to two parents, the total number of tournaments needed to generate the entire next generation is N . Therefore, the time complexity of tournament selection is $O(kN)$.

GP is recognised as a computationally intensive method, often requiring a parallel architecture to improve its efficiency. Furthermore, it is not uncommon to have millions of individuals in a population when solving complex problems (Koza et al. 2003); thus, sorting a whole population is time consuming. The support of parallel architecture and the linear time complexity have made tournament selection very popular in GP and the sampling-with-replacement tournament selection has become the standard tournament selection (STS) scheme in GP. The literature includes many studies on the STS (Back 1994; Blickle and Thiele 1995, 1997; Branke et al. 1996; Goldberg and Deb 1991; Miller and Goldberg 1995, 1996; Motoki 2002; Poli and Langdon 2006).

Although STS is very popular in GP, it still has some open questions. For instance, because individuals are sampled with replacement, it is possible to have the same individual sampled multiple times in a tournament (the multi-sampled issue). It is also possible to have some individuals not sampled at all when using small tournament sizes (the not-sampled issue). These two issues may lower the probability of some good individuals being sampled or

selected. Additionally, they may aggravate premature convergence and loss of population diversity (Lima et al. 2007; Sokolov and Whitley 2005), which might in turn affect the system performance of EAs (Gustafson 2004). However, such views have not been thoroughly investigated. In addition, although it seems that the selection pressure can be easily changed using different tournament sizes to influence the convergence of the genetic search process, two problems exist during population convergence: (1) when groups of programs have the same or similar fitness values, the selection pressure between groups increases regardless of the given tournament size, resulting in “better” groups dominating the next population and possibly causing premature convergence; and (2) when most programs have the same fitness value, the selection behaviour effectively becomes random.² Therefore, tournament size itself is not always adequate for controlling selection pressure. Furthermore, the evolutionary learning process itself is very dynamic, requiring adapting selection pressure during an EA run (de Jong 2007). For instance, from our experimental studies we realised that at some stages, it requires a fast convergence rate (i.e., high parent selection pressure) to find a solution quickly; at other stages, it requires a slow convergence rate (i.e., low parent selection pressure) to avoid being confined to a local optimum. However, STS does not fulfill the adaptation requirements. There exists a strong demand to clarify the open issues of STS in order to conduct an effective selection process in GP. To do that, a thorough investigation of tournament selection is necessary.

This paper aims to clarify whether the two sampling behaviour-related issues are critical in STS and to determine whether further research should focus on developing alternative sampling strategies in order to conduct effective selection processes in GP.

Section 2 gives a review of selection pressure measurements. Section 3 presents the necessary assumptions and definitions. Section 4 shows the selection behaviour in STS, providing a baseline for investigating the multi-sampled and not-sampled issues. Sections 5 and 6 analyse the impacts of the multi-sampled and the not-sampled issues via modelling and simulations, respectively. Section 7 discusses the evolutionary dynamics of the tournament selection schemes. Section 8 investigates the two issues via experiments and Sect. 9 concludes this paper.

2 Selection pressure measurements

A critical issue in designing a selection technique is selection pressure, which has been widely studied in EAs

¹ It is the degree to which the better individuals are favoured (Miller and Goldberg 1995).

² Other selection schemes may also suffer this problem.

(Affenzeller et al. 2005; Blickle and Thiele 1995; Goldberg and Deb 1991; Miller and Goldberg 1995; Motoki 2002; Winkler et al. 2008). Many definitions of selection pressure can be found in the literature. For instance, it is defined as (1) the intensity with which an environment tends to eliminate an organism and thus its genes, or gives it an adaptive advantage; (2) the impact of effective reproduction due to environmental impact on the phenotype; and (3) the intensity of selection acting on a population of organisms or cells in culture. These definitions originate from different perspectives but they share the same aspect, which can be summarised as the degree to which the better individuals are favoured (Miller and Goldberg 1995). Selection pressure gives individuals of higher quality a higher probability of being used to create the next generation, so that EAs can focus on promising regions in the search space (Blickle and Thiele 1995).

Selection pressure controls the selection of individual programs from the current population to produce a new population of programs in the next generation. It is important in a genetic search process because it directly affects the population convergence rate. The higher the selection pressure, the faster the convergence. A fast convergence decreases learning time, but often results in a GP learning process being confined in a local optimum or “premature convergence” (Ciesielski and Mawhinney 2002; Koza 1992). A low convergence rate generally not only decreases the chance of premature convergence but also increases the learning time, and may not be able to find an optimal or acceptable solution in a predefined limited time.

In tournament selection, the mating pool consists of tournament winners. The average fitness in the mating pool is usually higher than that in the population. The fitness difference between the mating pool and the population reflects the selection pressure, which is expected to improve the fitness of each subsequent generation (Miller and Goldberg 1995).

In biology, the effectiveness of selection pressure can be measured in terms of differential survival and reproduction and consequently in change in the frequency of alleles in a population. In EAs, there are several measurements for selection pressure in different contexts, including takeover time, selection intensity, loss of diversity, reproduction rate, and selection probability distribution.

Takeover time is defined as the number of generations required to completely fill a population with just copies of the best individual in the initial generation when available operators are limited to only selection and copy operators (Goldberg and Deb 1991). For a given fixed-sized population, the longer the takeover time, the lower the selection pressure. (Goldberg and Deb 1991) estimated the takeover time for STS as

$$\frac{1}{\ln k} (\ln N + \ln(\ln N)) \tag{1}$$

where N is the population size and k is the tournament size. The approximation improves when $N \rightarrow \infty$.

Selection intensity was first introduced in the context of population genetics to obtain a normalised and dimensionless measure (Bulmer 1980), and, later was adopted and applied to GAs (Muhlenbein and Schlierkamp-Voosen 1993). Blickle and Thiele (1995, 1997) measured it using the expected change of the average fitness of the population. As the measurement is dependent on the fitness distribution in the initial generation, they assumed the fitness distribution followed the normalised Gaussian distribution and introduced an integral equation for modelling selection intensity in STS.

For their model, analytical evaluation can be done only for small tournament sizes and numerical integration is needed for large tournament sizes. The model is not valid in the case of discrete fitness distributions. In addition to these limitations, the assumption that the fitness distribution followed the normalised Gaussian distribution is not valid in general (Popovici and de Jong 2003). Furthermore, the model is of limited use because tournament selection ignores the actual fitness values and uses the relative rankings instead.

Loss of diversity is defined as the proportion of individuals in a population that are not selected during a parent selection phase by Blickle and Thiele (1995, 1997). For STS, they estimate it to be

$$k^{-\frac{1}{k-1}} - k^{-\frac{k}{k-1}} \tag{2}$$

However, Motoki (2002) pointed out that Blickle and Thiele’s estimation of the loss of diversity in tournament selection does not follow their definition, and indeed their estimation is of loss of fitness diversity. Motoki recalculated the loss of program diversity in a wholly diverse population, i.e., every individual has a distinct fitness value, on the assumption that the worst individual is ranked 1st, as

$$\frac{1}{N} \sum_{j=1}^N (1 - P(W_j))^N \tag{3}$$

where $P(W_j) = \frac{j^k - (j-1)^k}{N^k}$ is the probability that an individual of rank j is selected in a tournament.

Reproduction rate is defined as the ratio of the number of individuals with a certain fitness f after and before selection (Blickle and Thiele 1995, 1997). A reasonable selection method should favour good individuals by giving them a high ratio and penalise bad individuals by giving a low ratio. Branke et al. (1996) introduced a similar measure which is the expected number of selections of an

individual. It is calculated by multiplying the total number of tournaments N conducted in a parent selection phase by the selection probability of the individual in a single tournament $P(W_j)$:

$$N \times P(W_j) \quad (4)$$

This measure is termed *selection frequency* in this paper hereafter as *reproduction* has another meaning in GP.

Selection probability distribution of a population at a generation is defined as consisting of the probabilities of each individual in the population being selected at least once in a parent selection phase (Xie et al. 2007). Although tournaments indeed can be implemented in a parallel manner, in Xie et al. (2007) they are assumed to be conducted sequentially so that the number of tournaments conducted reflects the progress of generating the next generation. As a result, the selection probability distribution can be illustrated in a three-dimensional graph, where the x -axis shows every individual in the population ranked by fitness (the worst individual is ranked 1st), the y -axis shows the number of tournaments conducted in the selection phase (from 1 to N), and the z -axis is the selection probability which shows how likely a given individual marked on x -axis can be selected at least once after a given number of tournaments marked on y -axis. Therefore, the measure provides a full picture of the selection behaviour over the population during the whole parent selection phase. Figure 1 shows the selection probability distribution measure for STS of tournament size 4 on a wholly diverse population of size 40.

3 Assumptions and definitions

To model and simulate selection behaviours in tournament selection, we make a number of assumptions and definitions in this section.

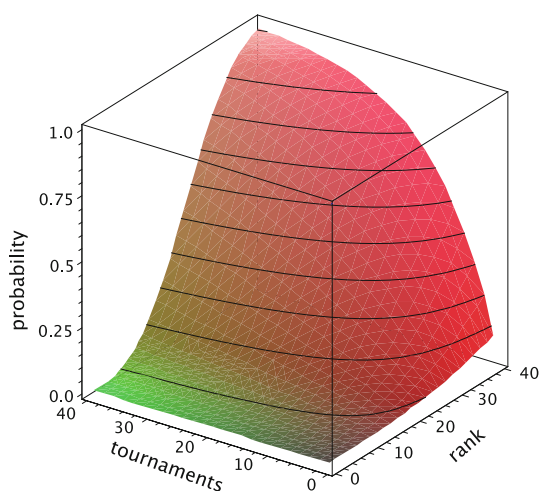


Fig. 1 An example of the selection probability distribution measure

A population can be partitioned into bags consisting of programs with equal fitness. These “fitness bags” may have different sizes. As each fitness bag is associated with a distinct fitness rank, we can characterise a population by the number of distinct fitness ranks and the size of each corresponding fitness bag, which we term *fitness rank distribution* (FRD). If S is the population, then we used the notation N to be the size of the population, S_j to be the bag of programs with the fitness rank j and $|S_j|$ to be the size of the bag, and $|S|$ to be the number of distinct fitness bags. We denoted tournament size by k and ranked the program with the worst fitness 1st. We followed the standard breeding process so that the total number of tournaments is N at the end of generating all individuals in the next generation.

In order to make the results of the selection behaviour analysis easily understandable, we assumed that tournaments were conducted sequentially. We chose only the loss of program diversity, the selection frequency, and the selection probability distribution measures for the selection behaviour analysis and ignored the takeover time and the selection intensity due to their limitations.

We used three populations with different FRDs, namely *uniform*, *reversed quadratic*, and *quadratic*, in our simulations. The three FRDs are designed to mimic the three stages of evolution but by no means to model all the real situations happening in a true run of evolution. The uniform FRD represents the initialisation stage, where each fitness bag has a roughly equal number of programs. A typical case of the uniform FRD can be found in a wholly diverse population. The reversed quadratic FRD represents the early evolving stage, where commonly very few individuals have good fitness values. The quadratic FRD represents the later stage of evolution, where a large number of individuals have converged to better fitness values.

