Keep-Best Reproduction: A Selection Strategy for Genetic Algorithms

Kay Wiese and **Scott D. Goodwin**
Department of Computer Science, University of Regina
Regina, Saskatchewan, S4S 0A2, Canada
Phone: (306)-585-5210, Fax: (306)-585-4745
Email: wiese@cs.uregina.ca, goodwin@cs.uregina.ca

Key Words: Selection Strategies, Applications

Abstract

This paper presents an empirical study of a new intermediate selection strategy for genetic algorithms for constraint optimization problems. In a standard genetic algorithm the children replace their parents. The idea behind this is that both parents pass on their good genetic material to their children. In practice however, children can have worse fitnesses than their parents. We therefore propose another intermediate selection step, which we will call *keep-best reproduction*, which works as follows: After two parents are chosen for reproduction, the crossover operator is applied with its probability setting. Then the newly created offspring are evaluated. The worst offspring is replaced by the best parent. This makes sure that new genetic information is entered into the gene pool in the form of the best child, as well as ensuring that good previous genetic material is being preserved in the form of the best parent. Then mutation is applied to these strings with a certain probability and the resulting encodings are inserted into the new population. We will demonstrate the superiority of keep-best reproduction on several instances of the traveling salesman problem. Keep-best reproduction not only finds better solutions, but also finds better solutions faster than the standard generational replacement.

1 Introduction

Many problems in artificial intelligence and simulation can be described in a general framework as a *constraint satisfaction problem* (CSP) or a *constraint optimization problem* (COP). Informally a CSP (in its finite domain formulation) is a problem composed of a finite set of *variables*, each of which has a finite *domain*, and a set of *constraints* that restrict the values that the variables can simultaneously take. For many problem domains however not all solutions to a CSP are equally good. For example in the case of *job shop scheduling* different schedules which all satisfy the resource and capacity constraints can have different makespans (the total time to complete all orders), or different inventory requirements. So in addition to the standard CSP, a constraint optimization problem has a so-called *objective function* f which assigns a value to each solution of the underlying CSP. A global solution to a COP is a labelling of all its variables, so that all constraints are satisfied, and the objective function f is optimized.

Since it usually takes a complete search of the search space to find the optimum f value, for many problems global optimization is not feasible in practice. That is why COP research has focused on *local search* methods that take a candidate solution to a COP and search in its local neighborhood for improving neighbors. Such techniques include iterative improvement (hill climbing), threshold algorithms [3], simulated annealing [1, 10], taboo search [4, 5, 6], and variable depth search. Since these methods are only searching a subset of the search space, they are not *complete*, i.e., are not guaranteed to return the overall optimum. Another optimization technique is *genetic algorithms* (GAs). Genetic algorithms were originally designed to work on bitstrings. These bitstrings encoded a domain value of a real valued function that was supposed to be optimized. They were originally proposed by Holland [9]. More recently, researchers have focused on applying GAs to combinatorial optimization problems, including constraint optimization problems such as the travelling salesman problem (TSP) and the job shop scheduling problem (JSSP).

This paper discusses a new intermediate selection strategy, which we call *keep-best reproduction.* We discuss the results of an empirical study of the application of this new selection method to a genetic algorithm used to solve the travelling salesman problem. We compare our work to the standard selection technique, and demonstrate the superiority of keep-best reproduction.

In section 2, we look at the traditional selection step and its shortcomings and propose a new intermediate selection step, called *keep-best reproduction*, that eliminates the problem of loss of good genetic material. Section 3 presents the results of a study that compares traditional selection and keep-best selection on different sized travelling salesman problems. After our conclusion in section 4, we discuss some ideas and thoughts that are subject to further research in section 5. Section 7 contains a brief biography of the authors.

```
Top Level Genetic Algorithm
===================

Initialize a population of chromosomes;
Evaluate the chromosomes in the population;

while (stopping criteria not reached) do
   for i=1 to sizeof(population)/2 do
      select 2 chromosomes for recombination;
      apply crossover operator to them;
      apply mutation operator to them;
      evaluate the new chromosomes;
      insert the new chromosomes into
      the next generation;
      i = i + 1;
   endfor
   update stopping criteria;
endwhile
```

