

Program Search with Simulated Annealing

Mitsunori Miki
Department of Knowledge
Engineering and Computer
Sciences
Doshisha University
610-0394 Kyoto, Japan
mmiki@mail.doshisha.ac.jp

Masafumi Hashimoto
Department of Information
Systems Design
Doshisha University
610-0394 Kyoto, Japan
mhashimo@mail
.doshisha.ac.jp

Yoshihisa Fujita
Graduate School, Department
of Knowledge Engineering and
Computer Sciences
Doshisha University
610-0394 Kyoto, Japan
yoshihisa@mikilab
.doshisha.ac.jp

Categories and Subject Descriptors

G.4 [MATHEMATICAL SOFTWARE]: Algorithm design and analysis

General Terms

Algorithms

Keywords

program search, simulated annealing, genetic programming, syntactic introns, bloat, fixed temperatures

1. INTRODUCTION

In many optimization problems, a search method suited to the problem is used, and optimization is performed efficiently using this method. Thus, we believe that the same is true for program search, and by applying different search methods suitable to the problem, optimization can be achieved more efficiently.

In this study, the possibility of the above was examined by conducting a program search using SA, which is a representative meta-heuristic search method, similarly to GA, and comparing it to standard GP. The program search method using SA is named Simulated Annealing Programming (SAP). Temperature schedule has a marked impact on the search in SA, but in previous research, the temperature schedule was not examined. In addition, there were a few studies on program search using SA, and we believe the effectiveness of SA in program search has yet to be demonstrated. In this paper, SAP with the optimized temperature schedule was compared to standard GP to examine the effectiveness of SAP.

2. SIMULATED ANNEALING PROGRAMMING

Simulated Annealing Programming (SAP) is a program search method that expands Simulated Annealing (SA).

SAP is a mutation-based program search method, and the specific generation method first randomly selects the mutation point on the current solution, deletes the subtree that has the selected point as its root node, and then inserts

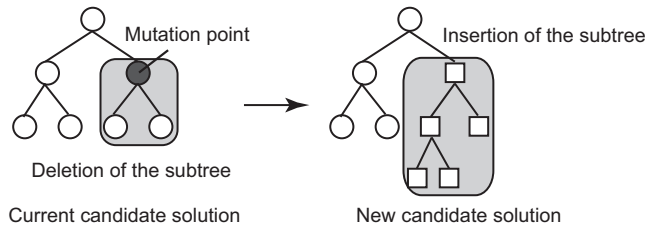


Figure 1: Method for generation of new candidate solution in SAP

a randomly generated subtree (Fig. 1). Its most remarkable feature is having a mechanism that not only allows the transition of improvement direction but also stochastically allows changes in the uphill direction according to a control parameter called temperature. The Metropolis criterion shown in (1) is used as the acceptance criterion. The above feature means that this program search method can be expected to yield optimal solutions even for problems with local minima.

$$P_{AC} = \begin{cases} 1 & \text{if } \Delta E \leq 0 \\ \exp(-\frac{\Delta E}{T}) & \text{otherwise} \end{cases} \quad (1)$$

3. EXPERIMENTS AND RESULTS

A comparative experiment on SAP and standard GP was performed to examine the effectiveness of SAP. The results of a numerical experiment indicated that SAP using a temperature schedule with a fixed temperature did not produce bloating even without the use of a special method for preventing bloat, such as restricting the depth and size of the program, including program size in the evaluation function, and devising operation methods, allowing it to generate more compact programs as compared with the standard GP.

The above results indicate that, and in problems with the possibility of syntactic introns, and with equal complexity to the problems such as Santa Fe trail and Wall-following, the performance of SAP is equal to or better than that of the standard GP. Thus, by selecting search methods appropriate for each problem, the possibility of conducting a search more effectively than with GP has increased.

In future, we will explore the potential of SAP by investigating its effectiveness in problems that require many nodes, and by investigating operation methods - *i.e.*, what types of node should be inserted, or how to select mutation points.