Since the impact of population size on selection behaviour is unclear, we tested several different commonly used population sizes, ranging from small to large. This paper illustrates only the representative results of the uniform FRD with a population of size 40, and the quadratic and the reversed quadratic FRDs with populations of size 2000. Note that although the populations with different FRDs are of different sizes, the number of distinct fitness ranks is designed to be the same value (i.e. 40) for easy visualisation and comparison purposes (see Fig. 2). We also studied and visualised other different numbers of distinct fitness ranks (100, 500 and 1000) and obtained similar results (these results are not shown in the paper).

Furthermore, for the selection frequency and the selection probability distribution measures, we chose three different tournament sizes (2, 4, and 7) commonly used in the literature, to illustrate how tournament size affects the selection behaviour.

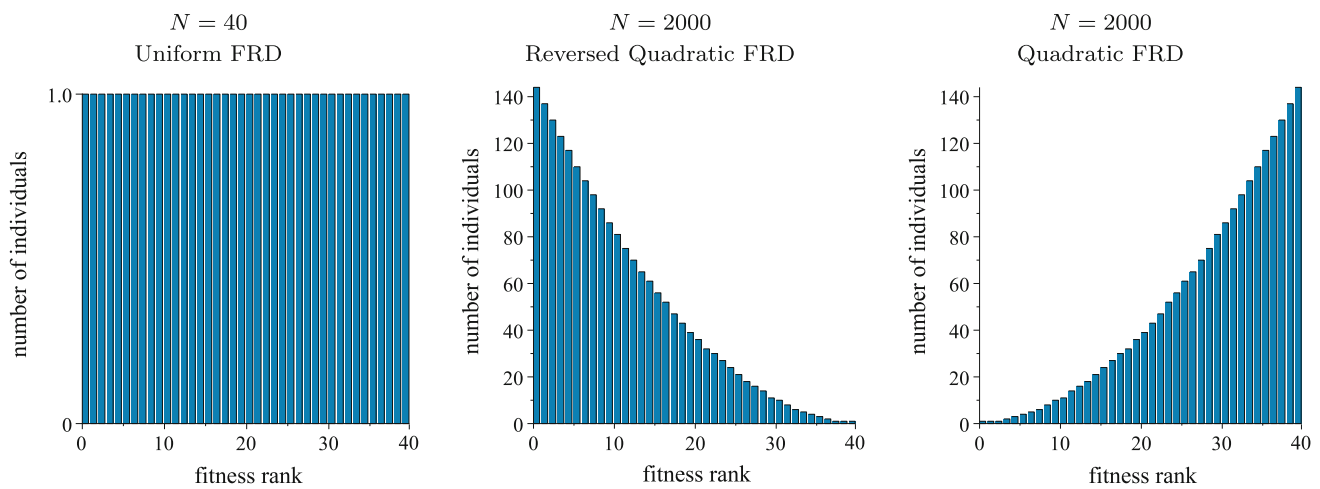


Fig. 2 Three populations with different fitness rank distributions

4 Selection behaviour in standard tournament selection

In order to make a valid comparison when investigating the multi-sampled and not-sampled issues, it is essential to show the selection behaviour in STS using the same set of measurements and simulation methods.

From Xie et al. (2007), the probability of an event that any program p is sampled at least once in $y \in \{1, \dots, N\}$ tournaments is

$$1 - \left(\left(\frac{N-1}{N} \right)^N \right)^{\frac{y}{N^k}} \tag{5}$$

According to Eq. 5, we calculate the probability trends of a single program being sampled at least once using six different tournament sizes (1, 2, 4, 7, 20, and 40) in three populations of sizes 40, 400, and 2000 (shown in Fig. 3). The figure shows that the larger the tournament size, the higher the sampling probability. Furthermore, for a given tournament size, the trends of sampling probabilities of a program in the selection phase (along the increments of the number of tournaments) are very similar in different-sized populations.

From Xie et al. (2007), the probability of an event W_j that a program $p \in S_j$ is selected from a tournament is

$$P(W_j) = \frac{\left(\frac{\sum_{i=1}^j |S_i|}{N} \right)^k - \left(\frac{\sum_{i=1}^{j-1} |S_i|}{N} \right)^k}{|S_j|} \tag{6}$$

We then calculate the total loss of program diversity using Eq. 3 in which $P(W_j)$ is replaced by Eq. 6. We also split the total loss of program diversity into two parts. One part is from the fraction of the population that is not sampled at all during the selection phase. We calculate it also using Eq. 3 by replacing $1 - P(W_j)$ with $\left(\frac{N-1}{N} \right)^k$,

which is the probability that an individual has not been sampled in a tournament of size k . The other part is from the fraction of population that is sampled but never selected. We calculate it by taking the difference between the total loss of program diversity and the contribution from not-sampled individuals.

Figure 4 shows the three loss of program diversity measures, namely the total loss of program diversity and the contributions from not-sampled³ and not-selected⁴ individuals for STS on the three populations with different FRDs. Overall there were no noticeable differences between the three loss of program diversity measures on the three different populations with different FRDs.

For each of the three populations with different FRDs, we also calculate the expected selection frequency of a program in the selection phase based on Eq. 4 using the probability model of a program being selected in a tournament (Eq. 6). Figure 5 shows the selection frequency in STS on the three populations with different FRDs. Instead of plotting the expected selection frequency for every individual, we plot it only for an individual in each of the 40 unique fitness ranks so that plots in different-sized populations have the same scale and it is easy to identify what fitness ranks may be lost. From the figure, not surprisingly, overall STS favours better-ranked individuals for all tournament sizes, and the selection pressure is biased towards better individuals as the tournament size increases. Furthermore, skewed FRDs (reversed quadratic and quadratic) aggravate selection bias quite significantly.

³ It refers to individual programs that have never participated into any tournament in a parent selection phase.

⁴ It refers to individual programs that have participated into tournaments but have never won any tournament, and accordingly have not been selected for mating.

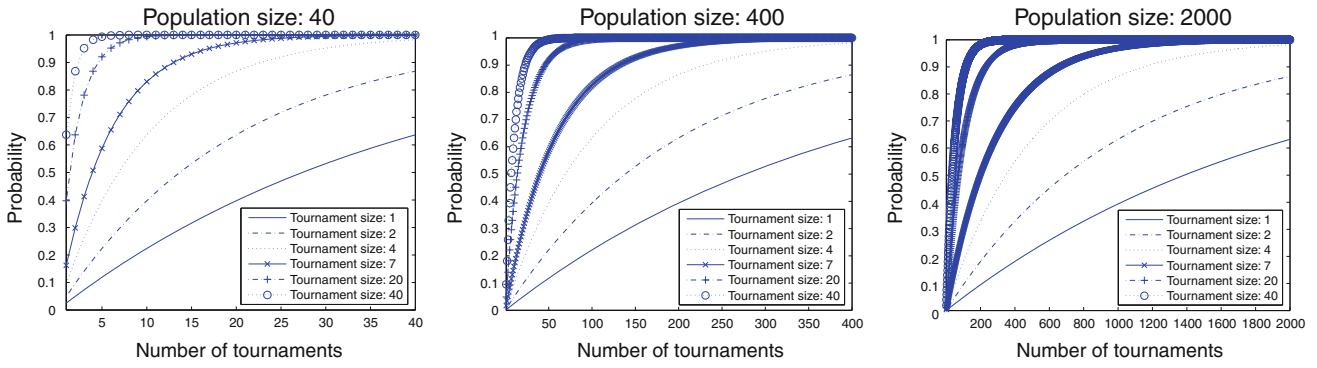


Fig. 3 Trends of the probability that a program is sampled at least once in STS in the parent selection phase (note that the scales on the x-axes differ)

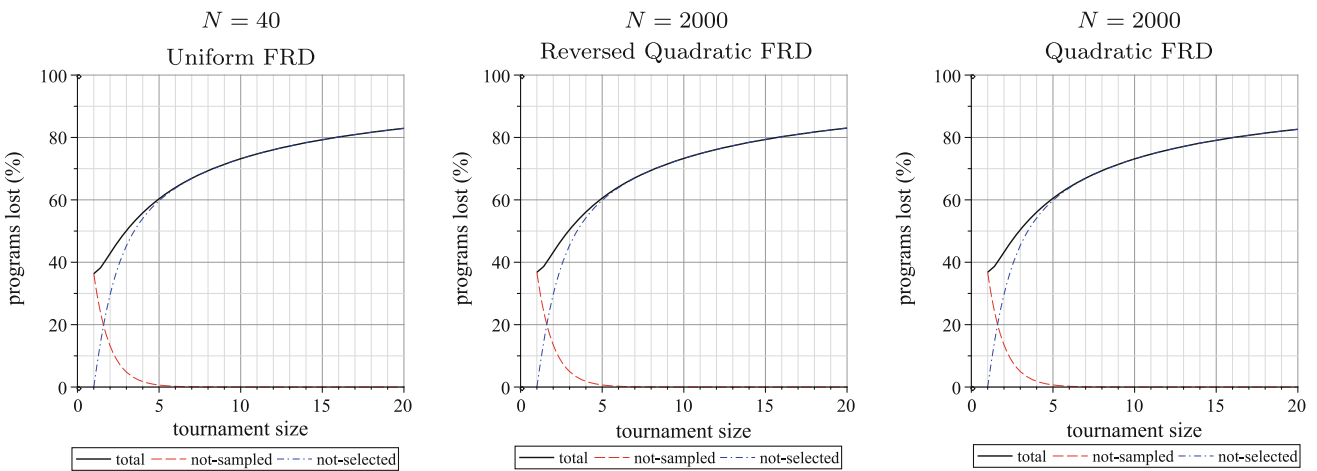


Fig. 4 Loss of program diversity in STS on three populations with different FRDs. Note that the tournament size is discrete but the plots show curves to aid interpretation

