GENETIC ALGORITHM TUNING: OVERCOMING DIVERSITY LOSS IN TOURNAMENT SELECTION

By

Nathan LeStage

A Project Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved:

W. Randolph Franklin, Project Adviser

Rensselaer Polytechnic Institute
Troy, New York

April 2009
(For Graduation May 2009)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This study analyzes the efficacy of using unbiased tournament selection in Genetic Algorithms compared to the performance of well-tuned random tournament selection with respect to the Travelling Salesman Problem. By stopping diversity loss by non-selection, unbiased tournament selection has been shown to perform better than other forms of selection when neither algorithm has been tuned for optimum performance. However, with parameter tuning and standard optimizations it is shown that the gains made by unbiased selection may be overcome in many cases. Both overall fitness and time to converge (speed) are examined against a series of city tours, populations, crossover pressures, and mutation rates.

1. INTRODUCTION

Genetic Algorithms (GA) are modelled after biological processes to quickly solve or approximate answers to difficult problems not easily answered with raw computing power. For example, NP-hard or NP-complete problems, such as the Knapsack problem, Travelling Salesman problem, or the Hamiltonian Path problem are are often tackled by GAs

Since there are so many different variables in genetic programs, much work has been done to parameterize and create benchmark tests for various aspects of a genetic algorithm. This includes things such as population size, crossover probability, mutation probability, crossover type, population replacement models, elitism, and even the pseudo-random number generators used [1, 2]. In addtion, one area in which there is much research is in comparing the efficiencies of various selection methods, the process by which parents are chosen to propegate their digital DNA to the next generation.

It has long been known that an important aspect of a genetic algorithm is how well it manages loss of diversity between generations. As the various operators are performed upon a generation, if there is too much diversity between members of a population then the algorithm will converge slowly, or perhaps not at all. If there is too little diversity, the population may become sterile and quickly converge upon a sub-optimal solution.

Although there are many different types of selection methods, tournament selection is one of the most well known and commonly used methods in many various genetic algorithms. It allows a finer control of selection pressure, and by extension, optimization, than other methods [3, 4]. By itself, tournament selection is known to have loss of diversity due to a statistical bias from non-sampling and non-selection. Work has been done to remove this bias and increase diversity with the development of an unbiased tournament selection method [5].

While unbiased tournament selection has been shown to perform better in some cases, the studies have focused on simplified algorithm comparisons by minimizing

or removing effects from other parameters, such as mutation. However, most genetic algorithms would likely not leave out these in practice and would instead be tuned or optimized.

This paper investigates whether random tournament selection can perform as good as or better than the unbiased selection methodology when optimization and tuning techniques have been applied. Testing will be performed using one of the best known problems, the Travelling Salesman problem (TSP).

Much work has been done in trying to determine optimal parameters and methodologies to use in genetic programs. This research will help to extend work in evaluating the overall effects of diversity loss that can occur during the selection stages of the genetic algorithm. While it has been demonstrated that an unbiased-tournament selection method can create a more effective GA under specific circumstances, other questions must still be asked. Does unbiased selection always perform better than random selection? As we optimize and tune each GA around the selection methods, does the difference in performance increase or decrease? Can mutation or crossover techniques render the diversity loss to be insignificant?

The answers to these questions can help to direct the choices of designers of future programs that implement GA. Knowing for sure that unbiased methods always perform better would be very useful in determining the proper selection method. Conversely, if we find that the loss of diversity does not make a significant impact after tuning, a designer might make the choice to spend time tuning parameters for a well known random tournament selection instead of implementing a more complex or less efficient selection scheme.

Additionally, one growing area in genetic programming is in the field of dynamic or self optimized genetic algorithms. In these cases, deeper knowledge of selection methods, their potential performance, and the effects of tuning will assist in the design of such algorithms. Knowing that diversity loss can or cannot be compensated for by standard operators will allow for informed decisions.

2. HISTORICAL REVIEW

The basic Genetic Algorithm or Genetic Program takes both its terminology and its inspiration from its biological analogue. Populations of chromosomes are evaluated on the most fit members, which then mix their DNA to create children with properties of each parent.

