A Little Respect (for the Role of Common Components in Heuristic Search)

Stephen Chen[1]
1 School of Analytic Studies and Information Technology, York University 4700 Keele Street, Toronto, Ontario M3J 1P3
sychen@yorku.ca
WWW home page: http://www.atkinson.yorku.ca/~sychen

Abstract. A search process implies an exploration of new, unvisited states. This quest to find something new tends to emphasize the processes of change. However, heuristic search is different from random search because features of previous solutions are preserved – even if the preservation of these features is a passive decision. A new parallel simulated annealing procedure is developed that makes some active decisions on which solution features should be preserved. The improved performance of this modified procedure helps demonstrate the beneficial role of common components in heuristic search.

1 Introduction

Heuristic search techniques can be analyzed by focusing on their "search operators" and their "control strategies". A search operator creates new solutions by making some (usually small) changes to existing solutions. All of the remaining aspects of a heuristic search technique can be included as part of its control strategy [15] – which search operator(s) to apply; which solutions to keep, change, or remove; and when to stop.

The role of the (local) search operator is to create a new candidate solution. Depending on the control strategy, (small) random changes may be sufficient. However, search operators should ideally perform two tasks in creating a new solution from the existing solution(s) – identify both the superior solution parts that are worth keeping and the weaker solution parts that should be changed. A search operator that performs these tasks and which produces better than random changes may be able to improve the performance of any control strategy/heuristic search technique that it is used with.

Examples of heuristic search techniques that normally use random search operators are hill climbing and simulated annealing [10]. In hill climbing, random changes are applied (e.g. two-opt swaps for the Travelling Salesman Problem) and

the control strategy accepts only improving changes. In simulated annealing, the control strategy accepts non-improving changes probabilistically based on the temperature which is adjusted according to a cooling schedule. Attempts to improve the performance of simulated annealing often focus on finding better cooling schedules [9].

Another heuristic search technique that emphasizes the features of the control strategy over the design of the search operator is tabu search [6]. In tabu search, the control strategy causes escapes from local optima by making certain operations tabu. However, this tabu list of restricted operators is not designed to improve the isolated performance of the search operators. Specifically, tabu search and simulated annealing can use the same search operators [8], so both of these heuristic search techniques are potential beneficiaries of improved search operators.

In the definitions of other heuristic search techniques like genetic algorithms [7], the search operators and the control strategy are both included. The primary search operator for genetic algorithms is crossover, and the three mechanisms of crossover are respect, transmission, and assortment [12]. Although assortment /recombination is seen as the "overt purpose" of crossover [14], respect/the preservation of common components is also an important feature [4]. In particular, this feature can be used to specify which solution parts should be kept – a useful complement to the focus that most search operators have on what to change.

In the meta-heuristic of memetic algorithms [13], it has previously been shown that respect/the preservation of common components can be a beneficial feature [5]. Since the local optima of a globally convex search space share many similar features [2][11], it is reasonable to restart the heuristic search technique in a neighbourhood near good local optima. However, the potential benefits of respect for single-parent search operators has not been analyzed in isolation.

To summarize, it is hypothesized that a search operator will be more effective if it actively performs both tasks of choosing what to keep and choosing what to change. Ideas for how to develop these operators will be taken from genetic algorithms in sections 2 and 3. These ideas will be transferred into simulated annealing in section 4. Experimental results will be developed and presented in sections 5 through 7. These results are discussed in section 8 before conclusions are drawn in section 9.

2 Background

The three primary features of a genetic algorithm (GA) are a population of solutions, fitness-based selection, and the crossover search operator [7]. In crossover, there are three mechanisms: "respect", "transmission", and "assortment" [12]. The principle of respect is that the common components of two parent solutions should be preserved in the offspring. Transmission states that all components in the final offspring should have come from one of the two parents, and assortment is the equivalent of recombination – parents with two non-competing traits should be able to produce an offspring with both features.

The two mechanisms that are unique to multi-parent operators are recombination and respect – it takes multiple parents to have/identify common and uncommon components that can be recombined or preserved. It is generally assumed that the advantage of crossover is its ability to assemble better offspring solutions by combining the superior solution parts of two parents [7][14]. However, attempts to transfer this advantage to other search operators and heuristic search techniques have led to mixed results and comments like "crossover can be compared to the very unpromising effort to recombine animals of different species" [16].