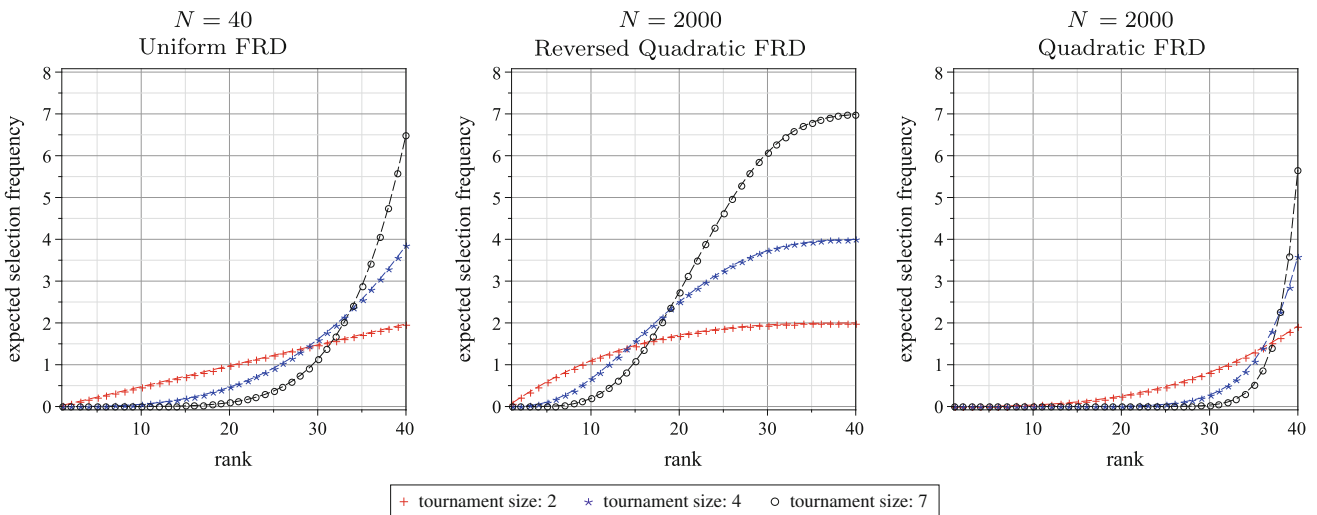


Fig. 5 Selection frequency in STS on three populations with different FRDs

From Xie et al. (2007), the probability that a program p of rank j is selected at least once in $y \in \{1, \dots, N\}$ tournaments is

$$1 - (1 - P(W_j))^y \tag{7}$$

where $P(W_j)$ is the probability of a program being selected from a tournament (see Eq. 6).

We finally calculate the selection probability distribution based on Eq. 7. Figure 6 illustrates the selection probability distribution using the three different tournament sizes (2, 4, and 7) on the three populations with

different FRDs. Again, we plot it for each of the 40 unique individual ranks. Clearly, different tournament sizes have a different impact on the selection pressure. The larger the tournament size, the more the selection pressure favours individuals of better ranks. For the same tournament size, the same population size but different FRDs (i.e. the second and the third rows in Fig. 6) result in different selection probability distributions.

From additional visualisations on other-sized populations with the three FRDs, we observed that similar FRDs but different population sizes result in similar selection

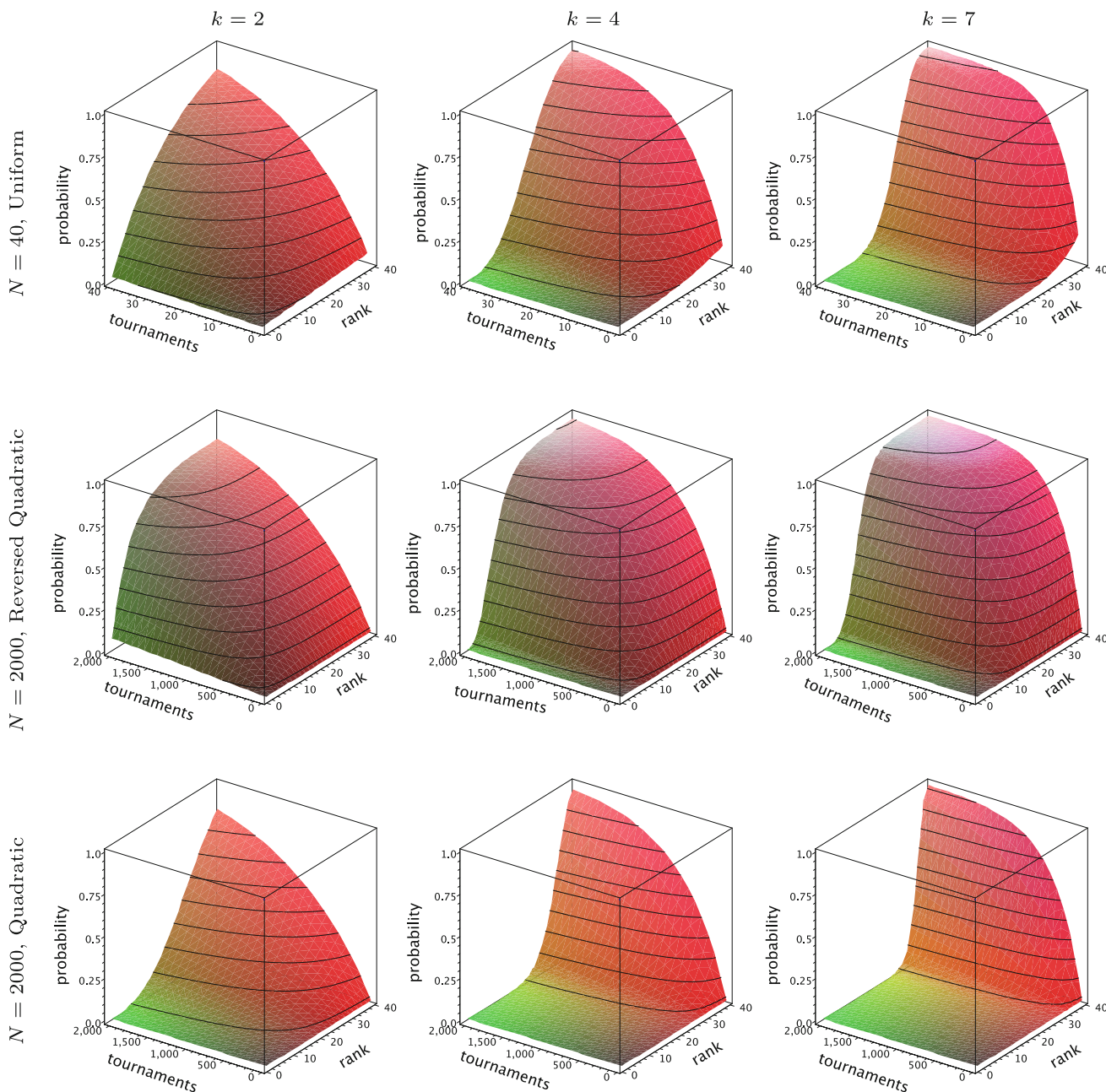


Fig. 6 Selection probability distribution in STS with tournament size 2, 4, and 7 on three populations with different FRDs

probability distributions, indicating that population size does not significantly influence the selection pressure. Note that in general the genetic material differs between populations of different sizes, and the impact of genetic material in different-sized populations on GP performance varies significantly. However, understanding that impact is another research topic and is beyond the scope of this paper.

5 Analysis of the multi-sampled issue via simulations

As mentioned earlier, the impact of the multi-sampled issue was unclear. This section shows that the multi-sampled issue is not a serious problem. This is done by analysing the *no-replacement* tournament selection scheme (NRTS), which removes the multi-sampled issue. It then compares NRTS to STS, showing there is no significant difference between them from the perspective of the metrics used.

5.1 No-replacement tournament selection

NRTS samples individuals into a tournament but does not return the sampled individuals back to the population immediately; thus, no individual can be sampled multiple times into the same tournament. After the winner is determined, it then returns all individuals of the tournament to the population. According to Goldberg and Deb (1991), NRTS was introduced at the same time as STS. However, NRTS is less commonly used in EAs.

5.2 Modelling no-replacement tournament selection

The only factor making NRTS different from the standard one is that any individual in a population will be sampled at most once in a single tournament and will have k chances to be drawn out of the population N . Therefore, if D is the event that an arbitrary program p is drawn or sampled in a tournament of size k , the probability of D is

$$P(D) = \frac{k}{N} \quad (8)$$

If I_y is the event that p is drawn or sampled at least once in $y \in \{1, \dots, N\}$ tournaments, the probability of I_y is

$$\begin{aligned} P(I_y) &= 1 - (1 - P(D))^y = 1 - \left(1 - \frac{k}{N}\right)^y \\ &= 1 - \left(\frac{N-k}{N}\right)^{Ny} \end{aligned} \quad (9)$$

Lemma 1 For a particular program $p \in S_j$, if $E_{j,y}$ is the event that p is selected at least once in $y \in \{1, \dots, N\}$ tournaments, the probability of $E_{j,y}$ is:

$$\begin{aligned} P(E_{j,y}) &= 1 - \left(1 - \frac{1}{|S_j|} \left(\frac{\binom{\sum_{i=1}^j |S_i|}{k}}{\binom{N}{k}} - \frac{\binom{\sum_{i=1}^{j-1} |S_i|}{k}}{\binom{N}{k}} \right)\right)^y \end{aligned} \quad (10)$$

Proof The probability that all the programs sampled for a tournament have a fitness rank between 1 and j (i.e. are from S_1, \dots, S_j) is given by

$$\frac{\binom{\sum_{i=1}^j |S_i|}{k}}{\binom{N}{k}}$$

If T_j is the event that the best-ranked program in a tournament is from S_j , the probability of T_j is

$$P(T_j) = \frac{\binom{\sum_{i=1}^j |S_i|}{k}}{\binom{N}{k}} - \frac{\binom{\sum_{i=1}^{j-1} |S_i|}{k}}{\binom{N}{k}}. \quad (11)$$

Let W_j be the event that the program $p \in S_j$ is selected in a tournament. As each element of S_j has equal probability of being selected in a tournament, the probability of W_j is

$$P(W_j) = \frac{P(T_j)}{|S_j|}. \quad (12)$$

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

$$P(E_{j,y}) = 1 - (1 - P(W_j))^y. \quad (13)$$

Substituting for $P(W_j)$ we obtain Eq. 10. \square

For the special simple situation that all individuals have distinct fitness values, $|S_j|$ becomes 1. Substituting this into Eqs. 11 and 12, we obtain the following equation, which is identical to the model presented in Branke et al. (1996).

$$P(W_j) = \frac{\binom{j}{k} - \binom{j-1}{k}}{\binom{N}{k}} \quad (14)$$

5.3 Selection behaviour analysis

The loss of program diversity, the selection frequency, and the selection probability distribution for NRTS are calculated by substituting Eq. 12 into Eqs. 3, 4 and 7, and illustrated in Figs. 7, 8, and 9, respectively. Comparison results of these figures and Figs. 4, 5 and 6 show that the selection behaviour in NRTS is almost identical to that in STS.

