To appear in the Notes of the IJCAI-01 Workshop on Stochastic Search.

Stochastic Tree Search: Where to Put the Randomness?

Wheeler Ruml[*]
Harvard University
ruml@eecs.harvard.edu

Abstract

In this short note, I argue against two commonly-held biases. The first is that stochastic search is applicable only to improvement search over complete solutions. On the contrary, many problems have effective greedy heuristics for constructing solutions, making a tree-structured search space more appropriate. The second is that stochastic tree search algorithms should explore the same space of decisions as systematic methods. Constructing search trees in the traditional manner, by choosing the default variable at the parent and valuing it differently at each child, makes sense for efficient complete search, but is not necessarily the best choice for incomplete methods. In an empirical study using the combinatorial optimization problem of number partitioning, I show that the opposite approach, choosing a different variable at each child and giving it the default value, can be a good choice for incomplete stochastic algorithms.

1 Stochastic Tree Search

A large number of papers have appeared in recent years (including at AI conferences such as IJCAI and AAAI) devoted to stochastic improvement search for optimization problems, in which an algorithm attempts to improve a complete but potentially suboptimal solution. Many of these 'local search' methods, such as tabu search or simulated annealing, are completely general and use as their only source of problem-specific information the ability to evaluate the objective function on a complete solution. Others, such as WalkSAT, take advantage of heuristic guidance in the form of a function that identifies variables that might be profitably changed. Improvement methods are often contrasted with complete search methods, which use techniques such as branch-and-bound or dynamic backtracking [Ginsberg, 1993] to systematically extend an empty solution in all possible ways, implicitly traversing a tree containing all possible solutions. When run to completion, such methods guarantee an optimal solution. But it is easy to over-generalize and assume that stochastic search is applicable only to improvement search. In fact, the call for papers for this workshop exhibits this tendency. It summarizes the field of interest as "stochastic local search algorithms, as well as randomised systematic search methods," leaving the impression that stochastic search applies only to local improvement and that tree-structured search spaces are best left to systematic methods (perhaps suitably randomized). Such a bias is ill-founded for two reasons. First, there are many problems for which heuristic guidance is available when the problem is considered as a tree. And second, as I will explain, trees provide an convenient avenue for attacking one of the main research questions in stochastic search: how to represent information learned during the search.

For many problems, effective heuristic guidance is available when viewing the problem as a tree search. This is evidenced by the fact that greedy constructive heuristics exist for many problems. Such methods can be viewed as incomplete tree searches that visit only a single leaf without backtracking [Korf, 1995]. Expanding a node in the tree corresponds to extending a partial solution, and the leaves of the tree correspond to complete solutions which can be evaluated according to the objective function. The greedy algorithm provides a way of choosing which component of the problem should be set next (with analogy to CSPs, call this variable choice) and what it should be set to (call this value choice). Systematic algorithms such as depth-first search and limited discrepancy search [Harvey and Ginsberg, 1995] can exploit this heuristic knowledge, and the fact that such algorithms can perform well on certain problems despite their fixed and non-adaptive search order indicates that this heuristic guidance is powerful indeed. Stochastic approaches are often inherently incomplete. But for large problems in which the complete tree cannot be enumerated, there is no reason to think that stochastic tree search algorithms could not do better than existing systematic methods.

Several stochastic tree searches have already been proposed. Simple unguided random probing was found effective on scheduling benchmarks by Crawford and Baker [1994]. Bresina's [1996] Heuristic-Biased Stochastic Sampling (HBSS) makes repeated probes into a tree, weighting its choice of child according to the value-ordering function. The Greedy Randomized Adaptive Search Procedure (GRASP) approach of Feo and Resende [1995] is actually

[*]Division of Engineering and Applied Sciences, 33 Oxford Street, Cambridge, MA 02138 USA. www.eecs.harvard.edu/~ruml.

a combination of heuristically biased probing with improvement search on the resulting leaf (see also Marchiori and Steenbeek [1998]). Juillé and Pollack [1998] use random tree probing as a value choice heuristic during a beam search into a tree, and Abramson proposed similar methods for game trees [1991].

