Interactive Distributed Configuration

Peter Tiedemann, Tarik Hadzic,
Thomas Stuart Henney, and Henrik Reif Andersen

Computational Logic and Algorithms Group, IT University of Copenhagen,
Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark
{petert, tarik, tsh, hra}@itu.dk

Abstract. Interactive configuration is the process of assisting a user in selecting values for parameters that respect given constraints. Inspired by the increasing demand for the real-time configuration in Supply Chain Management, we apply a compilation approach to the problem of interactive distributed configuration where the user options depend on constraints fragmented over a number of different locations connected through a network. We formalize the problem, suggest a solution approach based on an asynchronous compilation scheme, and perform experimental verification.

1 Introduction

An interactive configuration problem is an application of Constraint Satisfaction Problems (CSP) where a user is assisted in interactively assigning values to variables by a software tool. This configurator assists the user by displaying the valid choices for each unassigned variable in what are called valid domains computations. In many application domains user options depend on constraints fragmented over a number of different locations connected through a network. This paper attempts to extend the functionality provided by centralized configuration to distributed configuration while taking advantage of the distribution of the problem when possible. One of the current solution approaches to interactive configuration is to divide the computational effort into an offline and online phase. First compile the set of all valid solutions offline into a compact symbolic representation based on Binary Decision Diagrams (BDD). In the on-line phase we can deliver valid domains computation in time bounded polynomially in the size of the BDD representation [123].

2 Interactive Distributed Configuration

One application domain where constraint information is fragmented is in Supply Chain Management [4], where the user options depend on many interrelated businesses. The dominating relationship in a supply chain is the supplier producing items offered to consumers. We model these relationships with a supply chain graph (SCG), which is a directed graph $G(\mathcal{N}, A)$ where every node $N_i \in \mathcal{N}$
denotes a business entity and there is an arc \((N_i, N_j) \in A\) iff a company \(N_i\) supplies goods to \(N_j\). We will assume that an SCG is a directed acyclic graph. Note that this will not imply in any way an acyclic constraint graph. In the SCM domain each company have independent variable namespaces in their models. For ease of exposition of the techniques discussed in this paper we will in the rest of the paper assume a global namespace between the distributed models, for details see [5]. We now introduce some basic terminology and define the distributed configuration model. An assignment \(\rho\) is a set of variable-value pairs \((x, v)\), such that any one variable occurs only once. \(\text{dom}(\rho)\) is the set of variables assigned by \(\rho\). A total assignment is an assignment \(\rho\) such that \(\text{dom}(\rho) = X\). An assignment \(\rho\) is valid iff there exists a total assignment \(\rho'\) such that \(\rho \subseteq \rho'\) and \(\rho'\) satisfies all constraints.

**Definition 1 (Distributed Configuration Model).** A distributed configuration model \(\text{DisC}(X, D, F, N)\) consists of a set of variables \(X\), variable domains \(D\), nodes \(N = \{N_1, \ldots, N_k\}\) and a set of sets of propositional formulas over subsets of \(X\), \(F = \{F_1, \ldots, F_k\}\). Each node \(N_i\) is associated a constraint \(C_i = \bigwedge_{f \in F_i} f\). The scope of this constraint is denoted \(X_i\), and the restriction of the domains to \(X_i\) is denoted \(D_i\). A solution to the distributed configuration model is a total assignment to \(X\) satisfying \(C_i\) for all nodes \(N_i \in N\).

We will interchangeably refer to constraints such as \(C_i\) as a constraint and as the set of solutions to this constraint over all variables \(X\).

### 2.1 Overall Solution Approach

To solve the problem we take advantage of the fact that the user in any reasonable configuration problem over a supply chain is only interested in choosing values for a small subset of the variables, denoted \(X_u\). We designate a node as the user node \(N_u\), storing a constraint over variables \(X_u\). The user performs interactive configuration by interacting with this node only. Let \(\text{Sol}\) be the set of solutions to the global problem, \(\text{Sol} = \bigcap_{i=1}^k C_i\). At \(N_u\) we need to generate the minimal solution space over the user variables, that is \(\text{minSol}_u = \pi_{X_u}(\text{Sol})\). Hence all solutions in \(\text{minSol}_u\) are globally consistent. After generating \(\text{minSol}_u\) at \(N_u\), the standard interactive configuration approach can be applied, resulting in a partial assignment \(\rho \in \text{minSol}_u\). Since \(\text{minSol}_u\) is globally consistent we can extend \(\rho\) to \(\rho' \in \text{Sol}\). Discovering this extension is called \text{validation} and can be performed efficiently. Due to space constraints we do not describe the validation algorithm here.

### 3 Distributed Compilation Algorithms

The most complex part of our solution approach is to generate a minimal solution space on a user node. The BDD representing \(\text{minSol}_u\) can be calculated as: \(\exists (X \setminus X_u). (C_1 \land C_2 \land C_3 \ldots \land C_k)\). Our algorithms will process a schedule placing and ordering the above conjunctions. We define a distributed conjunction schedule.