An Efficient Approximation to the Barrier Tree Using the Great Deluge Algorithm

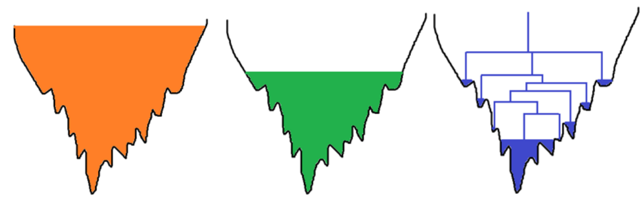
Hansang Yun
Department of Computer Science and Engineering
Seoul National University
Seoul, Korea
hs@soar.snu.ac.kr

Byung-Ro Moon
Department of Computer Science and Engineering
Seoul National University
Seoul, Korea
moon@snu.ac.kr

ABSTRACT

A barrier tree is a model for representing the hierarchical distribution of local optima and valleys. While it is useful, constructing a barrier tree is challenging for a large problem instance. In this paper, we propose an efficient method to approximate the barrier tree. One important subgoal is to estimate a saddle point between two solutions, and it is achieved by exploiting the bias of the Great Deluge Algorithm. We also present a case study of a pseudo-boolean problem of size 296, which is roughly 6 times larger than the scale that the existing methods can handle.

ACM Reference Format:
Hansang Yun and Byung-Ro Moon. 2018. An Efficient Approximation to the Barrier Tree Using the Great Deluge Algorithm. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3205651.3205715



(a) Brute force (b) Branch and bound (c) Suggested method

Figure 1: Performance comparison.

1 INTRODUCTION

We discuss a landscape induced by a fixed unary operator M given as an irreducible, reversible Markov chain. Let S be the finite set of all solutions, and $\pi \in \mathbb{R}^S$ be the stationary distribution of M. The objective function $f : S \to \mathbb{R}$ is to be minimized.

Definition 1.1. A *landscape* of (S, M) is the undirected graph $G = (V, E)$, where $V = S$ and $E = \{(x, y) \in S \times S | M(x, y) > 0\}$.

Definition 1.2. Let $L_h = \{x \in S | f(x) \leq h\}$, and $G[L_h]$ be the subgraph in G induced by L_h. A connected component of $G[L_h]$ is called a *cycle* of a height h. For $x \in S$ and $h \geq f(x)$, let $C_h(x)$ be the unique cycle of the height h that contains x.

Definition 1.3. We define an equivalence relation $\sim_h$ on L_h by setting $x \sim_h y$ if there exists a path in G from $x \in L_h$ to $y \in L_h$ such that all solutions on the path have objective values less than or equal to h.

It can be shown that the inclusion relations between the cycles forms a tree structure [4], which is called a *barrier tree*. The cycles in the barrier tree naturally form a hierarchical clustering of a landscape, and they are known to be highly associated with the asymptotic behaviors of algorithms that rarely accept worsening moves, such as Simulated Annealing (SA) [1].

The known methods of constructing exact barrier trees are quite expensive. The simplest method enumerates all the solutions in the landscape (Figure 1a). The fastest known method uses a branch and bound technique to enumerate the solutions below some bound, and creates a truncated barrier tree [5] (Figure 1b). In both methods, it is required to keep the enumerated solutions in memory, which is infeasible for large problem instances, e.g., a pseudo-boolean problem of size 50 cannot be handled with 4GB RAM.

2 METHOD OVERVIEW

Our method first builds multiple subtrees of a barrier tree and then hierarchically group them into one larger tree (Figure 1c). Each disjoint subtree is obtained by enumerating the solutions in the corresponding cycle. For a single subtree, the method first generates a local minimum x, and collects the solutions in the largest cycle $C_h(x)$ that can be enumerated under the memory constraints.

In order to make groups of the subtrees at the correct height, one must be able to calculate the height of the minimal cycle which includes any given pair of cycles. Note that, for any two disjoint cycles C_1 and C_2, and any two solutions $x \in C_1$ and $y \in C_2$, the minimal cycle including C_1 and C_2 is identical to the minimal cycle including x and y. Therefore, the problem can be solved by obtaining the smallest h such that $x \sim_h y$, i.e., the objective value of a saddle point between x and y is the height that we want to compute.

3 MODIFIED GREAT DELUGE ALGORITHM

We present a variant of the Great Deluge Algorithm (GDA) [3] in which the cooling schedule of the water level h is automatically set (Algorithm 1). By Proposition 3.1, the whole process of it can be regarded as consecutively choosing child nodes in the barrier tree; it starts from the node $C_{h_0}(x)$, and repeats moving to one of the

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3205715

child nodes until it reaches a leaf node. The probability of choosing a child is proportional to its size (with respect to π), as long as t_0 and t are large enough.[1] Hence, GDA is biased toward cycles that are larger than their siblings in the barrier tree.