Figure 1: Top Level Genetic Algorithm

2 Keep-Best Reproduction: How to Get Better Results Through an Intermediate Selection Strategy

Most of the research since 1985 on applying GAs to the travelling salesman problem has focused on designing new representations and genetic operators and comparing their performance [7, 8, 11, 13, 12]. Examples of different representations for the TSP are *direct representation*, *ordinal representation*, and *adjacency representation* [8]. We used direct representation for our implementation, where a list **2 4 5 1 3 6** means start at city 2, then go to city 4, then to city 5, and so on. Some of the operators discussed in the literature for direct representation include *partially mapped crossover* (PMX) [7], *order crossover* (OX) [2], *cycle crossover* (CX) [11], and *edge recombination operator* [13]. We have used PMX in our study.

Surprisingly, not much research has been published on the effect of different selection strategies in this problem domain. Most researchers have used roulette wheel selection, a global selection strategy, and either generational replacement or steady-state reproduction. Figure 1 displays a typical standard GA that uses a global parent selection mechanism and generational replacement.

After selection, the selected individuals (parents) undergo crossover and mutation. The newly created individuals (children) are then inserted into the next generation. Crossover and mutation however can and usually will produce at least one child, whose fitness is lower than the fitness of at least one parent. Let us briefly illustrate this with a small example of a binary GA. Assume we are supposed to optimize the real function $f : [0,15] \to \mathbb{R}, f(x) = x^2$ using a GA. Of course we know that 15 is the domain value that optimizes f, since f is monotone increasing over $\mathbb{R}^+$. The GA does not know this however. Let us have a look at what can happen when we apply crossover to the following two individuals x_1^i and x_2^i, who we assume were chosen by roulette wheel selection in generation i:

$$x_1^i = 0\,1 \mid 1\,0 \tag{1}$$
$$x_2^i = 1\,0 \mid 0\,1 \tag{2}$$

Applying standard one point crossover at the crossover site 2, indicated by "|", yields the following children x_1^{i+1} and x_2^{i+1}.

$$x_1^{i+1} = 0\,1 \mid 0\,1 \tag{3}$$
$$x_2^{i+1} = 1\,0 \mid 1\,0 \tag{4}$$

It is clear that x_1^{i+1}'s fitness is worse than the fitnesses of either parent.

Genetic Algorithm with Keep-Best Reproduction

====================

```
Initialize a population of chromosomes;
Evaluate the chromosomes in the population;

while (stopping criteria not reached) do
   for i=1 to sizeof(population)/2 do
      select 2 chromosomes for recombination;
      apply crossover operator to them;
      apply mutation operator to them;
      evaluate the new chromosomes;
      compare the parent's fitnesses and
      remember the best parent;
      replace the offspring chromosome with
      lower fitness by the best parent chromosome;
      i = i + 1;
   endfor
   update stopping criteria;
endwhile
```

Figure 2: Keep-Best Reproduction

In order to preserve good genetic information as well as to introduce new, good genetic information into the population, we propose the following intermediate selection strategy: Keep only the best of the 2 offspring chromosomes and replace the other by the best parent. Since this ensures that both the best offspring and parent chromosome are kept, we call this technique *keep-best reproduction*. Alternatively we could have chosen to keep the 2 best out of the set consisting of the two parents and the two offspring. There is a potential danger in this approach. In case that the parents frequently have better evaluations than the offspring, the GA would not make significant progress, using the same strings over and over again. The other case where both offspring would have better evaluations than the parents is not likely to happen too often. The standard selection strategy, which always keeps both offspring, is shown to have inferior performance on our test problems, so the likelihood of consistently generating offspring that are both superior is highly unlikely. Figure 2 shows a modification of algorithm 1 that incorporates keep-best reproduction. In the following section we discuss the results of an empirical study of keep-best reproduction versus standard generational replacement.

3 Comparison of Traditional Selection and Keep-Best Reproduction

We have conducted a large number of tests on asymmetric travelling salesman problems. We have used 33, 50 and 100 cities. The cost between two cities was a random integer number between 0 and *maxcost*, where *maxcost* was set to 100 times the number of cities. Our implementation is based on the algorithm displayed in figure 2. The parent selection was done via roulette wheel selection. The mutation operator was a simple swap operation, that picks two random locations in the tour, and exchanges the two cities in those locations. The crossover operator we used was the partially mapped crossover (PMX). A detailed description of PMX can be found in [7]. No explicit elitism was used. Elitism is a technique applied by some GA researchers and practitioners to ensure that the top x percent of individuals are simply copied into the next generation. Elitism can speed up GAs but can also lead to premature convergence because of the dominance of these "super-individuals". To be able to study the effects of keep-best reproduction with fewer variables, we decided to not include elitism in our study. The fitness function we used was $f_i = c_{max} - c_i$, where c_i is the actual tour cost of individual i and c_{max} is the maximum cost in the population.