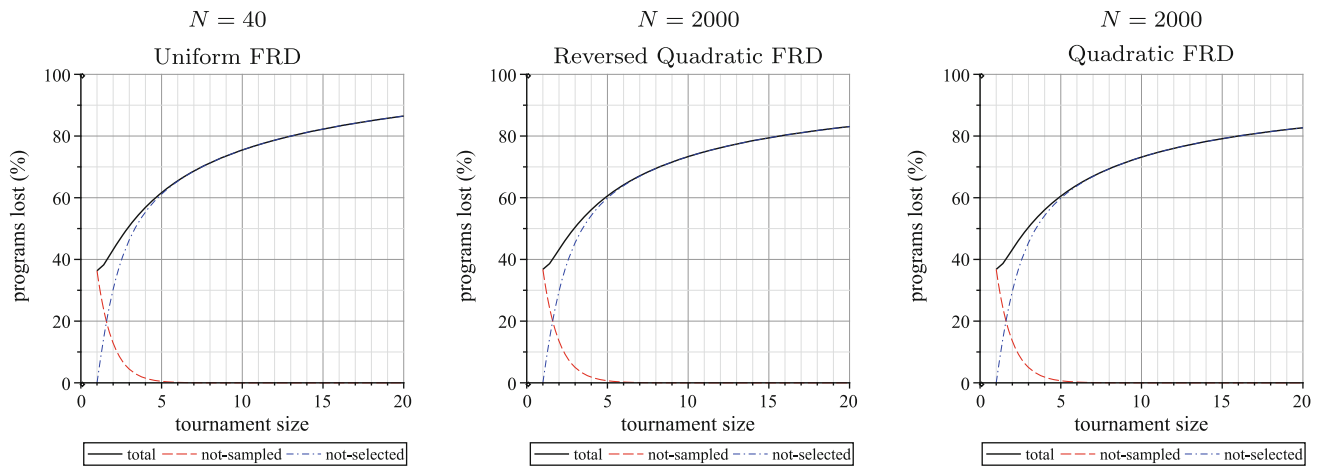


Fig. 7 Loss of program diversity in NRTS on three populations with different FRDs. Note that tournament size is discrete but the plots show curves to aid interpretation

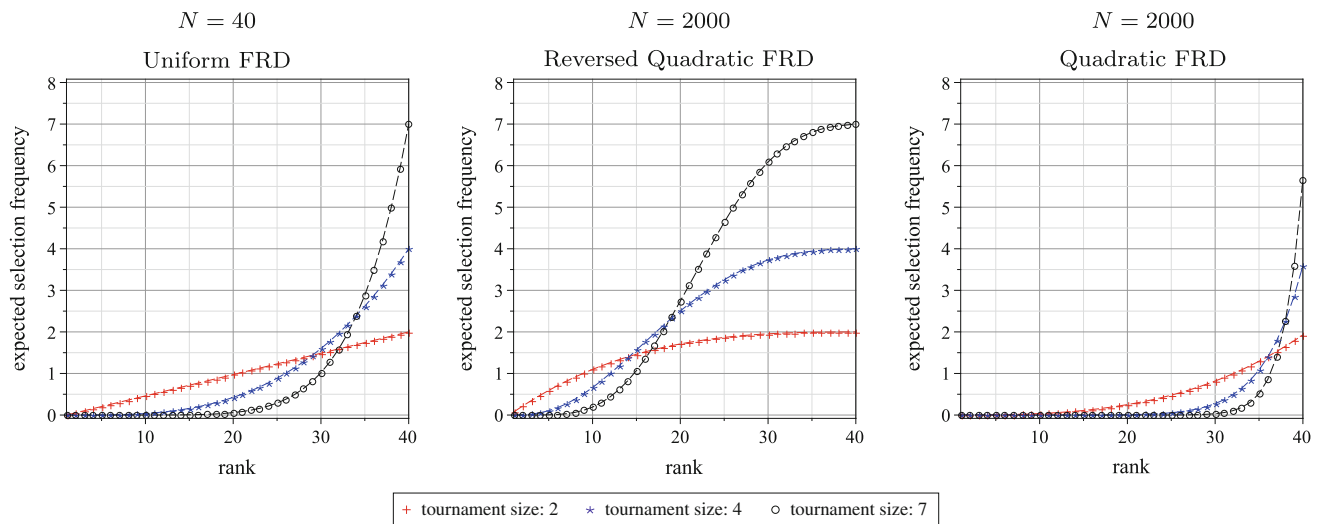


Fig. 8 Selection frequency in NRTS on three populations with different FRDs

With a closer inspection of the total loss of program diversity measure, we observed that when large tournament sizes (such as $k > 13$) are used, a higher total program lost occurs in NRTS on the small-sized population ($N = 40$), whereas no noticeable difference exists on the other sized populations. A possible explanation is that in NRTS, according to Eq. 9, the probability that a program has never been sampled in $y = N$ tournaments for large N/k is

$$\left(\frac{N-k}{N}\right)^N = \left(\frac{N-k}{k}\right)^{\frac{N}{k}} \approx e^{-k}. \tag{15}$$

This equation is approximately the same as that (derived from Eq. 5) in STS. However, for the smaller sized population when larger tournament sizes are used, this approximation is not valid. Therefore, the no-replacement

strategy does not help the loss of program diversity, especially when the size of a population is large.

Similar observations can be obtained by comparing the other two selection pressure measures. The results show that if common tournament sizes (such as $k = 4$ or 7) and population sizes (such as $N > 100$) are used, no significant difference in selection behaviour has been observed between STS and NRTS. The next subsection examines the sampling behaviour to explore the underlying reasons.

Note that overall there were no noticeable differences between the three loss of program diversity measures on the three different populations with different FRDs. The loss of program diversity measure depends almost entirely on the tournament size, and is almost independent of the FRD, whilst the other two measures can reflect the changes in FRDs. The loss of program diversity measure cannot

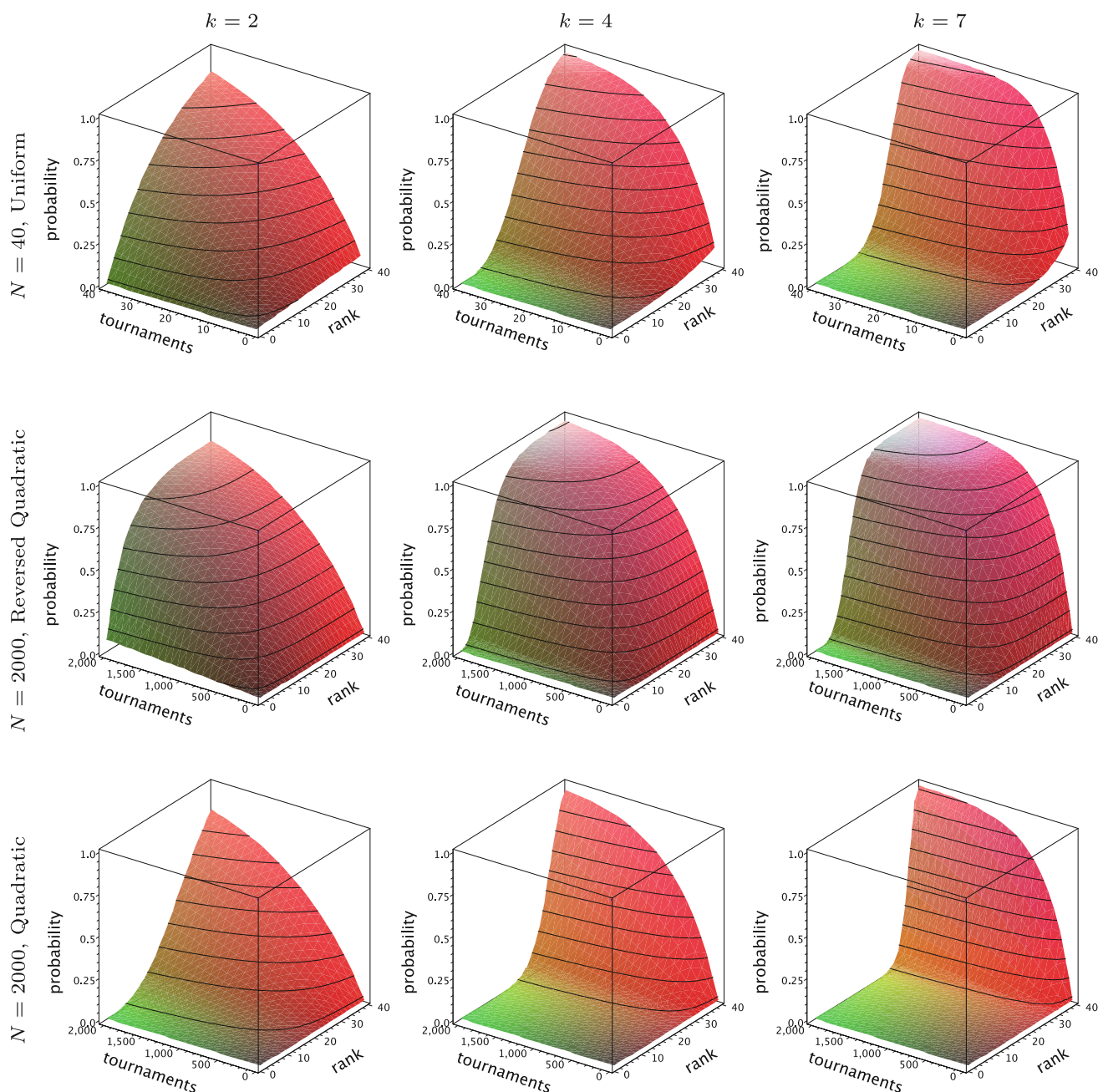


Fig. 9 Selection probability distribution in NRTS with tournament size 2, 4, and 7 on three populations with different FRDs

capture the effect of different FRDs, implying that it is not an adequate measure of selection pressure.

5.4 Sampling behaviour analysis

Figure 10 demonstrates the sampling behaviour in NRTS via the probability trends of a program being sampled using six tournament sizes in three populations as the number of tournaments increases up to the corresponding population size. By comparing Figs. 10 and 3, apart from the case of population size 40 and tournament size 40, which produces

the 100% sampling probability in NRTS, there are no noticeable differences between corresponding trends in the standard and no-replacement tournament selection schemes. The results are not surprising since both Eqs. 5 and 9 can be approximated by $1 - e^{-k^2/N}$ for large N .