Focusing instead on the mechanism of respect, the Commonality Hypothesis [4] suggests that the advantage of the crossover search operator is its ability to leverage the knowledge accumulated in previous solutions and to use this knowledge as a foundation for further explorations. In particular, it has previously been shown that many combinatorial optimization problems have "globally convex" fitness landscapes that cause the best solutions to share many similarities [2][11]. By transferring respect to general search operators, it should be possible to allow other heuristic search techniques to more effectively explore globally convex fitness landscapes.

3 An Analysis of Respect and Recombination

Before adding respect to a search operator for simulated annealing, it is important to develop a functional model for how this mechanism generates benefits. To start, the performance of one-point, two-point, and uniform crossover are shown for the OneMax problem (where the fitness of a binary string solution is equal to its number of ones). The results presented in figure 1 are for 100 trials of a genetic algorithm with generational replacement where only solutions with a fitness higher than the average fitness of the entire population are allowed to mate.

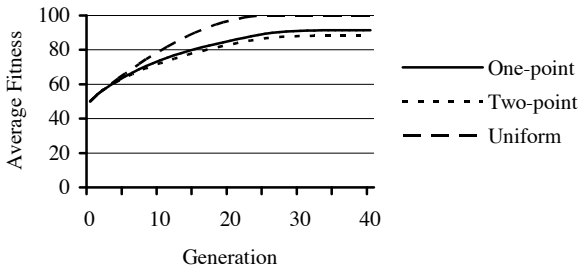

Fig. 1. Results for a 100 bit OneMax problem started from a random population of 100 solutions. 100 generations were used, but all trials converged within 40 generations. Average fitness refers to all solutions in the population and are averaged over 100 trials

One interpretation for the superior performance of uniform crossover is that its larger number of crossing sites allow greater opportunities for recombination.

However, it is shown in figure 2 that the uncommon components being recombined have a below average fitness compared to the solution as a whole and to the above average fitness of the common components preserved by all forms of crossover. In a binary solution space, an uncommon component is necessarily a 1 in one parent and a 0 in the other. Since these uncommon components will have an average fitness of 0.5 (for the OneMax problem), recombination is effectively random search in this instance.

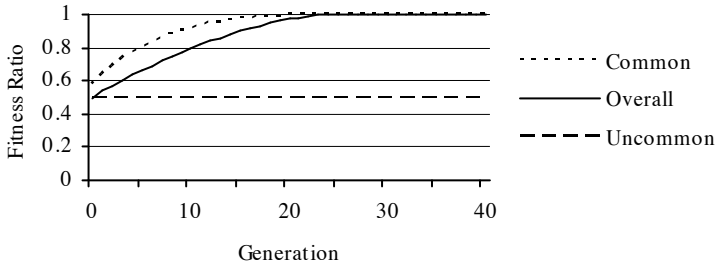

Fig. 2. Results for a 100 bit OneMax problem started from a random population of 100 solutions. 100 generations were used, but all trials converged within 40 generations. Average fitness refers to all solutions in the population and are averaged over 100 trials

An alternative interpretation for the superior performance of uniform crossover is that it is a more efficient search operator than one-point and two-point crossover. Although all three operators preserve common components, uniform crossover searches with the most "unbiasedness" [1]. Focusing on the common components, the concept of "genetic repair" [1] provides a worthy explanation for why the accumulation of common components can be a beneficial part of the search process – these common components (i.e. the population centroid) are fitter than the individual solutions (see the common ratio in figure 2). Unfortunately, genetic repair gives little guidance for the design of search operators in discrete problem domains.

Recalling the hypothesis that a search operator may be more effective if it actively performs the two tasks of identifying the superior solution parts worth keeping and the weaker solution parts to change, it is clear from figure 2 that preserving common components can do both. This benefit of preserving common components is explicitly demonstrated in figure 3. The uncommon components of two parent solutions necessarily have an average fitness of 0.5 (per component), and this will be below the average component fitness of two above-average parents (0.7 in figure 3). Therefore, the average fitness of the remaining common components must be even greater (0.83 in figure 3). The effectiveness of a search operator should benefit from preserving common components because a solution improvement is more likely when changes are applied to the less fit uncommon components.

```
Parent 1:     1 0 1 1 0 1 0 1 1 1
Parent 2:     1 1 0 1 0 1 1 1 0 1

Common:       1     1 0 1   1   1

Uncommon 1:     0 1       0   1
Uncommon 2:     1 0       1   0
```