The best candidates for GA are difficult problems, like the NP-Hard problems mentioned above, or other types of complex combinatorial problems. GA have been even used to help designers optimize complex circuit design and other electronics issues [6].

After initially generating random solutions to the problem at hand, the stochastic process uses genetic recombination techniques to form another population of possible solutions based upon the best qualitfied members of the previous group. Each potential solution of is represented by a chromosome which is simply the data points that make up the solution. The data in each chromosome is encoded in such a manner as to allow it to be acted upon by the various operators for selection, crossover, and mutation.

The most commonly used problem encodings seen in much of the research include binary encoding, which is literally a string of bits, often used for mathematical problems. Sections of the binary string are taken from the parents, and a bit can be simply flipped during mutation. Tree encoding is often used in genetic programming to create expression trees [7], such as when each leaf of the tree is a literal and internal nodes are mathematical operators. Entire sub trees can be swapped and recombined for both crossover and mutation operations. An alternative to tree encoding is graph encoding, which is similar but allows nodes to be connected in cycles. A benefit here is that graphs can often be more efficienct as they often can avoid the recursion used in regular trees [7]. To encode data solutions for problems where the order of members is important, like the Travelling Salesman, permutational encoding is used. In this, a simple list of the members is used indicating the sequence they should be in.

Taking the biological terminology further, each individual data point is referred to as an allele, the most basic component of the solution. In a Travelling Salesman Problem therefore, an allele would be a single city, the encoding of all cities into a single tour would be the chromosome, and the collection of chromosomes from which the parents are drawn is then the population.

Many researchers have studied in depth the wide assortment of functions and parameters that are common in GAs in order to improve the performance and usefulness on various problem sets. A major concern in designing a GA is loss of diversity, especially with respect to the chosen selection scheme. In a typical tournament-type selection, a set of members is chosen at random and all are compared based on the fitness function for the problem. Since it is entirely possible for the best members in the population to be overlooked thanks to the random selections, there is the potential for loss [8].

[9] proposes new types of selection - Correlative Tournament Selection and Correlative Family-based Selection - to help address diversity loss by more closely managing what individuals are chosen as parents or survive to the next generation. In a similar attempt, [5] offer what they call an Unbiased Tournament Selection which guarantees the most fit individual will be chosen as a parent [5].

Another angle of attack to battle diversity loss is through mutation. The mutation operator's main purpose is to inject diversity back into a population and avoid premature stagnation. Care should be taken, however, since too high a rate of mutation can act as plain noise in the system without bringing any benefit of progressing towards a solution. [10] suggests mutation does not show a linear response with performance at high mutation rates. Mutation operators may take on different forms depending on the chromosome encoding type. Bit-encoded chromosomes generally flip a random bit in string. For permutation encodings, such as in the TSP, common operators include "swap" and "insert" [11]. In the former, two randomly chosen elements within the chromosome are swapped. In the latter, one element is chosen and then inserted at random between two other elements.

The well known Travelling Salesman Problem (TSP) is an attempt to find the shortest Hamiltonian circuit on a weighted graph [12]. Or plainly put, it seeks

to get the salesman to every city on his route exactly 1 time, and then return home, by travelling the shortest distance possible. As TSP is known to be NP-Hard [12], genetic algorithms have often been the tool of choice for those attempting to approximate or find the best solution.

