DISPAR-Tournament: A Parallel Population Reduction Operator That Behaves Like a Tournament

Ogier Maitre, Deepak Sharma, Nicolas Lachiche, and Pierre Collet

LSIIT - UMR 7005 Pôle API Bd Sébastien Brant BP 10413 67412 Illkirch France
{ogier.maitre,nicolas.lachiche,deepak.sharma,pierre.collet}@unistra.fr

Abstract. This paper presents an experimental study of different variants of tournament selection, and proposes a new DISPAR-tournament (Disjoint Sets Parallel tournament) operator for population reduction to be used in parallel implementations of evolution strategies and other evolutionary algorithms.

1 Introduction

Selection is an important operator in evolutionary algorithms: too much selection pressure can lead to premature convergence, whereas too little may not allow to obtain good results. Many selection operators like proportionate selection, ranking, stochastic universal sampling, sigma truncation, tournament, stochastic tournament, evolutionary programming tournament etc. have been studied in literature. However, tournament selection is often preferred over other selection methods because it is fast, scales easily, can be used in distributed environments and does not select proportionally to fitness value.

As other selection operators, tournament can be used in two variants: with or without replacement, depending on whether it is used for parents selection (an individual can be selected several times) or population reduction (an individual can be selected only once). Where tournament with replacement has been extensively studied in the literature, not much can be found on the kind of tournament selection without replacement that is often used in $(\mu + \lambda)$ or (μ, λ) Evolution Strategies [9], even though it will be shown in this paper that the two variants behave differently with respect to selection intensity and loss of diversity.

Finally, the advent of parallel computers with global shared memory makes it desirable to have parallel versions of these operators. Indeed, in an evolutionary loop, all steps are parallelizable except for population reduction:

1. each processor can initialize individuals independently,
2. then, each processor can independently select n individuals in the population, in order to create a child through crossover and mutation,
3. evaluation of children can be performed in parallel,
4. reduction of the population of parents+children is impossible to do in parallel by using a standard selection operator such as tournament selection, if one wants to make sure that no individual is duplicated in the new population.

This last point is what prevents to fully parallelize an evolutionary algorithm that would need to use a tournament selector as a reduction operator.

This paper will first study the difference between tournament operators with and without replacement, and then propose a new Disjoint Sets Parallel (DISPAR) tournament operator that keeps the characteristics of the sequential tournament operator for population reduction, while allowing its parallelization over a many-core architecture such as Graphic Processing Unit (GPU).

2 Tournament Selection Variants

Tournament is a widely used selection scheme that has been studied in many papers [2,5,10]. This operator is interesting for its many qualities such as the fact that it can be used in asynchronous distributed environments, its diversity preservation, its translation and computational properties. Indeed, for studied tournament versions, execution is lightweight (with an $O(t)$ complexity, where t is the tournament size) and can be parallel. But one has to notice that tournament selection can be used in two different ways, depending on what is needed:

1. Tournament Selection can be used in order to create a set of n parents that will be used by n-ary variation operators to create one or several children. In this case, selected individuals are put back into the population so that they can be used several times as parents.

   The selection of a parent can also be done with or without replacement. In the "with replacement" variant, a t-tournament selector may possibly select the same individual several times, where in the "without replacement" variant, the tournament selector will make sure that all t individuals are different.

   Since in any of these two variants, the same individual can be used several times as a parent, parent selection can be implemented in parallel in a straightforward way: for instance, 100 cores can independently select 2 parents each, using a t-tournament selector to create a child using a binary crossover followed by a mutation.
2. Tournament selection is also used in Evolutionary Strategies to reduce populations: in a $(\mu + \lambda)$-ES where μ is the number of parents and λ the number of created children, it is necessary to select μ individuals from a $\mu + \lambda$ intermediate population to obtain the new generation, whereas in a (μ, λ)-ES (where typically, $\lambda > \mu$), it is necessary to select μ individuals among the λ created children in order to obtain the new generation.

The parents selection and population reduction phases can both use the same tournament selection principle, but the "with" or "without" replacement option drastically changes the behaviour of the operator, as shall be seen below.