Stochastic tree search provides an avenue for addressing one of the important active research questions in stochastic search: how to explicitly represent information learned during the search so that it can be used to guide future actions? (The collection of papers edited by Boyan, Buntine, and Jagota [2000] surveys current work in this area.) In local search, it is difficult to formulate a representation that can capture the past history of objective values because the search space is so unstructured. Tabu search methods implicitly represent regions of the search space which should not be explored. Boyan's [2000] STAGE system requires user-supplied subroutines that calculate features of solutions. It then uses them to generalize about good regions from which to start improvement search. Boese et al.'s [1994] Adaptive Multi-Start (AMS) uses solutions themselves as stand-ins to mark good regions of the search space, but this representation again requires a user-supplied combining function to implement generalization. Baluja's [1997] Population-Based Incremental Learning (PBIL) works only with binary problems, and hence can use a probability vector to represent learned information about which variables should be 0 or 1.

But in tree-based stochastic search, the tree itself can provide the geometry of the search space. As the work on discrepancy search shows [Harvey and Ginsberg, 1995; Korf, 1996; Walsh, 1997], one can usefully generalize across levels of the tree. I have been working on adaptive probing algorithms [Ruml, 2001] that learn at what depths of the tree one can trust the given child-ordering function. A stochastic framework is employed that chooses the heuristically-preferred child at each level according to the estimated probability that it is in fact the better choice. This varies according to the number of probes that have been performed and the data obtained. The ability to generalize about the child-ordering heuristic across the breadth of the tree provides a concise representation of the learned probing bias. (Adaptive probing may also help provide a principled grounding for algorithms such as HBSS, GRASP, and the Ant Colony Optimization work of [Dorigo and Gambardella, 1997], in which 'pheremone' accumulates to represent the combined information gathered by multiple search trials.)

The Squeaky Wheel Optimization (SWO) method of Joslin and Clements [1998] is another adaptive tree search method. In SWO, a greedy algorithm is used to construct solutions given an order in which to consider the variables. A variable choice function (similar to those used in improvement search) identifies variables that are poorly set in the resulting complete solution. Those variables are moved earlier in the ordering, usually resulting in their being handled better by the greedy heuristic. One can think of this technique as adapting the given variable choice heuristic.

Unlike many improvement search methods, stochastic tree search is not necessarily 'local'—it might visit very different solutions on consecutive iterations. But like improvement methods, this flexibility comes at the cost of incompleteness, and one must be alert to the possibility of expending redundant effort in the same part of the search space (a generalization of the 'local minima' of improvement search). Given the relative abundance of heuristic information in a tree-structured search space, and the way that the clear geometry of the tree provides a convenient form for representing learned information during search, stochastic tree search seems ripe for research attention.

2 Alternative Tree-Structured Search Spaces

One of the most fundamental questions in tree search is how to trade-off the information provided by the variable-choice heuristic and the value-choice heuristic. More simply put, what is the best way to structure the search? Complete systematic search methods typically trust the variable-choice heuristic completely. Children of a node vary in the value they assign to that variable; they can be ordered according to the value-choice heuristic. This results in a depth n tree with the branching factor depending on the number of values per variable. (Typically, the number of values is smaller than the number of variables.) This represents the smallest tree that completely enumerates the possibilities (size b^n). But efficiently enumerating all solutions is already out of the question for many real-world problems. In the context of incomplete search, it may in fact be more productive to acknowledge that the variable choice function could be fallible too. For some problems, it may be useful to consider an alternative tree representation in which one searches not over the value assignments but over the variable choices. Each child of a node would represent the choice of a different variable, which would be assigned the value chosen by the value choice heuristic. The children could be ordered according to the variable choice heuristic. From a conventional point of view, searching this tree is ludicrous. It is much larger than the conventional one, as it has a branching factor of $O(n)$, as well as depth n, for a size of $n!$, and it does not necessarily even include all possible solutions. But the solutions that it does contain may be of a higher quality, depending on the accuracy of the value choice heuristic. It is important to recognize that the choice of search space representation is not pre-ordained—the bias towards conventional representation may be inappropriate for incomplete methods.