We have run tests for population sizes of 200, 400, 600, 800, 1000, and 1500. Both GAs were fine-tuned towards optimal performance, that is the crossover and mutation rates (probabilities) were set in order to yield optimal results. The data that was logged for each run includes the selection technique used, the generation number, the number of cities, the population size, the crossover probability, the mutation probability, the minimum cost of a tour in the population, the maximum cost, the average cost, and the minimum, maximum, and average fitnesses as well as the sum of all fitnesses. 4320 different runs were logged for the varying problem sizes, population sizes, selection techniques, and operator probabilities.

Table 1, 2, and 3 show the results of optimal runs. The first column indicates the population size. The second column contains the lowest cost, the highest cost, and the average cost of tours contained in the initial population. Column 3 and 4 contain the lowest cost, the highest cost, and the average cost of tours contained in the population after 300 generations. They also contain the values of the operator probabilities that yielded optimal results. The first number is the crossover probability, the second num-

33 cities	Initial	Standard	Keep-Best
200	42,634	11,333	9,383
	68,525	22,145	16,006
	55,960	13,307	9,459
		0.9, 0.09	1.0, 0.1
400	42,351	10,383	7,895
	70,388	29,087	15,896
	55,792	16,024	8,483
		0.8, 0.05	0.4, 0.5
600	39,402	9,598	8,245
	70,470	17,796	17,415
	55,529	9,662	9,403
		0.7, 0.01	1.0, 0.7
800	35,153	10,203	8,494
	71,062	47,160	19,398
	55,672	25,770	9,476
		0.7, 0.02	1.0, 0.6
1000	35,153	8,770	8,741
	71,062	41,673	17,296
	55,695	20,796	8,775
		0.6, 0.08	0.6, 0.1
1500	35,153	10,327	8,723
	72,540	20,860	22,083
	55,566	11,074	10,158
		0.3, 0.01	0.6, 0.7

Table 1: Best results after 300 generations for the 33 city problem

50 cities	Initial	Standard	Keep-Best
200	99,947	30,013	19,333
	155,067	47,835	27,727
	124,780	31,151	19,551
		0.4, 0.09	0.5, 0.3
400	88,600	27,431	21,956
	155,067	47,487	37,643
	124,497	29,639	22,938
		0.6, 0.07	0.2, 0.5
600	88,600	26,155	19,972
	155,067	42,151	40,812
	124,367	28,024	23,468
		0.6, 0.05	0.4, 0.9
800	88,600	29,037	20,544
	155,067	41,509	38,000
	124,267	29,399	22,301
		0.5, 0.02	0.7, 0.7
1000	88,600	25,618	19,788
	155,067	57,526	32,917
	123,879	35,711	20,383
		0.6, 0.09	0.2, 0.4
1500	88,600	29,507	18,543
	156,143	44,069	36,372
	123,934	31,044	22,385
		0.4, 0.04	0.8, 0.8

Table 2: Best results after 300 generations for the 50 city problem

ber is the mutation probability. For example, for the 50 city problem and a population size of 800 the tour with minimum cost found after 300 generations using the standard generational replacement had a cost of 29,037. This was achieved with a recombination rate of 0.5 and a mutation rate of 0.02.

By examining the data in the tables we see that keep-best reproduction significantly improves the results obtained with the genetic algorithm. This becomes even more evident as the problem size increases. For the 100 city problem the cost of the cheapest tour found by the standard selection is between 55% and 68% higher than the tour-cost found by keep-best reproduction. For the 50 city problem these percentages range from 25% to 59%. In the case of the smallest problem with 33 cities we find these percentages to be significantly lower. They range from 0.33% to 31%. The 0.33% is with a population size of 1000, where the minimum cost for a tour found is 8,770 with standard selection and 8,741 with keep-best reproduction. Comparing these results with the best tours found using other population sizes, it seems that the standard technique found an exceptionally good tour for population size 1000. We consider this to be an atypical result.

