Performance of a Comprehensive and Efficient Constraint Library Based on Local Search*

J. H. M. Lee, H. F. Leung, and H. W. Won

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{jlee, lhf, hwwon}@cse.cuhk.edu.hk

Abstract. Constraint satisfaction is one of the major areas in AI that has important real-life applications. Lee et al. propose E-GENET, a stochastic solver for general constraint solving based on iterative repair. Performance figures show that E-GENET compares favorably against tree-search based solvers in many hard problems. On the other hand, global constraints have been shown to be very effective in modeling complicated CSP's. They have also improved substantially the efficiency of tree-search based solvers in solving real-life problems. In this paper, we present a comprehensive and efficient library of elementary and global constraints for E-GENET. Such a library is essential for applying E-GENET to complex real-life applications. We first present the improved performance of some constraints that appear in our previous papers, followed by the implementation details of three additional global constraints available in the CHIP constraint language. Experimental results, using standard benchmarks and a real-life problem, confirm empirically that the E-GENET architecture is comparable to, if not better than, state of the art in constraint solver technology.

1 Introduction

The research into constraint satisfaction problems (CSP) [14] is one of the major topics in artificial intelligence that have important real-life applications. A CSP can be stated as follows. We are given a finite set $V$ of variables. Each variable $x$ ranges over its domain $D_x$, a finite set of discrete constants. There is a finite set $C$ of constraints over these variables, which limits the allowed combination of values that can be taken by the variables. The goal is to find a consistent assignment of values to the variables so that all constraints are satisfied.

Two main approaches to tackle CSP's are backtracking tree-search, probably enhanced with consistency algorithms [10], and iterative repair methods [15]. An extension of GENET [5], E-GENET [11, 12] is a stochastic solver for general constraint solving based on iterative repair. Performance figures show that E-GENET compares favorably against tree-search based solvers (such as CHIP [7])

* This project is supported in part by a CUHK Direct Grant.
in many hard problems. E-GENET encapsulates both an architecture and an algorithm. The architecture encodes a compact representation scheme for binary and non-binary constraints, whereas the algorithm governs how solution is searched using an iterative repair procedure coupled with learning for escaping from local minima. The Breakout Method of Morris [16] emphasizes more on the algorithmic aspect of the method, without explaining how constraints are represented. Walser [25] and Walser et al. [26] restrict attention to linear pseudo-Boolean and integer constraints in the WSAT algorithm. Thornton and Sattar [23] applies weighted iterative repair to the solution of real-life over-constrained nurse rostering but no details of the solver architecture is given.

In order for E-GENET to be applicable to real-life applications, we design and implement a comprehensive library of constraints for E-GENET. The library consists of elementary and global constraints [3], the latter of which have been shown to be effective in modeling and solving a large class of difficult sequence, scheduling, geometrical placement and vehicle routing problems [19, 18, 20]. Algorithmic implementation of global constraints is nontrivial, requiring sophisticated and specific algorithms. We demonstrate how simplistic representation of these constraints in E-GENET can result in performance superior to state of the art algorithmic implementation.

While the design details of some constraints in the library were described previously [11, 12], we give improved performance figures of these constraints, resulting from fine tuning of our implementation. We further describe how three global constraints, namely *among*, *diffn*, and *cycle*, can be effectively implemented in E-GENET. Benchmarking results, including the Mystery Shopper Problem, confirm the representational and algorithmic efficiency of the E-GENET architecture and algorithm respectively.

The rest of the paper is organized as follows. We briefly review E-GENET in section 2. Performance results of the library and design of the new global constraints are reported in section 3. Section 4 summarizes and remarks on the results of the paper.

2 An Overview of E-GENET

An extension of GENET [5], E-GENET [11, 12] is a network architecture that solves general binary and non-binary constraint satisfaction problems [14] (CSP's) using local search. There are two types of nodes in E-GENET. Each *variable node* represents a variable in a CSP and contains the domain of the variable. Each *constraint node* represents a constraint in the CSP. A variable node \( x \) is connected to a constraint node \( c \) if \( x \) occurs in \( c \).\(^1\) For a constraint \( c(x_1, \ldots, x_n) \), each possible combination \( (v_1, \ldots, v_n) \) of values from domains of \( (x_1, \ldots, x_n) \) is given an *initial penalty value* \( \delta_c(v_1, \ldots, v_n) \), which is 0 if \( c(v_1, \ldots, v_n) \) is true, or -1 otherwise.

\(^1\) We abuse terminology by naming a variable node (constraint node) by the corresponding variable (constraint).