1. Introduction

“Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.”
Eugene C. Freuder, Constraints, April 1997

Overview

Constraint programming is an emergent software technology for declarative description and effective solving of large, particularly combinatorial, problems especially in areas of planning and scheduling. It has recently emerged as a research area that combines researchers from a number of fields, including Artificial Intelligence, Programming Languages, Symbolic Computing and Computational Logic. Constraint networks and constraint satisfaction problems have been studied in Artificial Intelligence starting from the seventies. Systematic use of constraints in programming has started in the eighties. In constraint programming the programming process consists of the generation of requirements (constraints) and solution of these requirements, by specialized constraint solvers.

Constraint programming has been successfully applied in numerous domains. Recent applications include computer graphics (to express geometric coherence in the case of scene analysis), natural language processing (construction of efficient parsers), database systems (to ensure and/or restore consistency of the data), operations research problems (like optimization problems), molecular biology (DNA sequencing), business applications (option trading), electrical engineering (to locate faults) and circuit design (to compute layouts).

This book is centered around the notion of constraint solving [189] and programming [141]. The interest in constraint satisfaction problems can be easily justified, since they represent a very powerful, general, and declarative knowledge representation formalism. In fact, many real-life situations can be faithfully described by a set of objects, together with some constraints among them. Examples of such situations can be found in several areas, like VLSI, graphics, typesetting, scheduling, planning, as well as CAD, decision-support systems and robotics.

1.1 From the Beginning . . .

The constraint idea comes from the early 1960’s when Sutherland introduced Sketchpad [187], the first interactive graphical interface that solved geometric
constraints. After that came Fikes’ REF-ARF [97] and at the beginning of 1970’s Montanari described fundamental properties of the constraints when applied to picture processing [149]. Another study in finite domain constraint satisfaction was done at the end of 1970’s in Laurièere’s ALICE [133], a system developed to solve prediction/detection problems in geology. After these, several constraint languages have been proposed in the literature: the language of Steele ([184]), CONSTRAINTS [186] of Sussman & Steele, Thinglab ([60]) and Bertrand ([134]).

From Sketchpad until now, a lot of research has been done and improvements made, and the classical constraint satisfaction problems (CSPs) [139,150] have been shown to be a very expressive and natural formalism to specify many kinds of real-life problems.

1.2 Applications

Today, the use of the constraint programming idea to solve many real-life problem is reality. Many important companies develop tools based on the constraint technology to solve assignment, network management, scheduling, transport and many other problems:

1.2.1 Assignment Problems

Assignment problems were one of the first type of industrial applications that were solved with the CLP technology. These problems usually have to handle two types of resources, and constraints among them, and try to assign one resource of the first kind to one of the second kind such that all constraints are satisfied.

An example is the stand allocation for airports, where aircrafts (the first kind of resources) must be parked on the available stands (the second kind of resources) during their stay at the airport. The first industrial CLP application was developed for the HIT container harbor in Hong Kong [161], using the language CHIP: the objective was to allocate berths to container ships in the harbor, in such a way that resources and stacking space for the containers is available. Other Hong Kong applications are at the airport, where a CHIP-based system is used to solve the counter allocation problem [69], and another constraint-based system, which uses the ILOG libraries, is used for the stand allocation problem since mid-1998 [70]. Another system, called APACHE [88], was a demonstrator for stand allocation at Roissy airport in Paris: the objective was to replan the allocation when a change of arrival/departure times occurred.

1.2.2 Personnel Assignment

Personnel assignment problems are a special case of assignment problems where one resource type consists of humans. This peculiarity makes them specific enough to be considered separately. In fact, changing work rules and regulations impose difficult constraints which evolve over time. Also, user preferences