2.1 Empirical Investigation

To demonstrate the relevance of this question, I have performed some experiments with the combinatorial optimization problem of number partitioning. The objective in a number partitioning problem is to divide a given set of numbers into two disjoint groups such that the difference between the sums of the two groups is as small as possible. It was used by Johnson et al. to evaluate simulated annealing [1991], Korf to evaluate his improvement to limited discrepancy search (ILDS) [1996], and Walsh to evaluate depth-bounded discrepancy search (DDS) [1997]. When the numbers are chosen uniformly over an interval, the difficulty of the problem depends on the relation between the number of digits in the numbers and the number of numbers. With few digits and

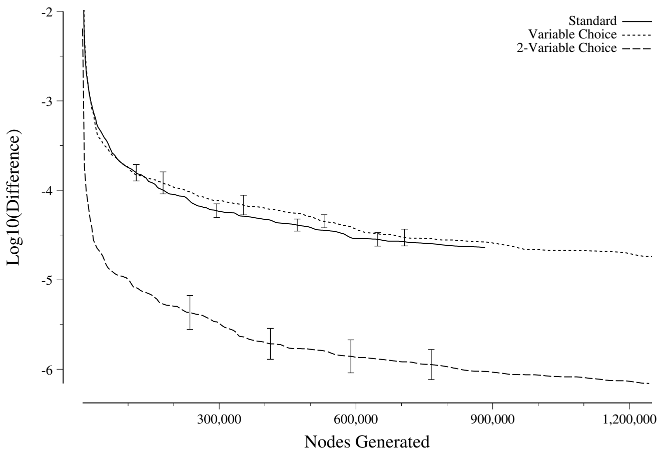

Figure 1: Searching the greedy spaces using random sampling.

many numbers, the probability of a partitioning with a difference of 0 and 1 increases [Karmarkar *et al.*, 1986]. This makes the tree search easier, as the search can terminate once such a partitioning is found. To encourage difficult search trees by reducing the chance of encountering a perfectly even partitioning [Karmarkar *et al.*, 1986], I used instances with 64 25-digit numbers or 128 44 digits numbers.[1] (Common Lisp, which provides arbitrary precision integer arithmetic, was used to implement the algorithms.) Results were normalized as if the original numbers had been between 0 and 1.

The Greedy Representation

We present results using two different representations of the problem. The first is a straightforward greedy encoding in which each variable specifies the partition to which a particular number will be assigned (either 0 or 1). Variable choice corresponds to choosing which number to assign, and the usual heuristic is to choose the largest remaining number. The usual value choice heuristic is to choose the currently smaller partition, as adding to the larger one merely exacerbates the partition difference. So the standard search space is one in which the largest remaining number is selected to be added to a partition and the smaller partition is tried first. The less-preferred option at each decision point is to add the selected number to the larger partition. The particular search algorithm used will determine when the less-preferred option will be chosen. The alternative search space always adds to the smaller partition, but all remaining numbers can be selected, with the options sorted from largest to smallest.

The easiest way to get a sense of the usefulness of a search space is to assess the average quality of the solutions. Figure 1 shows the performance of random probing when partitioning 128 numbers. This should give an overall impression of the distribution of solution quality in the search spaces. The vertical axis represents the partition difference, which the

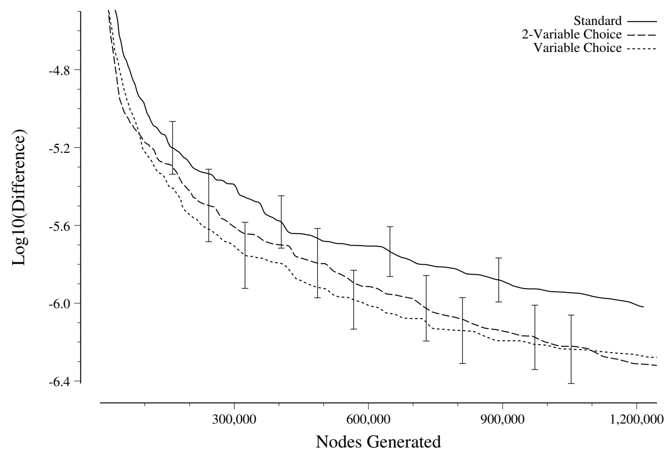

Figure 2: Searching the greedy spaces using HBSS.