5.5 Confidence analysis

To further investigate the similarity or difference between the sampling behaviour in STS and NRTS, we ask the following question: for a given population of size N , if we

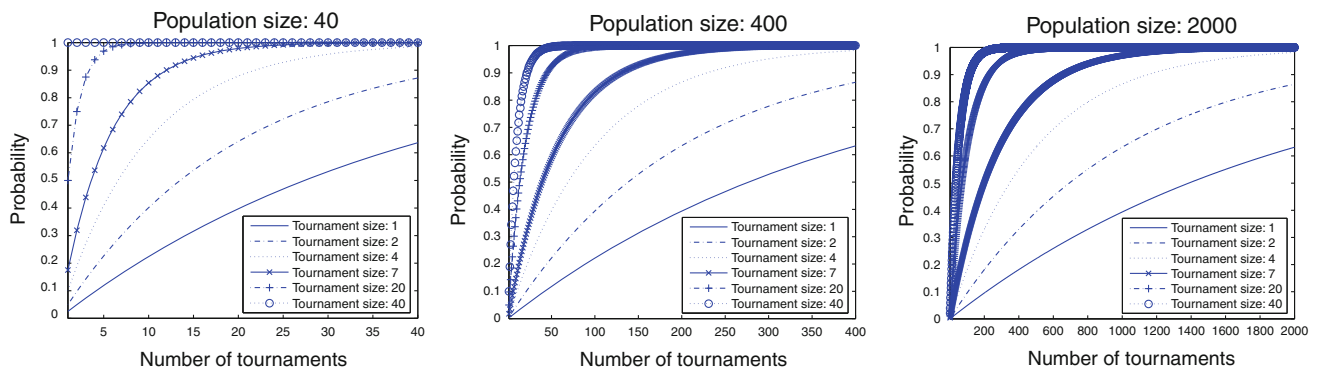


Fig. 10 Trends of the probability that a program is sampled at least once in NRTS in the selection phase (note that the scales on the x-axes differ)

keep sampling individuals with replacement, then what is the largest number of sampling events at a certain level of confidence that there will be no duplicates amongst the sampled individuals? Answering this question requires an analysis of the relationship between confidence level, population size, and tournament size. Equation 16 models the relationship between the three factors, where N^k is the total number of different sampling results when sampling k individuals with replacement, $\frac{N!}{(N-k)!}$ is the number of sampling events such that no duplicate is in the k sampled individuals, and $(1 - \alpha)$ is the confidence coefficient⁵

$$\frac{N!}{N^k(N - k)!} \geq 1 - \alpha. \tag{16}$$

Figure 11 illustrates the relationship between population size N , tournament size k , and the confidence level. For instance, sampling 7 individuals with replacement will not sample duplicates with 99% confidence when the population size is about 2000, and with 95% confidence when the population size is about 400, but with only 90% confidence when the population size is about 200. We also calculated that when the population size is 40, the confidence level is only about 57% for $k = 7$. These results explained why we have observed differences only between STS and NRTS on the very small-sized population using relatively large tournament sizes.

The results show that for tournament size 4 or less, we would not expect to see any duplicates except for very small populations. Even for tournament size 7, we would expect only to see a small number of duplicates for populations <200 with 90% confidence. For most common and reasonable settings of tournament sizes and population sizes, the multi-sampled event *seldom* occurs in STS. In addition, since duplicated individuals do not necessarily influence the result of a tournament when the duplicates have worse fitness values than other sampled individuals, the probability of

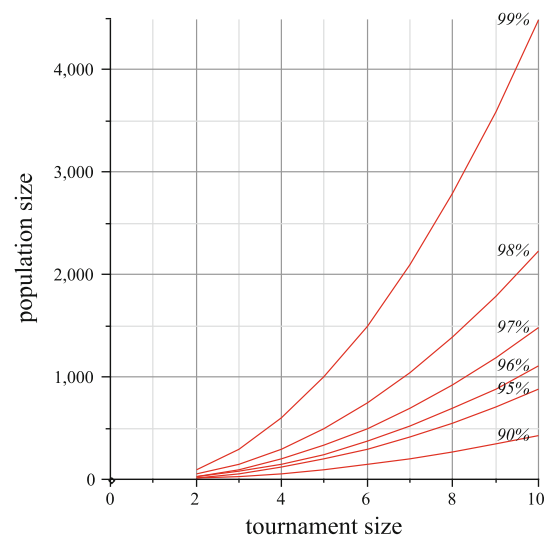


Fig. 11 Confidence level, population size, and tournament size. Note that tournament size is discrete but the plot shows curves to aid interpretation

significant difference between STS and NRTS will be even smaller. Therefore, eliminating the multi-sampled issue in STS is unlikely to change the selection performance significantly. As a result, the multi-sampled issue is generally not crucial to the selection behaviour in STS.

Given the difficulty of implementing sampling-without-replacement in a parallel architecture, most researchers have abandoned sampling-without-replacement, and used the simpler sampling-with-replacement scheme, hoping that the multi-sampled issue is not important. The results of our analysis justified this choice.

6 Analysis of the not-sampled issue via simulations

The not-sampled issue makes some individuals unable to participate into any tournament, aggravating the loss of program diversity. However, it is not clear how seriously it

⁵ α is significance level, and $100(1 - \alpha)\%$ is the confidence level.

affects GP search. This section shows that the not-sampled issue is insignificant.

An obvious way to tackle the not-sampled issue is to increase the tournament size, since larger tournament sizes provide a higher probability of an individual being sampled. However, increasing tournament size will increase the tournament competition level, and the loss of diversity contributed by not-selected individuals will increase, resulting in even worse total loss of diversity.

The not-sampled issue will only be completely solved if every individual in a population is guaranteed to be sampled at least once during the selection phase. However, the sampling-*with*-replacement method in STS cannot guarantee this no matter how other aspects of selection are changed. Therefore, a sampling-*without*-replacement strategy must be used for this purpose. One option is to use NRTS. Unfortunately, it still cannot completely solve the not-sampled issue unless we configure the tournament size to be the same as the population size. Obviously, applying NRTS with such a configuration is not useful as it is effectively equivalent to always selecting the best of a population.

To investigate whether the not-sampled issue seriously affects the selection performance in STS, we will first develop an approach that satisfies the following requirements: (1) minimises the number of not-sampled individuals, (2) preserves the same tournament competition level as in STS, and (3) preserves selection pressure across the population at a level comparable to STS. We then compare the approach with STS.

6.1 Solutions to the not-sampled issue

A simple sampling-*without*-replacement strategy that solves the not-sampled issue is to only return the losers to the population at the end of each tournament. We termed this strategy as *loser-replacement*. By using this strategy, the size of the population gradually decreases along the way to form the next generation. (At the end, the population will be smaller than the tournament size but these tournaments can be run at a reduced size.) The loser-replacement tournament selection will not have any selection pressure across the population. It will be very similar to a *random sequential selection* where every individual in the population can be randomly selected as a parent to mate but just once. The only difference between the outcomes of the loser-replacement tournament selection and the random sequential selection is the mating order. Although the loser-replacement strategy can ensure zero loss of diversity, it cannot preserve any selection pressure across population. Therefore, it is not very useful.

To satisfy all the essential requirements, we propose another sampling-*without*-replacement strategy. After choosing a winner, all sampled individuals are kept in a temporary pool instead of being immediately returned back to the population. For this strategy, if the tournament size is greater than one, after a number of tournaments, the population will be empty. At that point, the population is refilled from the temporary pool to start a new round of tournaments. More precisely, for a population S and tournaments of size k , the algorithm is

- 1: Initialise an empty temporary pool T
- 2: **while** need to generate more offspring **do**
- 3: **if** $size(S) < k$, **then**
- 4: Refill: move all individuals from T to S
- 5: **end if**
- 6: Sample k individuals without replacement from the population S
- 7: Select the winner from the tournament
- 8: Move the k sampled individuals into T
- 9: **end while**

We term a tournament selection using this strategy as *round-replacement* tournament selection (RRTS). The next subsections analyse this strategy to investigate the impact of the not-sampled issue.

6.2 Modelling round-replacement tournament selection

Assume N is a multiple of k ; then after N/k tournaments, the population becomes empty. The round-replacement algorithm needs to refill the population to start another round of tournaments. There will be k rounds in total in order to form an entire next generation (recall that this is because the standard breeding process is assumed, see Sect. 3). It is obvious that any program will be sampled exactly k times during the selection phase; thus, there is no need to model the sampling probability. The selection probability is given in Lemma 2.

Lemma 2 For a particular program $p \in S_j$, if W_j is the event that p wins or is selected in a tournament of size k , the probability of W_j is:

$$P(W_j) = \frac{\sum_{n=1}^k \frac{1}{n} \binom{|S_j| - 1}{n - 1} \binom{\sum_{i=1}^{j-1} |S_i|}{k - n}}{\binom{N}{k}} \quad (17)$$

Proof The characteristic of RRTS is that it guarantees p will be sampled once in just one of the N/k tournaments in a round. According to this, the effect of a full round of tournaments is to partition S into N/k disjoint subsets. The program p is a member of precisely one of these N/k subsets. Therefore, the probability of it being *selected* in

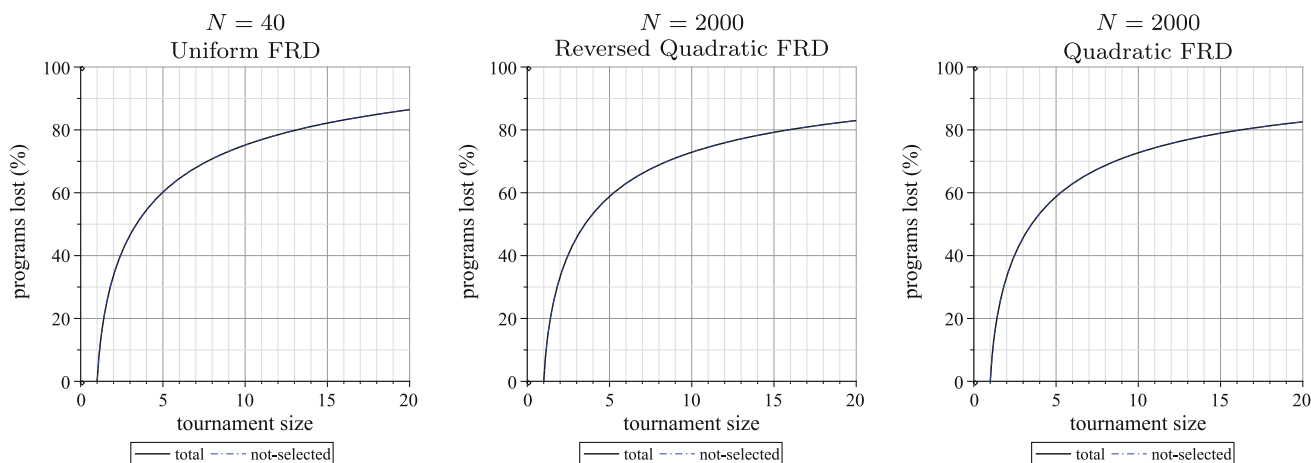


Fig. 12 Loss of program diversity in RRTS on three populations with different FRDs. Note that tournament size is discrete but the plots show curves to aid interpretation

one tournament in a given round is exactly the same as in any other tournament in the same round. Further, the probability of it being selected in one round is exactly the same as in any other rounds since all k rounds of tournaments are independent. Therefore, we only need to model the selection probability of p in one tournament of one round. p could be selected if it is sampled in the tournament and no better-ranked programs are sampled in the same tournament; its selection probability will depend on the number of other programs having the same rank that are sampled in the same tournament.

