

Fast Hypothetical Reasoning by Parallel Processing

Yutaka Matsuo and Mitsuru Ishizuka

University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan
{matsuo, ishizuka}@miv.t.u-tokyo.ac.jp

Abstract

Cost-based hypothetical reasoning is an important framework for knowledge-based systems because it is theoretically founded and useful for many practical problems. Basically, it tries to find the minimum-cost set of hypotheses that is sufficient for proving a given goal. However, since the inference time of hypothetical reasoning grows exponentially with respect to problem size, slow inference speed often becomes the most crucial problem in practice.

We have developed a new method to efficiently find a near-optimal solution, i.e., a solution whose cost is nearly minimal. This method uses parallel software processors to search for a solution. In order to grasp the search intuitively, we emulate parallel processing on a single processor and develop efficient algorithms. We assume that each variable and each Horn-rule (constraint) is a processor and behaves as follows: a Variable processor sends a message to lower the cost of a solution and a Constraint processor sends a message to satisfy itself.

Our approach is realized mathematically by the augmented Lagrangian method, an efficient parallel computation method. A Variable processor has a (prime) variable which takes the value $[0,1]$. A Constraint processor has also a (dual) variable which represents how strongly the constraint should be considered. The messages and updating procedures are defined by mathematical formulae.

This approach has two major advantages. First, we can generalize related methods such as our earlier SL method¹, the breakout method or Gu's nonlinear optimization method for SAT problems. They are all variants of our approach, obtained by modifying a part of the message to be sent.

Second, we can design new algorithms superior to previous algorithms such as SL method. We currently experiment with seven different algorithms of the parallel processor model, and find two algorithms are prominent as to inference time and solution cost. One algorithm is similar to the breakout method except that it considers the cost of the solution. The other is an entirely new algorithm, which adds one new processor to a problematic Horn-rule if search gets stuck in a local minimum. Using this algorithm, we can obtain solutions whose cost on average is lower than that of any other algorithm compared in our experiment.

¹ Y. Matsuo and M. Ishizuka: SL Method for Computing a Near-optimal Solution using Linear and Non-linear Programming in Cost-based Hypothetical Reasoning, Proc. PRICAL-98, pp.611-625, 1998.