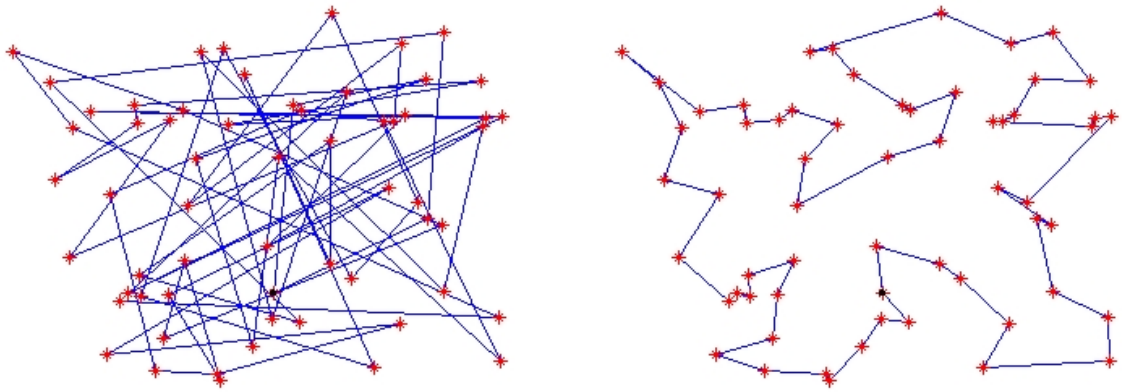

Figure 2.1: TSP: Initial and Final Tours [13]

The problem generally discussed is the symmetric TSP. This is simply the problem as expressed on an underlying symmetric graph where an edge's weight is the same regardless of the direction in which it is traversed. Not only is this a good approximation in most cases, it also significantly reduces the complexity of the problem and halves the number of possible solutions as compared to the asymmetric TSP.

Some research has been done in the field to investigate alternative approaches that can handle 2-D Euclidean graphs without reducing them to a linear representation [14]. This approach attempts to avoid the unavoidable data loss that occurs when mapping multiple dimensions to a single dimension. This is accomplished by randomly partitioning the graph with a series of cuts and reassigning edges that were destroyed by the partitioning process.

3. METHOD

3.1 Hypothesis

The null hypothesis to be confirmed or rejected by this study can be stated as follows:

> *After standard optimization of a Genetic Algorithm's parameters for solving the Travelling Salesman Problem, there is no overall performance gain achieved by implementing unbiased tournament selection compared to a random tournament selection.*

3.2 Measures

The primary indicator of whether diversity loss is overcome is a measure of the optimum fitness achieved by each selection type. The "fitness" is the solution that has been found by the GA - that is, the shortest path that has been found. The maximum fitness, or minimal solution, discovered at the time of convergence will be the major factor in determining the relative performance of the two selection methods.

In addition, the speed of the algorithm can be simply defined as the number of generations that have been completed before convergence upon a solution is realized. While this not the primary concern here, speed is a valuable metric to have in the case that there is no significant difference in fitness.

3.3 Selection Methods

3.3.1 Random Tournament Selection

For clarity, we will refer to the commonly known "tournament selection" operator instead as "*random* tournament selection [5]". This is appropriate in that the players in the tournament are chosen at random, unlike the tournament player matchups in the unbiased tournaments.

In a random tournament of size s, s players are chosen at random from the overall population. Then, the player with the best fitness is chosen as the winner and will be used as a parent for the next population [5]. Both conceptually simple and easy to implement, random selection is also easy to parallelize since all tournaments are completely independent of each other.

Intuitively, as the tournament size get larger it is more probable that the most fit individual (and indeed all individuals) will participate in more tournaments. This most fit member therefore is likely to be selected more often, thus decreasing overall diversity. Indeed, it has been shown that larger tournament sizes increase the loss of diversity [15], and [3] shows that in fact almost half the chromosomes in a population are not selected with a tournament size of only 5.

For these reasons, only binary tournaments for tests on the random selection method were used in this GA. This is additionally appropriate as the unbiased selection method only requires a tournament of size 2, as explained later.

3.3.2 Unbiased Tournament Selection

This selection scheme was developed by [5] as a way to eliminating diversity loss caused by individuals not being sampled to participate in the tournament. The basic concept is to line up two permutations of the entire population, taking care that no individual is matched with itself. Then, each pair is evaluated as a single binary tournament [5].

Since each individual must participate in the selection tournament exactly two times, it is no longer possible to lose the most fit individual due to simple gapping of the random selection function. Further, the most fit individual must be selected exactly twice while the least fit member is not selected at all. Thus, there is still a strong pressure towards convergence to the solution.