Let E_j be the event that $p \in S_j$ is selected in a round of tournaments. The total number of ways of constructing a tournament containing the program $p, n - 1$ other programs in the same S_j , and $k - n$ programs in S_1, S_2, \dots, S_{j-1} is⁶:

$$\sum_{n=1}^k \binom{|S_j| - 1}{n - 1} \binom{\sum_{i=1}^{j-1} |S_i|}{k - n} \tag{18}$$

As each of the n programs from $|S_j|$ has an equal probability to be chosen as the winner, and there are $\binom{N - 1}{k - 1}$ ways of constructing a tournament containing p , the probability of E_j is

$$P(E_j) = \frac{\sum_{n=1}^k \frac{1}{n} \binom{|S_j| - 1}{n - 1} \binom{\sum_{i=1}^{j-1} |S_i|}{k - n}}{\binom{N - 1}{k - 1}} \tag{19}$$

Since there are N/k tournaments in a round and the program p has an equal probability to be selected in any one of the N/k tournaments, the probability of W_j is

$$P(W_j) = \frac{P(E_j)}{N/k}; \tag{20}$$

thus, we obtain Eq. 17.

Let $T_{j,c}$ be the event that p is selected at least once by the end of c th round. As the selection behaviour in any two rounds are independent and identical, the probability of $T_{j,c}$ is

$$P(T_{j,c}) = 1 - (P(\bar{E}_j))^c. \tag{21}$$

This equation together with Eq. 17 will be used to calculate the selection probability distribution for RRTS.

6.3 Selection behaviour analysis

The loss of program diversity, the selection frequency, and the selection probability distribution for RRTS are illustrated in Figs. 12, 13, and 14, respectively.

In Fig. 12, there is only one trend in each chart. This is because individuals are guaranteed to be sampled (precisely sampled once in a round and k times in total), there is no trend of not-sampled individuals. As a result, the total loss of diversity measure and the contribution from not-selected individuals are identical, making the two trends overlapped. Therefore, RRTS minimises the loss of program diversity contributed by not-sampled individuals while maintaining the same tournament competition level as that in STS. Again, there are no noticeable differences between the loss of program diversity measures on different sized populations with different FRDs.

In addition, comparing Figs. 12 with 4, we can find that the total loss of program diversity with RRTS is significantly smaller than with the standard one for small tournament sizes ($k < 4$) in all populations, but slightly larger

⁶ Assuming $\binom{a}{b} = 0$ if $b > a$.

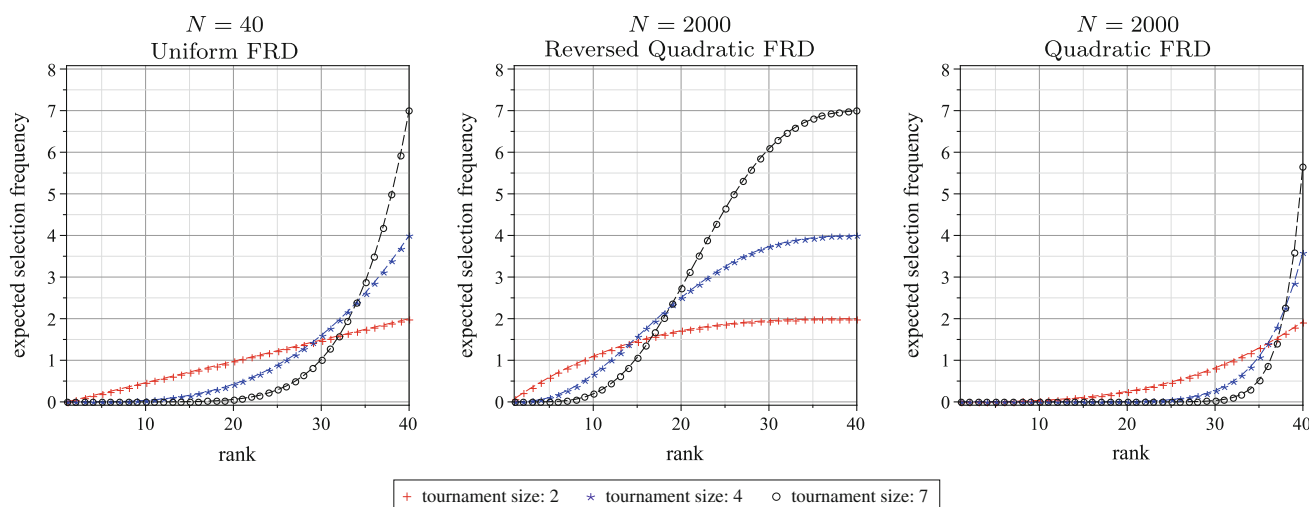


Fig. 13 Selection frequency in RRTS on three populations with different FRDs

for large tournament sizes ($k > 13$) in the small-sized population ($N = 40$).

From Fig. 13, the trends of the selection frequency across each population are still very similar to the corresponding ones in STS (Fig. 5). When a large tournament size (such as $k = 7$) is used, a slightly higher selection frequency is observable in RRTS on the small-sized population ($N = 40$), whereas no noticeable difference exists on the other sized populations. Surprisingly, we find that Fig. 13 seems to be identical to Fig. 8 in NRTS. In fact, Eqs. 12 and 17 are mathematically equivalent. The proof can be found in Appendix.

While the selection frequency is the same in NRTS and RRTS, the selection probability distribution measure reveals the differences. Figure 14 shows that RRTS has some different behaviour from STS (Fig. 6) and also from NRTS (Fig. 9), especially when the tournament size is 2. The differences are related to the top-ranked individuals, whose selection probabilities reach 100% very quickly in the first round.

From the simulation results, although every program can be sampled in RRTS, not all of these “extra” sampled programs can win tournaments. In addition, the number of extra programs which won the tournaments do not necessarily contribute to evolution. Therefore, the overall contribution to the GP performance from these extra sampled programs may be limited, and we will further investigate this via empirical experiments in Sect. 8.

Recall that the selection frequencies are identical between NRTS and RRTS, but the corresponding selection probability distributions are different. This shows that selection frequency is not always adequate for distinguishing selection behaviour in different selection schemes.

7 Discussion of awareness of evolution dynamics

As mentioned in Sect. 1, the evolutionary learning process is dynamic and requires different parent selection pressure at different learning stages. STS is not aware of the dynamic requirements. This section discusses whether the no-replacement and the round-replacement tournament selections are aware of the evolution dynamics and are able to tune parent selection pressure dynamically based on the simulation results of the selection frequency measure (see Figs. 8 and 13) and the selection probability distribution measure (see Figs. 9 and 14).

Overall, for the uniform FRD, NRTS, and RRTS favour better-ranked individuals for all tournament sizes as expected. For the reversed quadratic and the quadratic FRDs, the selection bias is even more significant.

In particular, for the reversed quadratic FRD, there are more individuals of worse-ranked fitness that received selection preference. The GP search will still wander around without paying sufficient attention to the small number of outstanding individuals. Ideally, a good selection schema should focus on the small number of good individuals to speed up evolution.

For the quadratic FRD, the selection frequencies are strongly biased towards individuals with better ranks. The population diversity will be quickly lost, the convergence may speed up, and the GP search may be confined in local optima. Ideally, a good selection scheme should slow down the convergence.

Unfortunately, neither NRTS nor RRTS can change parent selection pressure to meet the expectations. They are the same as STS, being unable to know the dynamic requests, and thus fail to tune parent selection pressure dynamically.

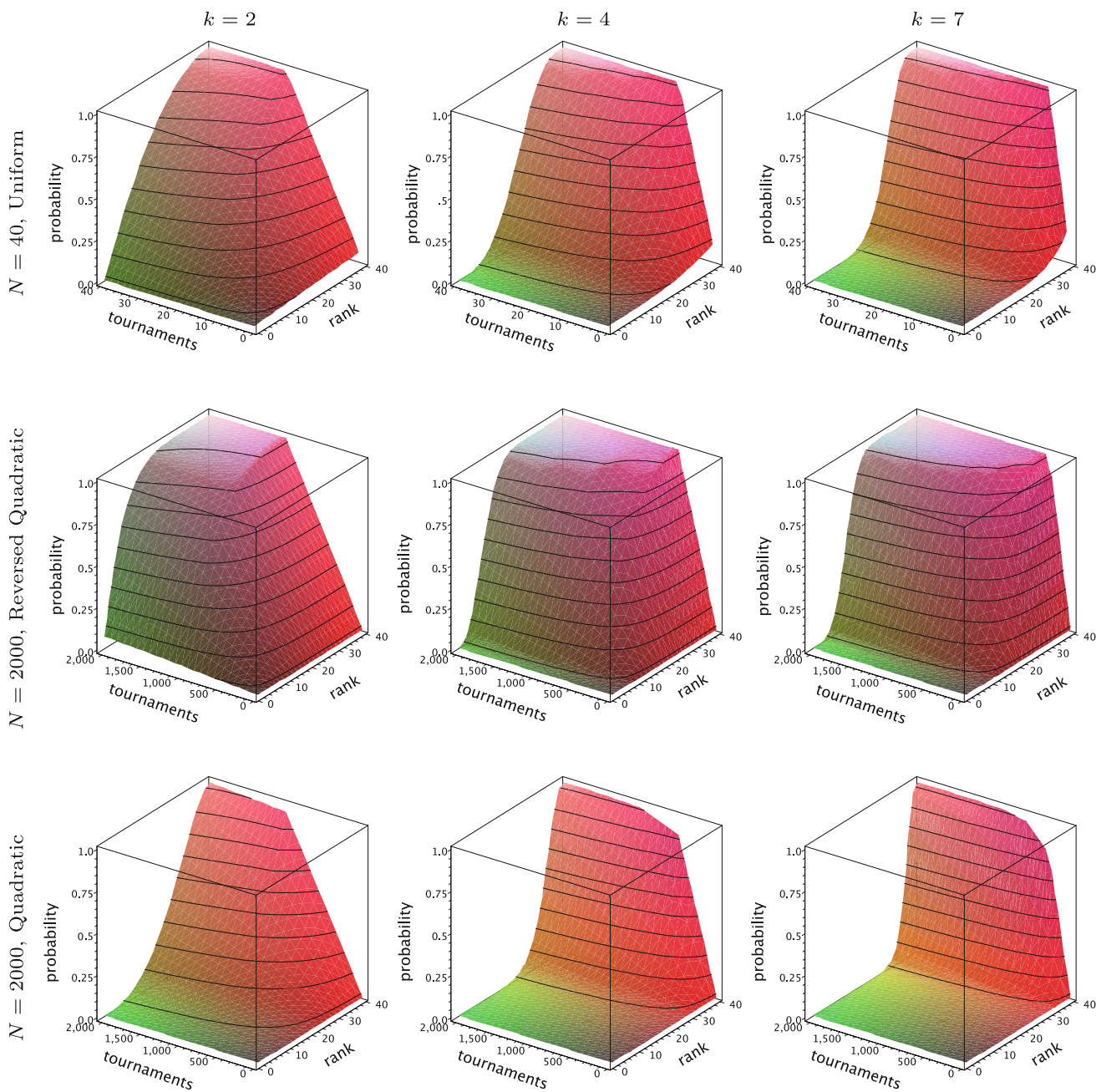


Fig. 14 Selection probability distribution in RRTS with tournament size 2, 4, and 7 on three different FRDs

8 Analyses via experiments

To further verify the findings in the mathematical modelling analysis, this section analyses and compares the effect of STS, NRTS, and RRTS via experiments.