Fig. 3. In the OneMax problem, an improvement occurs when a 0 is turned into a 1. A random change applied to either parent above leads to an improvement only 30% of the time. However, a "respectful" change that is applied to an uncommon component will lead to an improvement 50% of the time [4]

4 Simulated Annealing and Respect

The control strategy for simulated annealing (SA) is designed to allow probabilistic escapes from local optima. Assuming a minimization objective, the simulated annealing process can be visualized as a ball rolling downhill through a landscape of peaks and valleys. Depending on how much "energy" is in the ball, it has the ability to "bounce out" of local minima. When the temperature/energy approaches zero, the ball will come to rest in a final minimum – ideally the global minimum if the cooling schedule has been slow enough.

This control strategy does not specify how the "ball" will escape from local minima – it can climb any valley wall with equal probability. If local optima are randomly distributed throughout the search space, then this standard implementation of SA will be well suited. However, the local optima for many combinatorial optimization problems exhibit a "big valley" clustering – random local optima are more similar than random solutions, the similarities among local optima increase with their quality, and the global optimum is in the "centre" of the cluster of local optima [2][11].

The standard control strategy for simulated annealing is not as well suited for problems with "big valley" fitness landscapes. When escaping from local minima, simulated annealing makes no attempt to determine if it is climbing an "interior" wall (that is between the current solution and the global optimum) or an "exterior" wall (that is between the current solution and the perimeter of the big valley). (See figure 4.) Although the proof of convergence for simulated annealing does not require this insight, the practical (time-based) performance of an SA implementation may be affected.

Simulated annealing can be modified for big valley fitness landscapes by adding the mechanism of respect from genetic algorithms. Having features from both SA and GA, the new modified procedure is called SAGA. SAGA provides the multiple solutions required by respect to identify common components by using two parallel runs. (See figure 5.) Using an elitist SA approach where the best solution visited during each temperature cycle is used as the starting point for the next temperature cycle, common components are recorded from the best solution of each run.

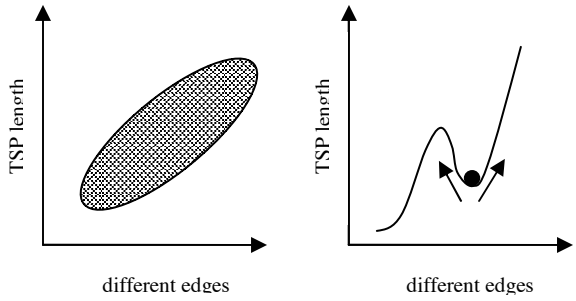

Fig. 4. In a typical distribution of two-opt minima for the TSP, the best solutions have more common edges [2]. When escaping from a local minimum, a change that produces additional different/uncommon edges may be moving away from the better solutions found at the centre of the big valley

The expectation that SAGA will perform better than a normal implementation of simulated annealing is based on the two previous observations. In figure 3, it can be seen that a change applied to an uncommon component is more likely to lead to an improvement. In figure 4a, it is shown that it is undesirable to have a large number of solution differences. By changing components that are already different, a respectful search operator is less likely to create additional solution differences that will lead the search process away from the centre of the big valley.

5 A Base SA Application for the TSP

To better isolate/emphasize the effects of the modified search operator, a relatively simple base SA application for the TSP (BaseSA) was used. BaseSA starts from a random solution, uses a geometric cooling schedule ($\mu = 0.9$) with $n = 500$ temperature cycles, applies two million two-opt swaps per cycle, and returns to the best found solution at the end of each temperature cycle. Since a fixed number of temperature cycles are used, the temperature is not decreased if the best solution does not improve during that cycle. To determine the initial temperature, a preliminary set of 50 two-opt swaps is used where only improving steps are accepted. At the end of this set, the initial temperature starts at the average length of the attempted backward steps divided by ln(0.5) (i.e. an average backward step has a 50% chance of being accepted).

```
SAGA procedure                                          line
Begin                                                    1
  create initial random solution A                       2
  create initial random solution B                       3
  set initial temperature                                4
  For i = 1 to n                                         5
    record common components of solutions A and B        6
    For j = 1 to m                                       7
      apply search operator to solution A                8
      apply search operator to solution B                9
    End                                                  10
    set solution A to best solution for A since line 6   11
    set solution B to best solution for B since line 6   12
    update temperature                                   13
  End                                                    14
  return best of solution A or B                         15
```