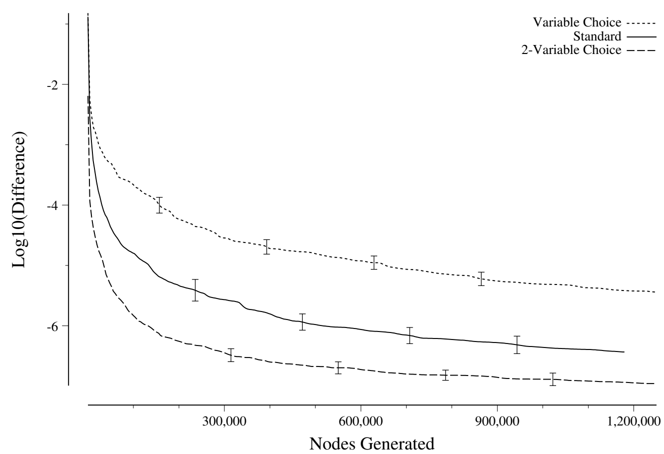

Figure 3: Searching the greedy spaces using adaptive probing.

algorithms are attempting to minimize. The solutions that are found in the standard value-choice search space are very similar to those found when searching over variable choices. (Error bars indicate 95% confidence intervals around the mean, derived using 20 instances.) Because the variable-choice search space always places numbers in the smaller partition, forcing some degree of balance, there are many poor solutions which exist in the standard search space but cannot be found in the variable-choice space. However, it seems that adding a random number to the smallest available bin is not much of an improvement over putting the largest number in a random bin. One can imagine restricting the range of generable solutions even more. A restricted version of the variable-choice space was also tested, in which only the largest two numbers were available. This tree should be roughly the same size as the standard value-choice search space. As Figure 1 indicates, this space is much richer in good solutions than the other two. These results show that the standard search space is not necessarily superior at all. In fact, a search space based on variable-choice is skewed toward better solutions.

Just because high quality solutions are more frequent

[1] These sizes also fall near the hardness peak for number partitioning [Gent and Walsh, 1996], which specifies $\log_{10} 2^n$ digits for a problem with n numbers.

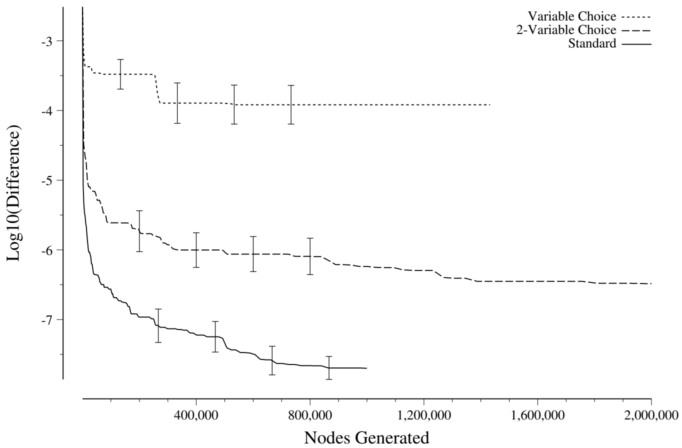

Figure 4: Searching the greedy spaces using DFS.

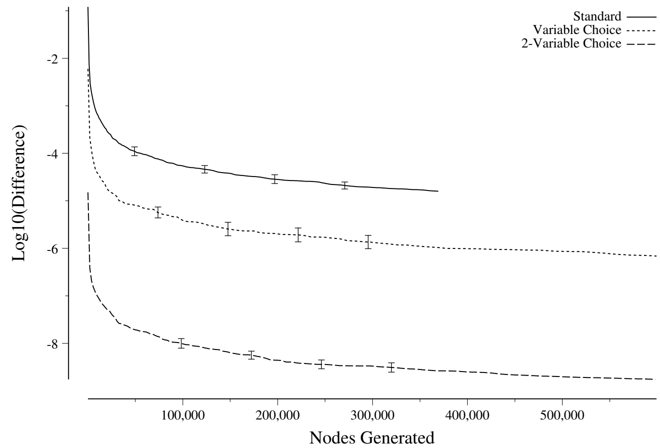

Figure 5: Searching the KK spaces using random sampling.