8.1 Data sets

We chose three typical problems of varying difficulty in different domains commonly used in GP in the experiments: an Even- n -Parity problem (EvePar), a Symbolic

Regression problem (SymReg), and a Binary Classification problem (BinCla).

8.1.1 EvePar

An even- n -parity problem has an input of a string of n Boolean values. It outputs *true* if there are an even number of true's, and otherwise *false*. The most characteristic aspect of this problem is the requirement to use all inputs in an optimal solution and a random solution could lead to a score of 50% accuracy (Gustafson 2004). Furthermore,

optimal solutions could be dense in the search space as an optimal solution generally does not require a specific order of the n inputs presented. EvePar considers the case of $n = 6$. Therefore, there are 2^6 combinations of unique 6-bit length strings as fitness cases.

8.1.2 SymReg

SymReg is shown in Eq. 22 and visualised in Fig. 15. We generated 100 fitness cases by choosing 100 values for x from $[-5, 5]$ with equal steps.

$$f(x) = \exp(1 - x) \times \sin(2\pi x) + 50\sin(x) \quad (22)$$

8.1.3 BinCla

BinCla involves determining whether examples represent a *malignant* or a *benign* breast cancer. The dataset is the Wisconsin Diagnostic Breast Cancer dataset chosen from the UCI Machine Learning repository (Newman et al. 1998). BinCla consists of 569 data examples, where 357 are benign and 212 are malignant. It has 10 numeric measures (see Table 1) computed from a digitised image of a fine needle aspirate of a breast mass and are designed to describe characteristics of the cell nuclei present in the image. The mean, standard error, and “worst” of these measures are computed, resulting in 30 features (Newman et al. 1998). The whole original data set is split randomly and equally into a training data set, a validation data set, and a test data set with class labellings being evenly distributed across the three data sets for each individual GP run.

8.2 Terminal sets, function sets, and fitness functions

The terminal set for EvePar consists of *six* Boolean variables. The terminal set for SymReg and BinCla includes a single

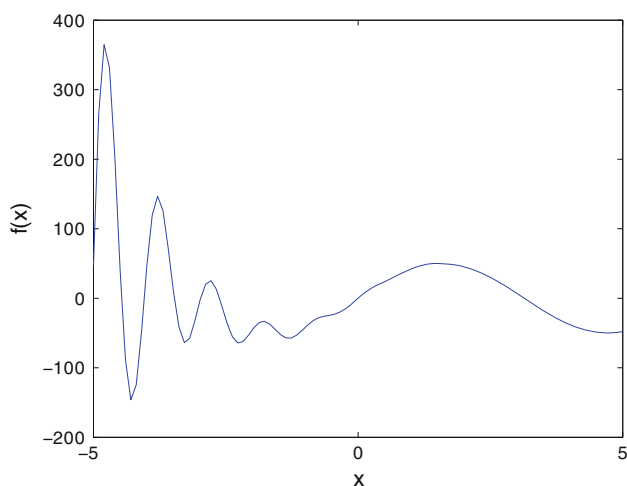


Fig. 15 The symbolic regression problem

Table 1 Ten features in the dataset of BinCla

a	Radius	f	Compactness
b	Texture	g	Concavity
c	Perimeter	h	Concave points
d	Area	i	Symmetry
e	Smoothness	j	Fractal dimension

Table 2 Function sets and fitness functions

	Unary	Binary	Ternary	Fitness function
EvePar		and, or, not	if	Misses
SymReg	Abs, sin, exp	+, −, *, /		RMS error
BinCla	Abs, sin, sqrt	+, −, *, /	if	Classification error rate

variable x and 30 terminals, respectively. Real-valued constants in the range $[-5.0, 5.0]$ are also included in the terminal sets for SymReg and BinCla. The function sets and the fitness functions of the three problems are shown in Table 2.

8.3 Genetic parameters and configuration

The genetic parameters are the same for all three problems. The ramped half-and-half method is used to create new programs and the maximum depth of creation is four. To prevent code bloat, the maximum size of a program is set to 50 nodes based on some initial experimental results. The standard subtree crossover and mutation operators are used (Koza 1992). The crossover rate, the mutation rate, and the copy rate are 85, 10, and 5%, respectively. The best program in the current generation is explicitly copied into the next generation, ensuring that the population does not lose its previous best solution. A run is terminated when the number of generations reaches the pre-defined maximum of 101 (including the initial generation), or the problem has been solved (there is a program with a fitness of zero on the training data set), or the error rate on the validation set starts increasing (for BinCla). Three tournament sizes 2, 4, and 7 are used. Consequently, the population size is set to 504 in order to have zero remainder at the end of a round of tournaments in RRTS.

We ran experiments comparing three GP systems using STS, NRTS, and RRTS, respectively, for each of the three problems. In each experiment, we repeated the whole evolutionary process 500 times independently. In each of the 500 runs, an initial population is generated randomly and is provided to all GP systems in order to reduce the performance variance caused by different initial populations.

8.4 Experimental results and analysis

Table 3 compares the performances of the three GP systems. The measure for EvePar is the failure rate, measuring the fraction of runs that were not able to return the ideal solution. The best value is zero per cent, meaning that every run is successful. The measures for SymReg and BinCla are the averages of the RMS error and the classification error rate on test data over 500 runs, respectively; thus, the smaller the value, the better the performance. Note that the standard deviation is shown after the \pm sign.

The results demonstrate that the GP system using NRTS has the almost identical performance as the GP system using STS. The results confirm that for most common and reasonable tournament sizes and population sizes, the multi-sampled issue seldom occurs and is not critical in GP.

However, the results show that the GP system using RRTS has some advantages over the GP system using STS. In order to provide statistically sound comparison results, we calculated the confidence intervals at the 95% level (two-sided) for the differences in failure rates, in RMS errors, and in error rates for EvePar, SymReg, and BinCla, respectively (see Table 4). For EvePar, we used the formula

$$\hat{P}_1 - \hat{P}_2 \pm Z \sqrt{\hat{P}_1(1 - \hat{P}_1)/500 + \hat{P}_2(1 - \hat{P}_2)/500} \quad (23)$$

where \hat{P}_1 is the failure rate using RRTS, \hat{P}_2 is the failure rate using STS, and Z is 1.96 for 95% confidence (Box et al. 2005). For SymReg and BinCla, we first calculated the difference of the measures between a pair of runs using

the same initial population for each of the 500 pairs of runs and then used the formula

$$\bar{x} \pm Z \frac{s}{\sqrt{500}} \quad (24)$$

to calculate the confidence interval, where \bar{x} is the average difference over 500 values and s is the standard deviation (Box et al. 2005). If zero is not included in the confidence interval, then the difference is statistically significant (Box et al. 2005).

From the table, for tournament size 2 and for SymReg and BinCla problems, the improvement of RRTS is statistically significant. However, practically the differences are small (see Table 3). For tournament sizes 4 and 7, there are no statistically significant differences between RRTS and STS as only 1.8 and 0.09% of the population are not-sampled, respectively, in STS (Poli and Langdon 2006).

We also compared the best performance of RRTS with the best performance of STS for SymReg and BinCla for different tournament sizes; the differences were not statistically significant either. The results confirm that these extra sampled programs have limited contribution to the overall search performance.

Sokolov and Whitley’s (2005) findings suggested that performance could be improved by addressing the not-sampled issue in a genetic algorithm using a tournament size of 2. Our experiments confirmed this in GP for some data sets and showed that the improvement was statistically significant, though not large. However, Sokolov and Whitley considered only tournament size 2. Our experiments included larger tournament sizes and showed that there was no statistically significant improvement for the larger tournament sizes in GP. Furthermore, the performance of larger tournament sizes with STS was as good as or better than the performance of tournament size 2 with RRTS. Therefore, there is little advantage in addressing the not-sampled issue in practice.

The results show that although the not-sampled issue can be solved, overall the different selection behaviour provided by RRTS alone appears to be unable to significantly improve a GP system for the given tasks for common settings. The not-sampled issue does not *seriously* affect the selection performance in STS.

Table 3 Performance comparison between STS, NRTS, and RRTS

Tournament selection		EvePar	SymReg	BinCla
Scheme	Size	Failure (%)	RMS error	Test error (%)
STS	2	100	48.2 ± 5.2	9.2 ± 2.9
	4	80.6	37.6 ± 8.3	8.7 ± 2.7
	7	82.4	40.9 ± 11.3	8.7 ± 2.7
NRTS	2	100	48.3 ± 5.2	9.2 ± 2.9
	4	80.6	37.6 ± 8.4	8.7 ± 2.7
	7	82.5	41.1 ± 11.2	8.7 ± 2.6
RRTS	2	99.6	47.4 ± 5.3	8.4 ± 2.7
	4	79.4	38.3 ± 8.0	8.6 ± 2.6
	7	77.6	40.6 ± 11.4	8.8 ± 2.7

Table 4 Confidence intervals for differences in performance between RRTS and STS at 95% level

Tournament size	EvePar	SymReg	BinCla
2	(−0.95, 0.15)	(−1.48, −0.24)	(−1.05, −0.43)
4	(−6.16, 3.76)	(−0.22, 1.57)	(−0.32, 0.24)
7	(−9.75, 0.15)	(−1.47, 0.85)	(−0.25, 0.32)

9 Conclusions

This paper clarified the impacts of the multi-sampled and the not-sampled issues in STS. It used the loss of program diversity, the selection frequency, and the selection probability distribution on three populations with different FRDs to simulate parent selection behaviours in the no-replacement and the round-replacement tournament selections, which are

the solutions to the multi-sampled and the not-sampled issues, respectively. Furthermore, it provided experimental analyses of the no-replacement and the round-replacement tournament selections in three problem domains with different difficulties. The simulations and experimental analyses provided insight into the parent selection in tournament selection and the outcomes are as follows:

The multi-sampled issue *seldom* occurs in STS when common and realistic tournament sizes and population sizes are used. Therefore, although the sampling-without-replacement strategy in no-replacement tournament selection can solve the multi-sampled issue, there is no significantly different selection behaviour between the no-replacement and the STS schemes. The simulation and experimental results justify the common use of the simple sampling-with-replacement scheme.

The not-sampled issue mainly occurs when small tournament sizes are used in STS. Our round-replacement tournament selection using an alternative sampling-without-replacement strategy can solve the issue without altering other aspects in STS. The different selection behaviour in the round-replacement tournament selection compared with the standard one leads to better results only when tournament size 2 is used for some problems (those that need low parent selection pressure in order to find acceptable solutions). However, there is no significant performance improvement for relatively large and common tournament sizes such as 4 and 7. The performance using these tournament sizes with STS was similar to that using a tournament size of 2 with the round-replacement tournament selection. Solving the not-sampled issue does not appear to significantly improve a GP system: the not-sampled issue in STS is not critical.