Several metrics have been used to study selection operators, such as take-over time [3], rate of elimination of weaker strings [4], selection intensity [6], loss of diversity [2], genetic drift of population fitness variance [7], probability based variance of loss of diversity [5] and many paper study tournament selection, but nearly all papers only study the "with replacement" variant (in [8], Sastry and

```
input  : population $J$, tournament size $t$, resulting population size $n$
output: resulting population $J'$
for $i \leftarrow 1$ to $n$ do
    BestIndividual := randomly selected individual inside $J$;
    for $j \leftarrow 1$ to $t$ do
        Competitor:=randomly selected individual inside $J$;
        if (Competitor is better than BestIndividual) then
            BestIndividual := Competitor;
        end
    end
    $J' \leftarrow$ BestIndividual;
    remove BestIndividual from $J$;//For a tournament without replacement
end
```

Algorithm 1. Tournament "with" and "without" replacement

Goldberg suggest that tournament selection "with" replacement can require more comparisons than "without").

One has to notice, that these definitions of "with" and "without" replacement designate the method to select individuals in order to fill the *tournament pool*. Which is different from the definition used in the rest of the paper. Indeed the term with or without replacement will concern the selected individual.

As proposed in [2], the following subsections use *selection intensity* and *loss of diversity* to compare the two variants of tournament selection.

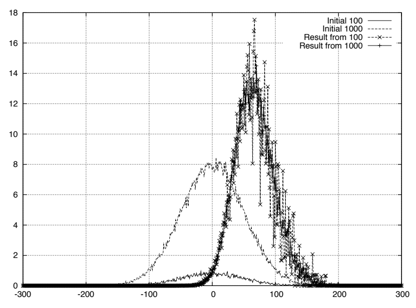
2.1 Tournament "With Replacement"

In this paper, a similar implementation of a tournament "with replacement" as can be found in [2] (*cf.* alg. 1) is used: the tournament selector randomly picks t individuals in the population (where t is the tournament size) and keeps the best individual among this temporary pool.

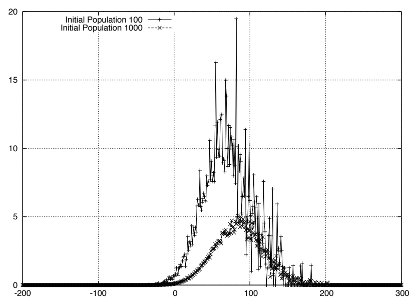
This version is parallel, because the population can be accessed in a read-only mode as one tournament has no influence on another one. [10] gives it as fast and parallel, [2] studies the selection intensity and loss of diversity of such an operator and finally [5] introduces a better formula to predict this loss of diversity.

When this tournament "with replacement" is repeated as many times as there are individuals in the initial population, its effect is to change the distribution of the population, independently of the population size with reference to tournament size. In [2,5] the population size is the same before and after the selection (distribution areas are similar) but fig. 1(a) shows the new distribution (curve shifted to the right) when 1000 7-tournaments with replacement are applied on two initial populations of 100 and 1000 individuals with a gaussian fitness distribution centered on 0.

Resulting distributions are roughly similar, but the new population of 1000 individuals created out of 100 is more noisy.



(a) $\mu = 100$ and $1000 \longrightarrow 1000$



(b) $\mu = 100$ and 1000

Fig. 1. Distribution and selection rates for 1000 7-tournament selections "with replacement" from a population of 100 and 1000

Figure 1(b) shows that the selection rate of fit individuals is higher than the selection rate of average individuals. This rate also gives an idea of the proportion of clones in the resulting mating pool (or population if tournament "with replacement" were used in Evolution Strategies): in 1000 selections among 1000 individuals, a good individual may be selected up to 5 times while in 1000 selections among 100 individuals, it will be selected around 10 times.

While it is fine to use tournament "with replacement" for parents selection (where it is not problematic to select identical individuals several times) using this same operator to create a new population will result in a loss of diversity.