Fig. 5. Pseudocode for SAGA. The two parallel runs (A and B) use the same cooling schedule. Information on common components is shared on line 6 before each temperature cycle (lines 7-10). See sections 5 and 6 for the values of n and m

6 Developing SAGA for the TSP

SAGA has been developed for the TSP by using two parallel runs of BaseSA. Each instance of BaseSA uses all of the above parameters except that only m = 1,000,000 two-opt swaps are now used during each annealing stage. This modification ensures that SAGA uses the same overall number of two-opt swaps (and thus a similar amount of computational effort) as the benchmark implementation of BaseSA.

A two-opt swap changes two edges in a (single) solution. To implement the mechanism of respect so that common components can be preserved, SAGA records the common edges of the two parallel BaseSA runs at the beginning of each temperature cycle (see figure 5). The respectful two-opt operator will then select one uncommon edge and one random edge (which could be either common or uncommon). This preference to change uncommon components will help preserve common components which should help direct the overall search process towards the centre of the "big valley" of local minima.

The only parametric difference between SAGA and BaseSA is an additional parameter to specify how often to apply the respectful two-opt operator versus the standard two-opt operator. This parameter was chosen from the 11 probabilities shown in table 1 by conducting a set of tuning experiments on PCB442. Measuring

the percent difference above the known optimal for the final solution in ten SAGA trials, it was determined that the respectful two-opt operator should be used 90% of the time (with random two-opt swaps being used for the remaining 10%). This parameter is used in all of the experiments presented in section 7.

The results of the above tuning experiments also indicate that the improvements in SAGA are not due strictly to the use of two parallel runs – a result that is more fully explored in [3]. With a 0% probability of using the respectful two-opt operator, SAGA is essentially two (smaller and less effective) independent runs of BaseSA. Compared to these two independent runs, the performance of SAGA tends to improve with the increased use respect. The key exception to this trend is for a probability of 100%. It appears that SAGA benefits from divergent changes just like traditional genetic algorithms benefit from mutation.

Table 1. Probability that the first-edge is specifically selected to be an uncommon edge. Values represent average percentage above optimal for 10 SAGA trials – n = 300 cycles, 60 million total two-opt swaps. Average performance of 10 BaseSA trials is provided for comparison

Probability	PCB442
0.0	4.21
0.1	3.60
0.2	3.92
0.3	3.68
0.4	3.76
0.5	3.49
0.6	3.01
0.7	3.17
0.8	3.08
0.9	**2.79**
1.0	3.40
BaseSA	3.79

7 Experiments

Using the above parameters, thirty trials of BaseSA and SAGA were each run on 5 test TSP instances (dsj1000, d1291, fl1400, fl1577, and pr2392). The results for the experiments are shown in table 2. The respectful two-opt operator in SAGA has led to consistent and significant improvements compared to the benchmark implementation of BaseSA.

It is important to note that only the relative results or differential performance should be considered from the above experiments. A rather simple control strategy was used in BaseSA to help isolate/emphasize the effects of the search operator. As a controlled parameter, any strengths or weaknesses in the basic operation of BaseSA should be reflected equally in the performance of both BaseSA and SAGA. Therefore, the superior performance of SAGA as compared to BaseSA is a clear demonstration of the benefits of respect.

Table 2. Average percentage above optimal for 30 trials of BaseSA and SAGA with one billion total two-opt swaps performed. One-tailed t-tests show that the chance of the SAGA results being the same as the BaseSA results is less than 0.01% for all five instances

Instance	BaseSA		SAGA	
	avg.	std. dev.	avg.	std. dev.
dsj1000	4.54	0.74	2.27	0.39
d1291	8.87	1.41	3.12	1.12
fl1400	3.07	1.04	2.00	0.88
fl1577	6.47	1.58	0.64	0.55
pr2392	9.24	1.37	6.53	0.56

8 Discussion

A heuristic search technique is defined by its search operator(s) and its control strategy. Some are defined almost exclusively by their control strategy (e.g. hill climbing, simulated annealing [10], and tabu search [6]). In these cases where there are relatively few restriction placed on the search operator, a new design model for search operators represents a broad opportunity to improve the performance of many implementations. The proposed design model suggests that the performance of a search operator (and of the heuristic search technique that uses it) can be improved if it preserves common components and focuses its changes on the uncommon components.