Algorithm 1 Modified GDA

Input: initial solution x, height $h_0 \geq f(x)$, number of steps t_0, t
Output: local minimum $y \in C_{h_0}(x)$

```
function GDA(x, h_0, t_0, t)
    (y, b) ← WalkBelow(x, h_0, t_0);
    while b = false do
        (y, b) ← WalkBelow(y, f(y), t);
    end while
    return y;
end function
function WalkBelow(x, h, t)
    y ← x; b ← (f(x) = h);
    for i ← 1 to t do
        generate a new solution y' by applying M to y;
        if f(y') < h then
            y ← y'; b ← false;
        else if f(y') = h then
            y ← y';
        end if
    end for
    return (y, b);
end function
```

Proposition 3.1. *Let Y be the output of GDA. For a cycle $C \subseteq C_{h_0}(x)$ which is not a leaf, let $\{C_j\}_{j \in I}$ be the set of child nodes of C.*

$$\lim_{t, t_0 \to \infty} \Pr[Y \in C_j | Y \in C] = \frac{\pi(C_j)}{\sum_{j' \in I} \pi(C_{j'})} \quad \forall j \in I.$$

4 SADDLE POINT APPROXIMATION

If two solutions x and y are in the same cycle which is small enough to be enumerated under the memory constraints, it is trivial to calculate the smallest h such that $x \sim_h y$. For general cases where the condition does not hold, the key idea is to move x and y to a smaller cycle by using the bias of GDA.

Let h be a height satisfying $x \sim_h y$, and suppose we want to verify if the relation is actually true. Let x' and y' be solutions obtained by running GDA(x, h, t_0, t) and GDA(y, h, t_0, t), respectively. Due to the bias, it is likely that both runs have converged to a cycle whose ancestors are larger than their siblings. If $C(\{x', y'\})$ is small enough as a result, one can easily check that $x' \sim_{h'} y'$ for some $h' < h$ by enumeration, implying that $x \sim_h y$ (Figure 2). The success rate of this process, denoted by r, can be expressed as

$$r = \sum_{C' \in \mathbb{M}(C)} \left(\Pr[GDA(x, h, t_0, t) \in C']\right)^2,$$

where $C = C_h(x) = C_h(y)$ and $\mathbb{M}(C)$ is the set of maximal subcycles of C that can be enumerated.[2] One can repeat the process s times to reduce the error rate to $(1-r)^s$. If the landscape has some large cycles dominating siblings like in Figure 2, r is likely to be high.

[1] t can be set to a value smaller than t_0, since annealing is done at previous heights.
[2] If $x \nsim_h y$, on the other hand, r is always zero.

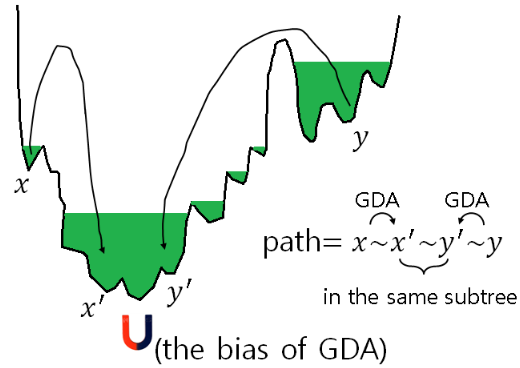


Figure 2: A successful case of verifying $x \sim_h y$. Each shaded region can be obtained under the memory constraints.

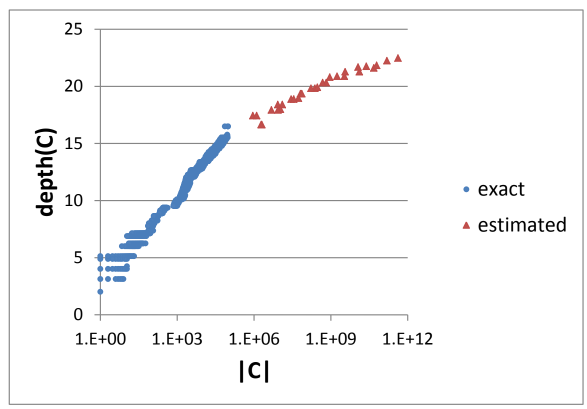


Figure 3: The sizes and the depths of cycles ($|C| = 2^{296}\pi(C)$). The cycles of cardinality less than 10^{12} are plotted.

5 CASE STUDY

As an example, we use a pseudo-boolean problem of size 296 presented in [2] which shows that Quantum Annealing performs significantly better on this kind of problems than SA. The barrier tree of the landscape induced by the 1-bit flip operator is approximated. In Figure 3, we plot the correlation between the sizes and the depths of cycles in the tree; a depth of a cycle is defined as the energy required to escape C from the best solution in C. The high correlation shows how well the *big valley hypothesis* holds in the landscape.

REFERENCES

[1] Olivier Catoni. 1999. Simulated annealing algorithms and Markov chains with rare transitions. In *Séminaire de probabilités XXXIII*. Springer, 69–119.
[2] Vasil S Denchev, Sergio Boixo, Sergei V Isakov, Nan Ding, Ryan Babbush, Vadim Smelyanskiy, John Martinis, and Hartmut Neven. 2016. What is the computational value of finite-range tunneling? *Physical Review X* 6, 3 (2016), 031015.
[3] Gunter Dueck. 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics* 104, 1 (1993), 86–92.
[4] Christoph Flamm, Ivo L Hofacker, Peter F Stadler, and Michael T Wolfinger. 2002. Barrier trees of degenerate landscapes. *Zeitschrift für physikalische chemie* 216, 2 (2002), 155.
[5] Jonathan Hallam and A Prugel-Bennett. 2005. Large barrier trees for studying search. *IEEE Transactions on Evolutionary Computation* 9, 4 (2005), 385–397.