One drawback of the unbiased tournament selection described is that it is not able to be parallelized since there are now constraints set on the tournaments. A parallel unbiased tournament selection method has also been devised to help address this, although it is not as complete [5]. We will not examine it here.

3.4 Optimization Parameters

A large number of tests were run on each selection method, stepping through small changes in each of a number of parameters. As stated earlier, tournament size was fixed at 2. All tests were performed on the Travelling Salesman Problem which lends itself to be permutationally encoded.

3.4.1 Uniform Order Based Crossover

The crossover rate controls how two parents of a population swap their alleles in creation of the child.

Crossover rates also have an impact upon the diversity of the populations as too little crossover may cause the GA to converge upon a local minima. Tests used Uniform Order Based Crossover (UOBX) for the permutation problems as it has been demonstrated to offer better performance than other common crossover techniques [11].

As explained by [11], in UBOX each allele in the first parent is transferred to the same position in the child with the probability of p. Then, each of the open positions in the child is filled from the second parent, taking care that the alleles remain in the same order as they were in the parent [11]. See Figure 3.1 for an example.

```
P1 = (6 5 2 8 4 1 3 7 )
P2 = (3 8 1 2 5 6 7 4 )
XO = (X X . X X X . . )
C1 = (6 5 3 8 4 1 2 7 )
C2 = (6 5 1 2 8 3 7 4 )
```

Figure 3.1: UBOX Example

Crossover probability was varied p from 0.5 to 0.7 in increments of 0.1

In addition, many GAs assign an overall probability for crossover which controls whether crossover even occurs. This activity will depress the overall diversity of the GA, so it was not used in this study.

3.4.2 Mutation

Mutation will artificially will create diversity of sorts in the population. Zero or very low mutation rates decrease the chance of pulling away from a local maximum. Too high of a mutation rate will cause undue noise and slow convergence. Very high mutation will begin to seriously degrade performance.

Since the TSP is permutationally encoded, a mutation event consists of swapping two positions in the individual.

We varied the mutation rate from 0 through 0.03 in increments of 0.005. Some suggest the mutation rate should be within 0.5%-1% [1], so this should ensure we did not stop collecting data before the optimum is reached.

3.4.3 Population Size

The population is the collection of potential solutions to the problem. It is from this population that the tournaments will select their players and determine the winners. There is no consensus on an optimal population size for general problems for genetic algorithms. However, it would be remiss to ignore the impact of the size of a population while evaluating effects of that population's diversity. We used population sizes of 30, 100, and 200.

3.4.4 Elitism

Elitism ensures that the most fit member of the population is included in the next generation. This practice is often used because it ensures each generation will be at least as good as its predecessor and in many cases will decrease the time needed to converge upon a solution. However, that is not always beneficial as it can increase the chance of convergence upon a local minimum or maximum. Regardless, this manipulation certainly decreases diversity so, since we are examining the ability to achieve the opposite, we will not use elitism.

3.5 Test Procedure

As noted, we uses the Travelling Salesman Problem as our benchmark test for evaluating the tournament selection schemes. There are several well known test

problems which have been published along with their shortest known tour lengths. We will examine the 52-city, 76-city and 105-city problems [16] as seen in Table 3.1.

Table 3.1: Problem Sets and Known Solutions

Problem Name	Num. Cities	Shortest Known Path
berlin52	52	7542
eli76	76	538
lin105	105	14379

Each of the tour problems was run against a battery of parameters detailed in the section on Parameters and summarized in Table 3.2. For each combination of the parameters, for each tournament selection type, and for each tour, a batch of 200 individual runs through the GA was performed.

Table 3.2: Algorithm Parameters

Parameter	Values
Tournament Type	Random, Unbiased
Tournament Size	2 (Binary Tournament)
Crossover Rate	0.50, 0.60, 0.70
Mutation Rate	0.000, 0.005, 0.010, 0.015, 0.020, 0.025, 0.030
Population Size	30, 100, 200

In addition to the crossover operator, some GA algorithms have an additional parameter that governs the rate at which parents combine to generate the child that is, there is an additional probability of whether crossover will even be performed. As crossover is the driving force behind changes to the population at large, and therefore a major cause of population diversity, the algorithm was constructed to always crossover.