Overall, different sampling replacement strategies have little impact on the parent selection pressure. Eliminating the multi-sampled issue and the not-sampled issues does not significantly change the selection behaviour over STS and cannot tune the selection pressure in dynamic evolution. In order to conduct effective parent selection in GP, further research should be emphasised on tuning parent selection pressure dynamically along evolution instead of developing alternative sampling replacement strategies.

Although this study is conducted in GP, the results are expected to be applicable to other EAs as we did not put any constraints on the representations of the individuals in the population. However, further investigation needs to be carried out.

Appendix: Proof of Eqs. 12 and 17 being equivalent

Equation 17 can be simplified to:

$$\begin{aligned}
 P(W_j) &= \frac{\sum_{n=1}^k \frac{1}{n} \frac{(|S_j|-1)!}{(n-1)! (|S_j|-1-n+1)!} \binom{\sum_{i=1}^{j-1} |S_i|}{k-n}}{\binom{N}{k}} \\
 &= \frac{\sum_{n=1}^k \frac{(|S_j|-1)!}{n! (|S_j|-n)!} \binom{\sum_{i=1}^{j-1} |S_i|}{k-n}}{\binom{N}{k}} \\
 &= \frac{\sum_{n=1}^k \frac{1}{|S_j|} \frac{|S_j|!}{n! (|S_j|-n)!} \binom{\sum_{i=1}^{j-1} |S_i|}{k-n}}{\binom{N}{k}} \\
 &= \frac{\sum_{n=1}^k \binom{|S_j|}{n} \binom{\sum_{i=1}^{j-1} |S_i|}{k-n}}{\binom{N}{k} |S_j|}
 \end{aligned}$$

After applying the relation $\sum_{m=0}^n \binom{r}{m} \binom{s}{n-m} = \binom{r+s}{n}$ (Abramowitz and Stegun 1965), we can further simply the equation to

$$\begin{aligned}
 &= \frac{\binom{|S_j| + \sum_{i=1}^{j-1} |S_i|}{k} - \binom{|S_j|}{0} \binom{\sum_{i=1}^{j-1} |S_i|}{k}}{\binom{N}{k} |S_j|} \\
 &= \frac{\binom{\sum_{i=1}^j |S_i|}{k} - \binom{\sum_{i=1}^{j-1} |S_i|}{k}}{\binom{N}{k} |S_j|}
 \end{aligned}$$

which is the same as Eq. 12.

References

- Abramowitz M, Stegun IA (eds) (1965) Handbook of mathematical functions. Dover, New York
- Affenzeller M, Wagner S, Winkler S (2005) GA-selection revisited from an ES-driven point of view. In: Artificial intelligence and knowledge engineering applications: a bioinspired approach. Lecture notes in computer science, vol 3562. Springer, Berlin, pp 262–271
- Agnelli D, Bollini A, Lombardi L (2002) Image classification: an evolutionary approach. Pattern Recognit Lett 23(1–3):303–309
- Akyol A, Yaslan Y, Erol OK (2007) A genetic programming classifier design approach for cell images. In: Mellouli K (ed) Proceedings of the 9th European conference on symbolic and quantitative approaches to reasoning with uncertainty, ECSQARU, Hammamet, Tunisia, October 31–November 2, 2007. Lecture notes in computer science, vol 4724. Springer, Berlin, pp 878–888

- Andreae P, Xie H, Zhang M (2008) Genetic programming for detecting rhythmic stress in spoken english. *Int J Knowl-Based Intell Eng Syst* (Special Issue on Genetic Programming) 12(1):15–28
- Back T (1994) Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. In: *Proceedings of the first IEEE conference on evolutionary computation*, pp 57–62
- Blickle T, Thiele L (1995) A mathematical analysis of tournament selection. In: *Proceedings of the sixth international conference on genetic algorithms*, pp 9–16
- Blickle T, Thiele L (1997) A comparison of selection schemes used in evolutionary algorithms. *Evol Comput* 4(4):361–394
- Box G, Hunter S, Hunter WG (2005) *Statistics for experimenters: design, innovation, and discovery*, 2nd edn. Wiley, Berlin
- Brameier M, Banzhaf W, Informatik F (2001) A comparison of linear genetic programming and neural networks in medical data mining. *IEEE Trans Evol Comput* 5:17–26
- Branke J, Andersen HC, Schmeck H (1996) Global selection methods for SIMD computers. In: *Proceedings of the AISB96 workshop on evolutionary computing*, pp 6–17
- Brindle A (1981) *Genetic algorithms for function optimisation*. PhD thesis, Department of Computing Science, University of Alberta
- Bulmer MG (1980) *The mathematical theory of quantitative genetics*. Oxford University Press, Oxford
- Castillo F, Kordon A, Sweeney J, Zirk W (2006) Using genetic programming in industrial statistical model building. In: O'Reilly U-M et al (eds) *Genetic programming theory and practice II*, chap 3. Springer, Berlin, pp 31–48
- Ciesielski V, Mawhinney D (2002) Prevention of early convergence in genetic programming by replacement of similar programs. In: *Proceedings of the 2002 Congress on evolutionary computation*. IEEE Press, Berlin, pp 67–72
- de Jong K (2007) Parameter setting in eas: a 30 year perspective. In: *Parameter setting in evolutionary algorithms*. Springer, Berlin, pp 1–18
- de Sa LB, Mesquita A (2008) Evolutionary synthesis of low-sensitivity equalizers using adjacency matrix representation. In: Keijzer M et al (eds) *Proceedings of the 10th annual conference on genetic and evolutionary computation*, Atlanta, GA, USA, 12–16 July 2008. ACM, New York, pp 1283–1290
- Eiben AE, Smith JE (2003) *Introduction to evolutionary computing*. Springer, Berlin
- Espejo PG, Ventura S, Herrera F (2010) A survey on the application of genetic programming to classification. *IEEE Trans Syst Man Cybernet C* 40(2):121–144
- Goldberg DE, Deb K (1991) A comparative analysis of selection schemes used in genetic algorithms. In: *Foundations of genetic algorithms*, pp 69–93
- Grefenstette JJ, Baker JE (1989) How genetic algorithms work: a critical look at implicit parallelism. In: Schaffer JD (ed) *Proceedings of the 3rd international conference on genetic algorithms*. Morgan Kaufmann, pp 20–27
- Gustafson SM (2004) *An analysis of diversity in genetic programming*. PhD thesis, University of Nottingham
- Holland JH (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor
- Hong J-H, Cho S-B (2004) Lymphoma cancer classification using genetic programming with snr features. In: *Proceedings of 7th EuroGP conference*, pp 78–88
- Koza JR (1992) *Genetic programming—on the programming of computers by means of natural selection*. MIT Press, Cambridge
- Koza JR, Bennett FH III, Andre D, Keane MA (1999) *Genetic programming III: Darwinian invention and problem solving*, 1st edn. Morgan Kaufmann
- Koza JR, Keane MA, Streeter MJ, Mydlowec W, Yu J, Lanza G (2003) *Genetic programming IV: routine human-competitive machine intelligence*. Kluwer, Dordrecht
- Lee W-C (2006) Genetic programming decision tree for bankruptcy prediction. In: *Proceedings of the 2006 joint conference on information sciences, JCIS 2006*, Kaohsiung, Taiwan, ROC, October 8–11 2006. Atlantis Press
- Li J, Tsang EPK (2000) Reducing failures in investment recommendations using genetic programming. In: *Computing in economics and finance*. Universitat Pompeu Fabra, Barcelona, Spain, 6–8 July 2000
- Lima CF, Pelikan M, Goldberg DE, Lobo FG, Sastry K, Hauschild M (2007) Influence of selection and replacement strategies on linkage learning in boa. In: *Proceedings of IEEE Congress on evolutionary computation*, pp 1083–1090
- Miller BL, Goldberg DE (1995) Genetic algorithms, tournament selection, and the effects of noise. Technical report 95006, University of Illinois at Urbana-Champaign, July 1995
- Miller BL, Goldberg DE (1996) Genetic algorithms, selection schemes, and the varying effects of noise. *Evol Comput* 4(2):113–131
- Motoki T (2002) Calculating the expected loss of diversity of selection schemes. *Evol Comput* 10(4):397–422
- Muhlenbein H, Schlierkamp-Voosen D (1993) Predictive models for the breeder genetic algorithm, I: continuous parameter optimization. *Evol Comput* 1(1):25–49
- Newman DJ, Hettich S, Blake CL, Merz CJ (1998) UCI repository of machine learning databases
- Poli R, Langdon WB (2006) Backward-chaining evolutionary algorithms. *Artif Intell* 170(11):953–982
- Popovici E, de Jong K (2003) Understanding EA dynamics via population fitness distributions. In: *Proceedings of the genetic and evolutionary computation conference 2003*, pp 1604–1605
- Popp RL, Montana DJ, Gassner RR, Vidaver G, Iyer S (1998) Automated hardware design using genetic programming, VHDL, and FPGAs. In: *IEEE international conference on systems, man, and cybernetics*, vol 3, San Diego, CA USA, 11–14 October 1998. IEEE, pp 2184–2189
- Schmidt MD, Lipson H (2007) Learning noise. In: Thierens D, et al (eds) *Proceedings of the 9th annual conference on genetic and evolutionary computation*, vol 2, London, 7–11 July 2007. ACM Press, pp 1680–1685
- Smits G, Kordon A, Vladislavleva K, Jordaan E, Kotanchek M (2005) Variable selection in industrial datasets using pareto genetic programming. In: Yu T, Riolo RL, Worzel B (eds) *Genetic programming theory and practice III*. Genetic programming, vol 9, chap 6, 12–14 May 2005. Springer, Ann Arbor, pp 79–92
- Sokolov A, Whitley D (2005) Unbiased tournament selection. In: *Proceedings of genetic and evolutionary computation conference*. ACM Press, New York, pp 1131–1138
- Vanyi R (2005) Practical evaluation of efficient fitness functions for binary images. In: Rothlauf F et al (eds) *Applications of evolutionary computing, EvoWorkshops2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. LNCS, vol 3449, Lausanne, Switzerland, 30 March–1 April 2005. Springer, Berlin, pp 310–324
- Winkler S, Affenzeller M, Wagner S (2008) Offspring selection and its effects on genetic propagation in genetic programming based system identification. *Cybernet Syst* 2:549–554
- Xie H, Zhang M, Andreae P (2007) Another investigation on tournament selection: modelling and visualisation. In: *Proceedings of genetic and evolutionary computation conference*, pp 1468–1475
- Zhang M, Ciesielski V, Andreae P (2003) A domain independent window-approach to multiclass object detection using genetic programming. *EURASIP J Appl Signal Process* 2003(8):841–859
- Zhang W, ming Wu Z, ke Yang G (2004) Genetic programming-based chaotic time series modeling. *J Zhejiang Univ Sci* 5(11):1432–1439
- Zhang M, Gao X, Lou W (2006) Gp for object classification: brood size in brood recombination crossover. In: *The 19th Australian joint conference on artificial intelligence*. LNAI, vol 4303. Springer, Berlin, pp 274–284