2.2 Tournament "Without Replacement"

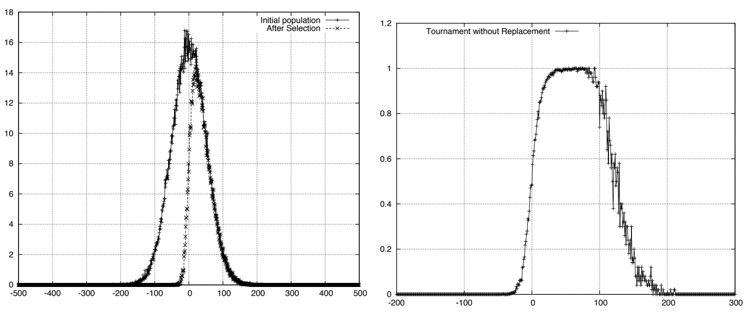
Evolution Strategies need to reduce a population from $(\mu + \lambda)$ (or simply λ for ES-comma) down to μ to create the next generation.

An easy way to implement the necessary reduction operator is to use a tournament selection "without replacement": whenever an individual is selected to be part of the next generation, it is removed from the input population (*cf.* alg. 1). Doing so prevents from selecting the same individual several times, which helps to preserve diversity in the population.

If the modification in the algorithm is very limited, the effects are important. The resulting fitness distribution is totally different. In the case of a (1000+1000)-ES, because cloning is impossible, selecting 1000 individuals out of 2000 yields a population whose distribution has a totally different shape as it hits the right part of the original gaussian curve (*cf* fig. 2(a)).

The curve of selection rates is also impacted, as shown in fig. 2(b): selection rate is always less or equal than 1. When the tournament size increases, the effects are comparable to fitness truncation.

Then, as the population is reduced as individuals are selected, there is no efficient way to parallelize this algorithm over several cores: it is impossible to prevent several cores from choosing identical individual at the same time, unless selection was using atomic operations and synchronizations that would result in



(a) Fitness distribution for t=10 (b) Selection rate for t=10

Fig. 2. Mean random fitness distribution and selection rate for $(\mu + \lambda)$-ES population reduction with 10-tournament "without replacement" for $\mu = \lambda = 1000$

making the selection more expensive, which would degrade the idea of exploiting the intrinsic parallelism of evolutionary algorithms.

2.3 Disjoint Set Parallel (DISPAR) Tournament

Using a serial algorithm on a many-core processor is a real waste of computation power, as such emerging architectures embed hundreds of cores in a chip, that are designed to execute parallel algorithms with many threads. Sequential parts of an algorithm become the bottleneck, as is stated by Amdahl's law [1].

The only way to implement a tournament "without replacement" selection on a truly parallel architecture is to execute parallel tournaments on disjoint sets of individuals.

This paper proposes a parallel tournament-like clone-proof selection algorithm based on the assumption that individuals are randomly distributed across the population, *i.e.* individuals have no spatial correlation. Another strong assumption is that $(\mu + \lambda)/t = k \times \lambda$, where k is an integer. The population is cut into sets of t neighbour individuals, where t is the desired tournament size. These sets are sorted according to their fitness values. Finally, only the k best individuals are kept and the other $t - k$ are discarded.

This scheme offers some advantages, the main one being that it is fully parallel. As in [10], the whole population goes through selection and λ unique individuals are drawn. Another marginal advantage is the guarantee that the best individual will pass on to the next population (elitist tournament reduction).

There are also some drawbacks, like the constraint on the population size and the fact that randomness should be maintained among the population.

As shown in Figure 3, this selection mechanism removes individuals in a regular manner ($t - k$ individuals among t), which allows the breeding step to introduce children in these free spaces, allowing the population to keep its random distribution. For the other constraint, μ and λ can be adapted in order to match it.

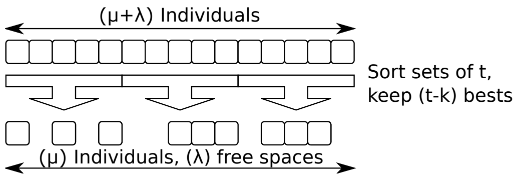


Fig. 3. Parallel tournament without replacement

3 Comparing DISPAR-Tournament with Standard Tournament

3.1 Metrics and Setup of Experimental Comparisons

Selection intensity and loss of diversity measurements from [2] have been used in order to compare these three different operators and results are presented as a mean over 100 runs.