Convergence is detected by the program via a weighted mean average. When the WMA comes within 0.01% of the best detected solution, the algorithm ends. While this is not the most complex method of determining convergence, it is relatively simple to implement. Configured with a look-behind of 100 generations, the technique may slightly inflate the generation count. However, it is equitable and

will not affect the overall performance comparisons between random and unbiased selection schemes.

4. RESULTS

Altogether, 63 batches were run against each city tour for each tournament selection scheme for a total of 378 batches. For each run of the batch, the best solution and the number of generations at convergence was recorded.

The mean number of generations for the batch at convergence and the mean solution for each batch was collected for initial analysis to determine which set of optimizations garnered the best results for each tour and each selection scheme, as seen in Table 4.1 and Table 4.2 .

Table 4.1: Best Fitness Means

Cities	Tournament	Mean Fit.	Mean Gens	Crossover	Mutation	Pop. Size
52	Random	8734	618	0.6	2.5	200
52	Unbiased	8751	656	0.5	3.0	200
76	Random	615	794	0.5	2.0	200
76	Unbiased	621	780	0.5	3.0	200
105	Random	18821	1005	0.5	1.5	200
105	Unbiased	19014	1031	0.5	3.0	200

Table 4.2: Best Speed Means

Cities	Tournament	Mean Fit.	Mean Gens	Crossover	Mutation	Pop. Size
52	Random	9078	577	0.7	0.0	200
52	Unbiased	103217	518	0.7	0.0	200
76	Random	647	702	0.7	0.0	200
76	Unbiased	756	651	0.7	0.0	200
105	Random	20683	915	0.7	0.0	200
105	Unbiased	26015	818	0.7	0.0	200

Note that the mean speed was actually significantly faster, that is, fewer generations before convergence, for the tests with a chromosome population of only 30. However, for analysis purposes, all tests performed with this population can be

ignored because performance was completely inadequate in terms of fitness. Figure 4.1 shows the average solutions for each test done with random selection, aligned by population.