doesn't necessarily mean that search algorithms will be able to find them any more easily. But Figure 2 shows that the HBSS heuristic-biased sampling algorithm is in fact more effective in the variable-choice spaces. Preliminary results for adaptive probing are shown in Figure 3. The algorithm has trouble learning the large number of parameters necessary to model the full variable-choice space. But when the size of the tree is controlled for, we see that the variable-choice space is clearly superior. Both stochastic tree search algorithms can perform better when allowed to search over the variable choice rather than the value choice.

This did not seem to hold for systematic search algorithms. Figure 4 shows results using depth-first search. (This is the most effective search strategy I know of for the standard space.) The standard space was clearly superior to the variable-choice spaces. Similar results were obtained with ILDS and DDS. This reinforces the idea that assumptions derived from long experience with systematic tree algorithms should be revisited when using stochastic algorithms.

The KK Representation

A more sophisticated representation for number partitioning was suggested by Korf [1995], based on the heuristic of Karmarkar and Karp [1982]. The essential idea is to postpone the assignment of numbers to particular partitions and merely constrain pairs of number to lie in either different bins or the same bin. Constrained numbers are reinserted in the list according to the remaining difference they represent. For instance, the numbers 4 and 5 might be constrained to be in different bins, leaving a remaining value of 1 to be accounted for later in the algorithm. At each node, variable choice corresponds to which numbers are chosen and value choice corresponds to whether they are constrained to be together or apart. The standard variable choice heuristic is to choose the two largest numbers and the standard value choice heuristic is to constrain them to different bins. (To avoid having n^2 children at a node, the largest number is always used and the decision is considered to be the choice of the second number to constrain it with.) This formulation creates a very different set of search spaces from the greedy heuristic.

Nevertheless, we find a similar pattern of results. Figure 5 shows the performance of random sampling (these instances contained 64 numbers). The variable-choice search space is even richer, relative to the standard value-choice, than in the greedy formulation. The KK differencing heuristic is very effective, even when the numbers to be differenced are not necessarily the two largest. Results similar to those on the greedy space were obtained using HBSS and adaptive probing.

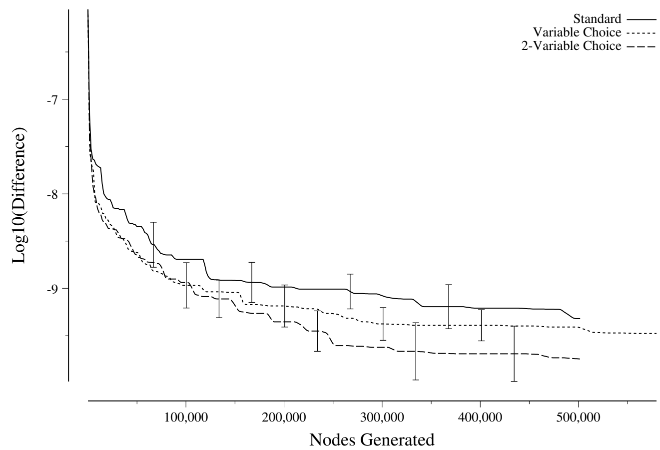

Figure 6: Searching the KK spaces using DDS.

Depth-first search and ILDS performed worse in the variable-choice KK spaces, as one would have expected from the greedy results. However, DDS showed a different pattern. Its performance is presented in Figure 6. Although it is difficult to assess the significance of the differences between adjacent pairs, it seems safe to conclude that DDS performs better in the restricted variable choice search space than in the standard value-choice space. This result suggests that research on appropriate spaces for incomplete stochastic search algorithms may well inform use of systematic methods.

3 Conclusions

I have argued against two prevailing assumptions in research on stochastic search. The first is that stochastic search is suitable only for improvement search and that tree search should be left to complete methods. There is no reason why incomplete stochastic search is inappropriate for trees and I briefly reviewed some algorithms that have been proposed. In fact, a tree-structured search space provides exactly the kind of structure that facilitates representation of information learned during search, a difficult task in an improvement search setting. The second assumption is that stochastic tree search should use the same search space as systematic methods. I presented evidence from two different formulations of number partitioning that suggests that such a conclusion is premature, if not downright incorrect.