In the literature we have studied [7], [8], and [12], the recombination rate was typically set between 50% and 60%. No explicit mutation was used in either publication. We have used mutation for our implementation, and found mutation rates of up to 10% useful for the standard selection strategy. With standard selection and replacement, higher mutation rates disrupt good schemata too frequently. If these disrupted chromosomes of inferior quality are being inserted into the next generation, the GA will usually converge slower or not at all. With keep-best reproduction we were able to speed up the convergence of the GA by using higher mutation rates, as one can see from the mutation rates in table 1, 2, and 3. This should not come as a surprise. While with the standard selection strategy, mutation is performed and the mutated chromosomes are inserted into the next generation, keep-best reproduction only keeps the best child. In case mutation lowers the fitness of the offspring, there is always the good genetic material of the best parent that is kept. So higher mutation rates are not as disruptive as with the standard selection strategy. On the other hand, without mutation keep-best reproduction very rapidly converges to local optima of low quality. The higher mutation rates help steer the GA away from these inferior local optima. One should note, however, that keep-best reproduction is not the equivalent of an iterative improvement or random search. Mutation rates of 100% are not beneficial. Optimal

100 cities	Initial	Standard	Keep-Best
200	430,672	151,374	97,449
	581,774	192,703	120,032
	499,398	162,810	102,125
		0.5, 0.09	0.6, 0.6
400	430,672	145,714	86,891
	581,774	185,052	111,588
	500,781	154,692	88,335
		0.6, 0.02	1.0, 0.5
600	425,016	128,600	81,115
	600,179	182,331	116,207
	500,520	136,592	90,410
		0.5, 0.08	0.8, 0.9
800	413,272	131,597	81,775
	600,420	178,439	108,604
	500,571	137,883	84,236
		0.3, 0.1	0.9, 0.4
1000	413,272	134,433	81,767
	600,420	181,588	113,471
	500,416	143,040	85,686
		0.4, 0.05	1.0, 0.7
1500	413,272	124,668	76,451
	600,420	176,807	118,576
	500,528	135,370	84,666
		0.5, 0.04	0.7, 0.8

Table 3: Best results after 300 generations for the 100 city problem

results were typically obtained with 40% to 90% for 100 city problems. We also find that higher recombination rates are beneficial when keep-best reproduction is used. In table 3 the recombination rates that yielded optimal results were between 60% and 100%. With standard selection, this rate was significantly lower, between 30% and 60% for the 100 city problem. A remark on crossover probabilities: Crossover with keep-best reproduction produces one child and clones one parent. A crossover probability of for example 60% is thus equivalent to a crossover probability of 30% with standard selection, since only 30% of the chromosomes in the new population were created by crossover. However we need to create two children in every crossover step. Thus a crossover rate of 60% for keep-best reproduction means that $0.6 \times sizeof(population)$ new chromosomes are created through crossover in each generation, but only $0.3 \times sizeof(population)$ are inserted into the new generation.

4 Conclusion

We have proposed a new intermediate selection strategy that compares parent orderings and child orderings, and keeps the best parent as well as the best child. We call this selection strategy *Keep-Best Reproduction.* Keep-best reproduction outperforms the standard generational replacement technique significantly on our test problems, especially as the problem size increases. The collected data also demonstrates that with keep-best reproduction we are able to increase the genetic operator probabilities, especially the mutation probability. With standard selection, a high mutation rate usually has a negative effect on the performance of the GA, because of schema disruption. Our selection strategy benefits from higher mutation rates, since we always keep the best parent tour, thus limiting the disruptive effect that mutation can have.

5 Further Research and Discussion

Current research

This paper describes initial research on a GA based system for constraint optimization. From initial analysis of the data collected we made the following observations: Even with significantly smaller population sizes, keep-best reproduction finds better solutions than standard selection with much larger populations. This means that a better solution can be found with less function evaluations and thus with less total computing time. Keep-best reproduction converges faster, yielding better results faster, thus leading to another reduction in total computing time. This fact was not revealed in tables 1, 2, and 3. While we only reported the results after 300 generations, in some cases keep-best reproduction has converged after as little as 180 generations. This faster conversion is however evident when we look at the average tour cost in the population after 300 generations, where keep-best's population is more converged than standard selection's. The optimal results found with keep-best reproduction have significantly different operator settings than the standard technique. Another observation we made is that keep-best reproduction is less susceptible to a change in crossover probability and mutation probability. All the above observations are only preliminary. We are currently investigating about 22 Mbyte of collected raw data from GA runs. If the above observations prove to be true, we would also have a more robust GA for COP, i.e., the performance would not so heavily depend on parameter settings such as population size, crossover probability, and mutation probability.