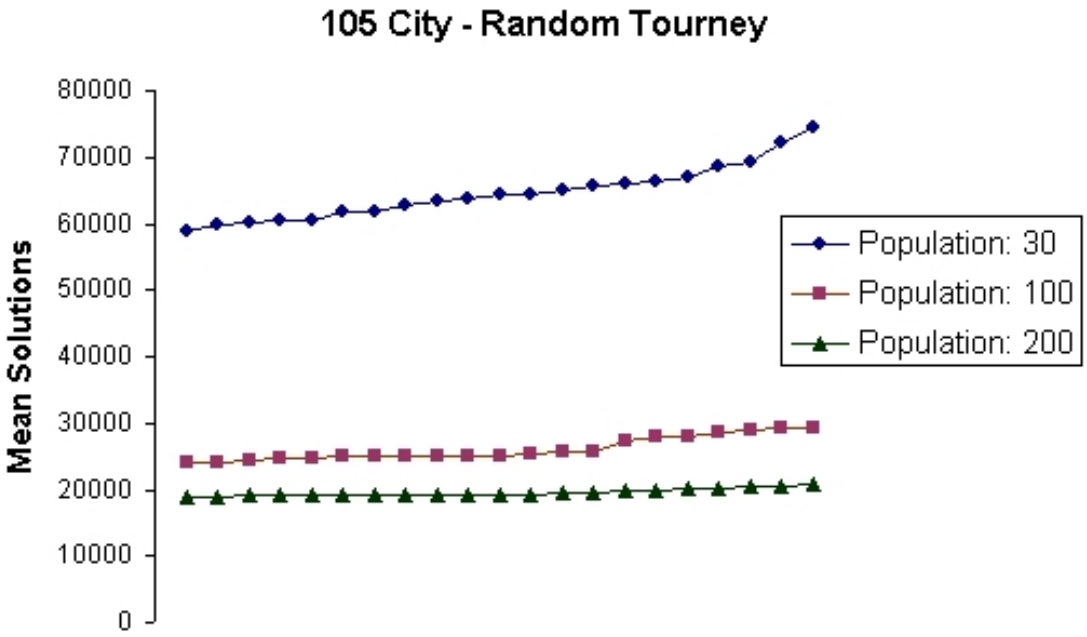

Figure 4.1: Mean Solutions By Population Size

Since the purpose is to determine if optimizations in general can overshadow the choice of selection scheme, the analysis between the two selection schemes should be performed between the best-optimized algorithm runs. Note that it is therefore both acceptable, and indeed necessary, to compare the data between the top performing random selection GA and the top performing unbiased selection GA, even though the optimization parameters are in fact different.

Based on the data from Tables 4.1 and Table 4.2, the data was analysed with the two-sample t-test for unequal variances. So for each tour length, the fastest dataset for random tournament selection was compared against the fastest dataset from for unbiased tournament selection. Likewise, the most fit data was matched between selection mechanisms.

In Table 4.3, the analysis results for fitness optimization is shown. Where p >0.05, the mean data was left out of the table as there is no statistical significance

and is therefore irrelevant.

Table 4.3: Fitness Optimization Analysis (R/U Random / Unbiased)

Cities	**p - Fitness**	p - Speed	Mean Fitness - R / U	Mean Speed - R / U
52	0.649	0.00098		**618** / 656
76	0.0101	0.116	**615** / 621	
105	0.195	0.0248		**1005** / 1031

Table 4.4: Significance: Speed Optimization

Cities	p - Fitness	**p - Speed**	Mean Fitness - R / U	Mean Speed - R / U
52	1.68E-34	4.16E-34	**9078** / 10327	577 / **519**
76	6.47E-28	2.24E-24	**64** / 756	702 / **651**
105	7.41E-29	3.81E-45	**20683** / 26015	915 / **818**

5. DISCUSSION

The chomosome population size had a much more pronounced effect on the outcome of the tests than anticipated. As mentioned earlier, the graph in 4.1 shows distinct banding based on population size. Although only one graph is shown, this banding occurs on all tour sizes and with both selection schemes and is a strong indicator of the relative importance of the population size parameter.

Statistical analysis of the optimized configurations for fitness yielded interesting, albeit mixed, results. For the 52 and 105 city problems, analysis revealed p >0.05, thus agreeing with the hypothesis that there is no gain by using unbiased tournament to reduce diversity. However since there is no significant difference in the fitness, the speed of each method at this configuration must be compared. In doing so, it is found that p $<= 0.05$. In fact it is the random selection method that is significantly faster than unbiased method in both cases.

For the 76 city problem, the opposite holds true - there is a significant difference in the fitness level, and again it is the random selection method that is out-performing the other.

Another picture is formed while looking at the speed-optimized result set in Table 4.4. Here, there is a significant improvement in speed made by the unbiased tournament selection method on all three city tours. This clearly refutes the hypothesis that no gain would be seen. While it is true that random selection has better fitness for each city, this is irrelevant certainly speed is what one cares about if that is the optimization that has been chosen.

Although the hypothesis has been refuted by the speed optimization trials, it would be better to consider each aspect separately when making a decision about which selection method to use. Certainly if speed is the only concern, then unbiased tournament selection would be a good choice. However, more analysis would need to be done to quantify the accuracy of this result and ensure it is acceptable.

A glance at Table 4.2 shows that the crossover-rate was fixed at 0.7, and there was no mutation whatsoever, for each of the fastest trials. A crossover rate other

than 0.5 will favour a one of the parents and cause, on average, more alleles to come from that favored parent. This, combined with a 0% mutation rate, should cause these test configurations to have less divertisy. Combined with the data results, this seems to indicate that there is indeed a higher chance of getting "stuck" on a local minima due to lower population diversity.