2000 random integer fitness values have been generated, using a gaussian pseudo-random generator with standard deviation of 50 and mean of 0. The selection operator is used to reduce this population of 2000 down to a population of 1000 individuals, therefore emulating a (1000+1000)-ES.

Tests have been done using a serial implementation. Then a single thread sorts every set of the population and keeps k best individuals, removing the $t-k$ discarded ones. The operator is parallel in theory, because no dependency exists between threads. Data is split across sets which are, as explained before, disjoint between each other. The simulation of this operator using a unique core is then completely fair. Sorting the t individuals can be avoided if population size ratio $\frac{\mu+\lambda}{\lambda}$ is equal to t. In this case, only the best individual among a set has to be kept, reducing the complexity of sorting $(O(n\log(n)))$ in a set travelling complexity of only $(O(n))$.

Selection Intensity. The theoretical selection intensity is computed as in [2], where M^* is the mean of the resulting population, M the mean of the input population and σ^* the standard deviation of the input population.

$$I = \frac{M^* - M}{\sigma} \qquad (1)$$

Figure 4 gives selection intensity for three different experiments and a theoretical curve, applied to gaussian fitness distribution, coming from [2]. A first interesting observation is that even using an input population of 2000, the experimental curve still matches the theoretical one.

Then, as explained above but never explored before, the selection intensity curves show that a tournament "with" replacement is very different from a tournament "without" replacement: selection intensity is nearly totally flat above a tournament size 10, where selection intensity of a tournament "with" replacement still increases with tournament size. This can be explained, because every

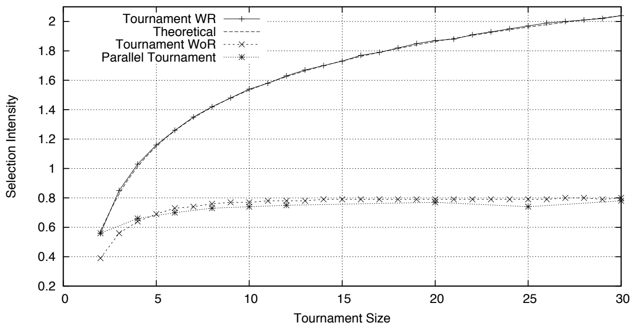


Fig. 4. Selection intensity for different types of selection operators (the implemented tournament "with replacement" and the theoretical curve on top are superimposed, while the, implemented tournament "without replacement" and DISPAR-tournament appear on the bottom)

time a good individual is selected then removed from the population, the average fitness of the population will decrease. As the population size also decreases, the tournament operator has less choice among decreasing quality individuals.

The DISPAR-tournament operator was tried for different tournament sizes, by choosing the population sizes λ and μ so that the constraints above are satisfied. For instance, for $\mu = 1000$, it was not possible to use the parallel operator with less than 3000 individuals for $t = 3$, which gives a selection intensity of more than 0.9, which may never be reached by a tournament "without replacement". These problematic tournament sizes were removed from the test set.

The DISPAR-tournament operator shows acceptable behavior (*i.e.* comparable to a non-parallel tournament "without replacement") for mainly used tournament sizes (3 to 10), but with an increased intensity for tournament size 2 (corresponding to a non-parallel 3-tournament "without replacement"). An implementation of a stochastic binary DISPAR-tournament could probably lower the selection intensity down to 0.4 (a stochastic tournament returns the best individual with probability p, with $.5 \le p < 1$).

Loss of Diversity. Loss of diversity is computed as the sum of the difference of individuals for which the selection rate is less than 1, as detailed in equations 2, 3 and 4. One can see this measurement as the number of individuals which disappear from the population.

$$R(f) = \begin{cases} \dfrac{s^*(f)}{s(f)} & \text{if s(f)>0} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$L(f) = \begin{cases} s(f) - s^*(f) & \text{if } R(f) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$Pd = \sum_{f \in s} (L(f)) \tag{4}$$