Acknowledgments

Thanks to Stuart Shieber and the Harvard AI Research Group for their many helpful suggestions and comments. This work was supported in part by NSF grants CDA-94-01024 and IRI-9618848.

References

[Abramson, 1991] Bruce Abramson. *The Expected-Outcome Model of Two-Player Games*. Pitman, 1991.

[Baluja, 1997] Shumeet Baluja. Genetic algorithms and explicit search statistics. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems 9*, 1997.

[Boese *et al.*, 1994] Kenneth D. Boese, Andrew B. Kahng, and Sudhakar Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16:101–113, 1994.

[Boyan *et al.*, 2000] Justin Boyan, Wray Bruntine, and Arun Jagota. Statistical machine learning for large-scale optimization. *Neural Computing Surveys*, 3:1–58, 2000.

[Bresina, 1996] John L. Bresina. Heuristic-biased stochastic sampling. In *Proceedings of AAAI-96*, pages 271–278. AAAI Press/MIT Press, 1996.

[Crawford and Baker, 1994] James M. Crawford and Andrew B. Baker. Experimental results on the application of satisfiability algorithms to scheduling problems. In *Proceedings of AAAI-94*, pages 1092–1097, 1994.

[Dorigo and Gambardella, 1997] Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[Feo and Resende, 1995] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[Gent and Walsh, 1996] Ian P. Gent and Toby Walsh. Phase transitions and annealed theories: Number partitioning as a case study. In *Proceedings of ECAI-96*, 1996.

[Ginsberg, 1993] Matthew L. Ginsberg. Dynamic backtracking. *Journal of Artificial Intelligence Research*, 1:25–46, 1993.

[Harvey and Ginsberg, 1995] William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. In *Proceedings of IJCAI-95*, pages 607–613. Morgan Kaufmann, 1995.

[Johnson *et al.*, 1991] David S. Johnson, Cecilia R. Aragon, Lyle A. McGeoch, and Catherine Schevon. Optimization by simulated annealing: An experimental evaluation; Part II, graph coloring and number partitioning. *Operations Research*, 39(3):378–406, May-June 1991.

[Joslin and Clements, 1998] David E. Joslin and David P. Clements. "Squeaky wheel" optimization. In *Proceedings of AAAI-98*, pages 340–346. MIT Press, 1998.

[Juillé and Pollack, 1998] Hughes Juillé and Jordan B. Pollack. A sampling-based heuristic for tree search applied to grammar induction. In *Proceedings of AAAI-98*, pages 776–783. MIT Press, 1998.

[Karmarkar and Karp, 1982] Narenda Karmarkar and Richard M. Karp. The differencing method of set partitioning. Technical Report UCB/CSD 82/113, Computer Science Division, University of California, Berkeley, 1982.

[Karmarkar *et al.*, 1986] Narenda Karmarkar, Richard M. Karp, George S. Lueker, and Andrew M. Odlyzko. Probabilistic analysis of optimum partitioning. *Journal of Applied Probability*, 23:626–645, 1986.

[Korf, 1995] Richard E. Korf. From approximate to optimal solutions: A case study of number partitioning. In *Proceedings of IJCAI-95*, 1995.

[Korf, 1996] Richard E. Korf. Improved limited discrepancy search. In *Proceedings of AAAI-96*, pages 286–291. MIT Press, 1996.

[Marchiori and Steenbeek, 1998] Elena Marchiori and Adri Steenbeek. An iterated heuristic algorithm for the set covering problem. In Kurt Mehlhorn, editor, *Proceedings of the Workshop on Algorithm Engineering*, pages 155–166, 1998.

[Ruml, 2001] Wheeler Ruml. Incomplete tree search using adaptive probing. In *Proceedings of IJCAI-01*, 2001. To appear.

[Walsh, 1997] Toby Walsh. Depth-bounded discrepancy search. In *Proceedings of IJCAI-97*, 1997.