Further Research

We need to test keep-best reproduction on some benchmark problems as well as for larger travelling salesman problems. We are confident however, that on these problems we will still be able to get better results than the standard selection.

It would be interesting to see how other GA techniques such as elitism and other genetic operators affect the performance of keep-best reproduction versus standard selection. Also we will compare our work with other advanced selection strategies.

We want to built a distributed genetic system for constraint optimization. We will implement a distributed GA with keep-best selection on a MIMD system with 64 transputer nodes. We will compare standard selection and keep-best selection for this parallel implementation and are confident that we will achieve similar good results as in the sequential case.

6 Acknowledgements

The authors would like to thank the Institute of Robotics and Intelligent Systems (IRIS/PRECARN) for funding this research. This research is part of the Intelligent Scheduling Project IC-6. Also the authors would like to take the opportunity to thank the referees for their insightful comments that helped to improve the quality of this paper.

7 Biography

Kay Wiese is a PhD student at the University of Regina. He has worked under the supervision of Dr. Scott Goodwin since 1995. Originally his work focused on object oriented constraint solving and optimization. Mr. Wiese's background is in computer algebra and massively parallel algorithms. He has done his Master's work on parallel lattice basis reduction at the Universitaet des Saarlandes, Germany. Since 1996 both Kay Wiese and Scott Goodwin are interested in genetic algorithms as a means of combinatorial and constraint optimization. Their research objective for the near future is to build a parallel/distributed genetic system for combinatorial optimization. Dr. Goodwin has received his PhD from the University of Alberta in 1991. His PhD work focused on statistically motivated defaults. He is now an Associate Professor of Computer Science at the University of Regina. Currently he is on a one year sabbatical which he will spend at the University of Waterloo and Griffith University, Australia.

References

[1] V. Cerny, "Thermodynamical Approach to the Traveling Salesman Problem", *Journal of Optimization Theory and Applications 45*, pp. 41–51, 1985.

[2] L. Davis, "Job Shop Scheduling with Genetic Algorithms", *Proceeding of the 1st ICGA*, pp. 136–140, 1985.

[3] G. Dueck and T. Scheuer, "Threshold Accepting; a General Purpose Optimization Algorithm", *Journal of Computational Physics 90*, pp. 161–175, 1990.

[4] F. Glover, "Tabu Search - Part I", *ORSA Journal on Computing 1*, pp. 190–206, 1989.

[5] F. Glover, "Tabu Search - Part II", *ORSA Journal on Computing 2*, pp. 4–32, 1990.

[6] F. Glover, E. Taillard, and D. De Werra, "A User's Guide to Tabu Search", *Annals of Operations Research 41*, pp. 3–28, 1993.

[7] D.E. Goldberg, R. Lingle, Jr., "Alleles, Loci, and the Traveling Salesman Problem", *Proceedings of the 1st ICGA*, pp. 154–159, 1985.

[8] J.J. Grefenstette, R. Gopal, B. Rosmaita, D. van Gucht, "Genetic Algorithms for the Traveling Salesman Problem", *Proceedings of the 1st ICGA*, pp. 160–168, 1985.

[9] J.H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI, University of Michigan Press.

[10] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, "Optimization by Simulated Annealing", *Science 220*, pp. 671–680, 1983.

[11] I.M. Oliver, D.J. Smith, and J.R.C. Holland, "A Study of Permutation Crossover Operators on the Traveling Salesman Problem", *Proceedings of the 2nd ICGA*, pp. 224-230, 1987.

[12] T. Starkweather, S. McDaniel, et. al, "A Comparison of Genetic Sequencing Operators", *Proceedings of the 4th ICGA*, pp. 69–76, 1991.

[13] D. Whitley, T. Starkweather, and D'Ann Fuquay, "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator", *Proceedings of the 3rd ICGA*, pp. 133–140, 1989.