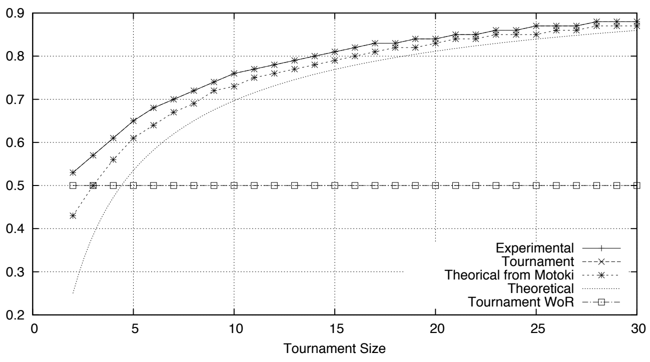


Fig. 5. Loss of Diversity for (1000+1000)-ES reduction applied to a gaussian fitness distribution

Tournament selection without replacement cannot generate any clones, which means that reproduction rates will be less than or equal to 1, even for the best individuals. In this condition, knowing the loss of genetic material is straightforward (it is equal to λ) because this is the number of individuals that are removed from the input population. As can be seen in fig. 5, loss of diversity is a constant equal to the ratio $\frac{\lambda}{\mu+\lambda}$, in this case 0.5.

Loss of diversity was also measured for other operators, where we found different results than in [2] as in [5]. If Motoki asserts in [5] that his way of computing the curve is more accurate, we were not able to reproduce exactly his results. The fact that we are using a larger population could be an explanation.

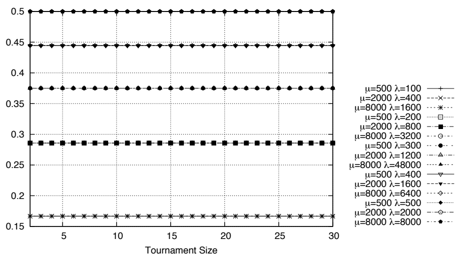
3.2 Effect of Population Size on Tournament

Because in some cases, DISPAR-tournament implies a modification of the population size, the influence of population size was explored. On tournament selection "without replacement," the $\frac{\lambda}{\mu+\lambda}$ ratio has an influence on the loss of diversity. It is obvious, given results from the previous section, that loss of diversity is the same for equal μ/λ ratios. Fig. 6(a) summarizes experiments done on 5 different ratios, for different μ values (500, 2000, 8000).

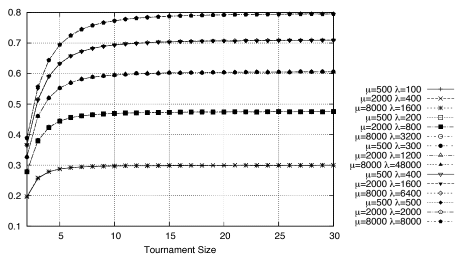
Selection intensity is also influenced by this μ/λ ratio, in the case of selection "without replacement." If selection intensity is still influenced by tournament size, the ratio has a great influence however.

One can think that the input population size has an influence on the fitness of individuals brought by a tournament. Concerning the best individual, the likelihood to select the best individual with a tournament of size t is $\frac{1}{\mu+\lambda} \times t$, for one tournament only. But as μ tournaments are done in order to fill the next parent population, this probability is influenced by the ratio $\frac{\lambda}{\mu+\lambda}$ and not only by the size $\mu + \lambda$.

Figure 6(b) gives the selection intensity for 5 different ratios, for 3 different population μ sizes. As long as the ratio remains the same, selection intensity evolves with the same trend.

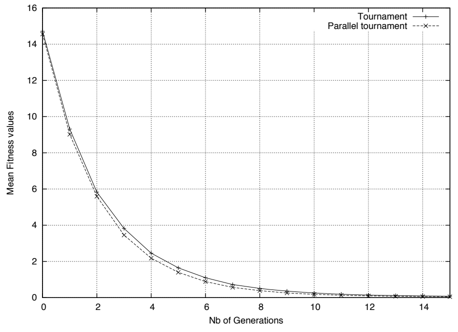


(a) Loss of Diversity

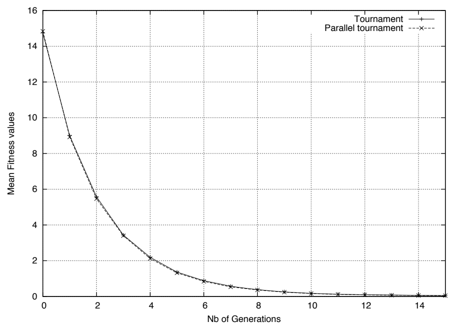


(b) Selection Intensity

Fig. 6. Tournament selection applied to a gaussian randomly distributed population for different population sizes (Curves are superimposed three by three)