6. CONCLUSION AND FUTURE WORK

Typical random tournament selection methods have a known deficiency wherein a population's diversity may be diminished because it may not sample fit individuals. Unbiased tournament selection aims to mitigate this issue by ensuring all members of a population can compete in the tournament. However, performance increases from unbiased tournament selection can be overshadowed by standard GA optimization.

Although speed improvements appear to be pronounced with unbiased selection, the overall difference in the achieved fitness can be insignificant at best, and significantly poorer at worst.

Due to the relative ease of implementation and current widespread usage of the random tournament selection, it makes sense to continue to consider or use this type of selection when designing a genetic algorithm. Additionally, the ease in which this selection scheme can be parallelized lends an even stronger argument for continued usage.

There are additional indications that the size of the chromosome population could have a stronger impact than any other GA parameter. Additional work to determine what that impact might be would be further helpful in making informed decisions about GA design. The banding behaviour based on the chromosomal population size suggests that there may be optimal population sizes based on the length of tour or other factors.

REFERENCES

[1] A. E. Eiben, R. Hinterding, and Z. Michalewicz. "Parameter control in evolutionary algorithms," *IEEE Transations on Evolutionary Computation*, vol. 3, no. 2, pp. 124-141, 1999.

[2] D. Jason, and M. Konstantinos, "An Experimental study of Benchmarking Functions for Evolutionary Algorithms," *International Journal of Computer Mathemathics*, Vol. 79, Apr., pages 403-416, 2002.

[3] T. Blickle, and L. Thiele, "A Comparison of Selection Schemes used in Genetic Algorithms," *TIK Report*, Nr. 11, Dec., 1995.

[4] D. Goldberg, and K. Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," in *Foundations of Genetic Algorithms*, G.J.E. Rawlins, Ed. San Mateo: Morgan Kaufman Publishers, Inc, pp. 69-93, 1991.

[5] A. Sokolov and D. Whitley, "Unbiased Tournament Selection," *Proceedings of the 2005 conference on Genetic and Evolutionary Computation*, pp.1131-1138, 2005.

[6] J. Zhang, H.S. Chung, and W. Lo, "Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, Jun., pp. 326-335,2007

[7] M. Schmidt and H. Lipson, "Comparison of Tree and Graph Encodings as Function of Problem Complexity", in *GECCO'07 Proceedings*, Jul., 2007.

[8] H. Xie, M. Zhang, P. Andreae, and M. Johnston, "Is the Not-Sampled Issue in Tournament Selection Critical?," in *2008 IEEE Congress on Evolutionary Computation*, pp. 3710-3717, 2008.

[9] K. Matsui, "New Selection Method to Improve the Population Diversity in Genetic Algorithms", in *IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC '99 Conference Proceedings*, vol. 1, pp:625-630, 1999.

[10] A. Piszcz, T. Soule, "Genetic Programming: Parametric Analysis of Structure Altering Mutation Techniques," in *GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pp.220-227, 2005.

[11] J. Gottlieb, "Permutation-Based Evolutionary Algorithms for Multidimensional Knapsack Problems," in *Proceedings of the 2000 ACM symposium on Applied computing*, vol. 1, pp. 408-414, 2000.

[12] J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed. Addison Wesley, 2007.

[13] G. Kjellstrm, "Kortad-rundtur (Short Tour)", Oct. 2007, [2009 Apr 17], Available at HTTP:
http://en.wikipedia.org/wiki/Travelling_salesman_problem
A tour obtained after the simulated evolution of 1500 generations. In each generation, 60 of 180 cards (individuals) selected.
These materials are included under the fair use exemption and are restricted from further use.

[14] S. Jung, B. Moon, "Toward minimal restriction of genetic encoding and crossovers for the two-dimensional Euclidean TSP". *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 6, Dec., pp. 557 - 565, 2002.

[15] T. Motoki, "Calculating the expected loss of diversity of selection schemes," *Evolutionary Computation*, vol. 10, no. 4, pp. 397-422, 2002.

[16] Research Group Combinatorial Optimization, "TSPLIB", [Online Repository], Aug. 2008, [2009 Apr 17], Available at HTTP:
http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/