The role of common components has been examined under many situations (e.g. in crossover operators [12] and memetic algorithms [5][13]). However, all of these previous situations involve populations of solutions, and the effects of preserving common components have not always been considered as favourable (e.g. premature convergence in genetic algorithms). The new model for search operators appears to be unique in that it applies respect to a (nominally) single-parent operator.

It is interesting to note that the respectful search operator would not likely be accepted as a crossover operator since it does not use the mechanism of assortment/recombination. For example, imagine a two-parent operator where an offspring is created by making a random mutation to an uncommon component. The offspring would not be a recombination of the parent components and nothing would have "crossed over" between them, so the respectful search operator would not be a recombination/crossover operator. Since the crossover search operator is a defining feature of genetic algorithms [7][14], the new model for search operators may be more suitable to evolution strategies where it appears that nothing similar to the new respectful search operators has been used [1].

9 Conclusions

The difference between heuristic search and random search is that heuristic search exploits/preserves some information from previous solutions. The proposed model for respectful search operators suggests that in globally convex search spaces, these preserved components should be common components. A search operator that

changes uncommon components can have a better than random chance of finding an improvement, and this increased effectiveness in the search operator may be able to improve the overall performance of a heuristic search technique.

References

1. Beyer, H.-G., Schwefel, H.-P.: Evolution Strategies: A comprehensive introduction. In Natural Computing, Vol. 1 (2002) 3-52
2. Boese, K.D.: Models for Iterative Global Optimization. Ph.D. diss., Computer Science Department, University of California at Los Angeles (1996)
3. Chen, S., Pitt, G.: The Coordination of Parallel Search with Common Components. In LNCS, Vol. 3533: Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. (2005)
4. Chen, S., Smith, S.F.: Introducing a New Advantage of Crossover: Commonality-Based Selection. In GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann (1999)
5. Chen, S., Smith, S.F.: Putting the "Genetics" back into Genetic Algorithms (Reconsidering the Role of Crossover in Hybrid Operators). In Banzhaf, W., Reeves, C. (eds.): Foundations of Genetic Algorithms 5. Morgan Kaufmann (1999) 103-116
6. Glover, F.: Tabu search Part I. In Operations Research Society of America (ORSA) Journal on Computing, Vol. 1 (1989) 109-206
7. Holland, J.: Adaptation in Natural and Artificial Systems. The U. of Michigan Press (1975)
8. Hoos, H.H., Stützle, T.: Local Search Algorithms for SAT: An Empirical Evaluation. In Journal of Automated Reasoning, Vol. 24 (2000) 421 - 481
9. Johnson, D.S., McGeoch, L.A.: The Traveling Salesman Problem: A Case Study in Local Optimization. In Aarts, E.H.L., Lenstra, J.K. (eds.): Local Search in Combinatorial Optimization. John Wiley and Sons (1997) 215-310
10. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by Simulated Annealing. In Science, Vol. 220 (1983) 671-680
11. Mühlenbein, H.: Evolution in Time and Space – The Parallel Genetic Algorithm. In: Rawlins, G. (ed.): Foundations of Genetic Algorithms. Morgan Kaufmann (1991)
12. Radcliffe, N.J.: Forma Analysis and Random Respectful Recombination. In Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann (1991)
13. Radcliffe, N.J., Surry, P.D. Formal memetic algorithms. In: Fogarty, T.C. (ed.): Evolutionary Computing: AISB Workshop. Springer-Verlag (1994)
14. Syswerda, G: Uniform Crossover in Genetic Algorithms. In Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann (1989)
15. Talukdar, S.N., de Souza, P.: Scale Efficient Organizations. In Proceedings of 1992 IEEE International Conference on Systems, Man and Cybernetics (1992)
16. Wendt, O., König, W.: Cooperative Simulated Annealing: How much cooperation is enough? Technical Report, No. 1997-3, School of Information Systems and Information Economics at Frankfurt University (1997)