(a) t=2, k=1



(b) t=4, k=2

Fig. 7. Mean fitness on the sphere problem

3.3 Implementation

The sphere problem was selected to try this operator, because the influence of selection intensity will show more on a simple convex problem. In fig. 7(a), a tournament size 2 is used for 50 runs and mean fitness value is plotted. Figure 7(b) uses tournament size 4, with the same process. For both experiments, $\mu = 1000$ and $\lambda = 1000$. This is convenient for tournament size 2, but for tournament size 4 $k = 2$, *i.e.* out of a partition of 4 individuals, the 2 best individuals are moved to the next population.

Using these two different tournament sizes, the new selection algorithm does not show any noticeable differences. Convergence follows the same trend, but for tournament size 2, the evolution is slightly faster in generations 2 to 7 during the middle of the runs. Finally, both converge towards the optimum at the same generation.

4 Conclusion and Future Work

A Disjoint Set Parallel (DISPAR) tournament operator has been designed to emulate a tournament "without replacement" operator that is used in Evolution Strategies for population reduction.

In order to mimic the standard operator, some experimental studies have been done using theoretical work on tournament selection that show that this operator does not behave at all as standard tournament selection "with replacement". If this behavior can be reasonably inferred from observation, no work could be found on this version of this well known operator.

A first obvious development is to implement this parallel operator onto a multi-core hardware. Some comparison should be made between a standard serial algorithm and a standard parallel algorithm with synchronization. Two different aspects could be studied, *i.e.* quality of the result and execution time (this operator is designed in order to be executed on a GPU hardware, which is an external computation accelerator). Furthermore, using this operator for population reduction can allow the algorithm to use many threads (as many as the number of tournaments occurring during this reduction step), therefore matching GPU capabilities and requirements.

The DISPAR tournament operator is very close to the original tournament. The principle remains the same, except that the population is cut into small disjoint sets. This allows the operator to behave similarly to the serial version, provided that the population size is adjusted to match operator constraints.

References

1. Amdahl, G.: Validity of the single processor approach to achieving large scale computing capabilities. In: Proceedings of the Spring Joint Computer Conference, pp. 483–485. ACM, New York (1967)
2. Blickle, T., Thiele, L.: A comparison of selection schemes used in evolutionary algorithms. In: Evolutionary Computation, pp. 361–394 (1997)
3. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. In: Foundations of Genetic Algorithms, pp. 69–93. Morgan Kaufmann, San Francisco (1991)
4. Hancock, P.J.B.: An empirical comparison of selection methods in evolutionary algorithms. In: Fogarty, T.C. (ed.) AISB-WS 1994. LNCS, vol. 865, pp. 80–94. Springer, Heidelberg (1994)
5. Motoki, T.: Calculating the expected loss of diversity of selection schemes. Evol. Comput. 10(4), 397–422 (2002)
6. Muhlenbein, H., Schlierkamp-Voosen, D.: Predictive models for the breeder genetic algorithm – i. continuous parameter optimization. Evolutionary Computation 1, 25–49 (1993)
7. Rogers, A., Prugel-Bennett, A.: Genetic drift in genetic algorithm selection schemes. IEEE Transactions on Evolutionary Computation 3(4), 298–303 (1999)
8. Sastry, K., Goldberg, D.E.: Modeling tournament selection with replacement using apparent added noise. Technical report, Intelligent Engineering Systems Through Artificial Neural Networks,11, 129–134 (2001) (also IlliGAL)
9. Schwefel, H.-P.: Numerical optimization of computer models, 2nd edn. John Wiley & Sons, Chichester (1995)
10. Sokolov, A., Whitley, D.: Unbiased tournament selection. In: GECCO 2005: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 1131–1138. ACM, New York (2005)