International Journal of Computer Applications Technology and Research
Volume 3– Issue 7, 395 - 399, 2014, ISSN: 2319–8656

A Review of Constraint Programming

Poonam Dabas
Department of CSE
U.I.E.T, Kurukshetra University
Kurukshetra, India

Vaishali Cooner
Department of CSE
U.I.E.T, Kurukshetra University
Kurukshetra, India

Abstract: A constraint is defined as a logical relation among several unknown quantities or variables, each taking a value in a given domain. Constraint Programming (CP) is an emergent field in operations research. Constraint programming is based on feasibility which means finding a feasible solution rather than optimization which means finding an optimal solution and focuses on the constraints and variables domain rather than the objective functions. While defining a set of constraints, this may seem a simple way to model a real-world problem but finding a good model that works well with a chosen solver is not that easy. A model could be very hard to solve if it is poorly chosen.

Keywords: Constraint Programming; Optimization; feasibility; problems; relations

1. INTRODUCTION

The development of high-tech systems is very difficult without mathematical modeling and analysis of the system behavior. For this, mathematical models are revealed in order to solve the tasks in many areas like in the modern engineering sciences like control engineering, communications engineering, and robotics. Therefore, the main focus is that without neglecting mathematical accuracy on comprehensibility and real-world applicability. Mathematical engineering has various methods to find the optimal and feasible solution like: Linear programming, Non-Linear programming, stochastic programming and Constraint programming.

Linear programming is effective only if the real world is reflected in the model used. They also sometimes give results that don't make sense in the real world. Even some situations have many possibilities to fit into linear programming. A constraint is a logical relation among several unknown quantities (or variables), each taking a value in a given domain.

2. CONSTRAINT PROGRAMMING

A logical relation among several unknown variables is known as a constraint, where each variable takes a value in a given domain. The basic idea behind constraint programming framework is to model the problem as a set of variables with domains and a set of constraints [16]. The possible values that the variables can take are restricted by the constraints.

In operations research constraint programming (CP) is an emergent field. It is based on finding a feasible solution i.e. feasibility rather than finding an optimal solution i.e. optimization. Basic CP constructs, the interface for advanced scheduling applications, and search specification are provided which are essential to a language supporting constraint programming and are represented as discrete variables [1].

The focus is not done on objective function rather than the constraints and variables domain. It possesses a strong theoretical foundation though it is quite new, a widespread and very active community around the world and an arsenal of different solving techniques. In problems with heterogeneous constraints CP has been successfully applied in planning and scheduling.

A programming paradigm where relations between variables are stated in the form of constraints is known as constraint programming. In other programming languages step or sequence of steps is not specified to execute. Because of this constraint programming a known as a form of declarative programming.

Various kinds of constraints are used in constraint programming: one is those used in constraint satisfaction problems for example- A or B is true, other one is those solved by the simplex algorithm for example- $x \leq 5$, and others.

To solve scheduling problems constraint programming is an interesting approach. Activities are defined by their starting date in cumulative scheduling; their duration and the amount of resource necessary are also defined for their execution.

Constraints are defined as just relations and which relation should hold among the given decision variables is stated by a constraint satisfaction problem (CSP). It may seem a simple while defining a set of constraints as a way to model a real-world problem but it is not easy to find a model that works well with a chosen solver. It is really hard to solve a poorly designed model. To take advantage of the features of the model such as symmetry solvers can be designed to save time in finding a solution. As many are over constrained this may exist as another problem with modeling real-world problems. Any language can be used to implement constraint solver.

For all the constraints to be satisfied there must exist an assignment of values to variables. To reduce the computational effort this technique is used which is needed to

solve combinatorial problems. Constraints are used in a constructive mode to deduce new constraints, not only to test the validity of a solution. Constraints also detect inconsistencies rapidly.

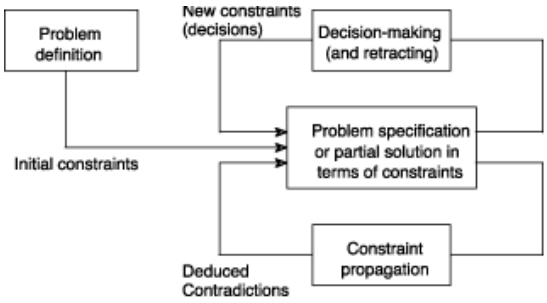

Figure. 1 Behavior of a Constraint Programming system

Constraint propagation is usually incomplete for complexity reasons. So, not all but some of the consequences of constraints are deduced. All inconsistencies cannot be detected by constraint propagation.

To determine if the CSP instance is consistent or not tree search algorithms must be implemented. The figure depicts the overall behavior of a constraint-based system.

First, variables and constraints are defined as terms of the problem

Then, constraint propagation algorithms are specified. Some pre-defined constraints can be used by the constraint programming tool like scheduling constraints for which the corresponding propagation algorithms have been pre-implemented.

Finally, at last the decision-making processes. It is the way the search tree is built, and is specified. How new constraints are added to the system are shown in it like ordering a pair of activities.

3. REVIEW ANALYSIS

Over the past few years, there has been lot of research going on in the field of mathematical engineering to find the optimal solutions for the problems. Researchers have done a lot in this field which is discussed below:

Willem-Jan van Hoeve[1] has presented the modeling language for basic constraint programming and advanced scheduling constructs and specify how search can be controlled. It provides easy development of hybrid approaches such as CP based column generation. Focus here is done on the constraint programming interface of AIMMS which is based on an algebraic syntax and offers access to integer linear programming, quadratic programming (QP) and nonlinear programming (NLP).

Arnaud Lallouet, M. lopez, L. Martin, C. Vrain [2] have made an algorithm which is designed combining the major qualities of traditional top-down search and bottom-up search techniques. The contributions of this paper are setting the framework of learning CSP specifications, then the choice of the rule language, and it's rewriting into CSP and the learning algorithm which allows guiding search when traditional method fails. In this the activity of finding the constraints that are to be stated is considered as a crucial part and a lot of work has been spent on the understanding and automation of modeling tasks for the novice users who have a limited knowledge regarding how to choose the variables. A framework is designed to bridge the gap between constraint programming modeling language and ILP (Inductive Logic Programming). The very first step of the framework consists in learning a CPS (Constraint Problem Specification) describing the target problem. ILP framework and its applications to learning problems are presented.

Barry O'Sullivan [3] has presented technical challenges in the area of constraint model acquisition, formulation and reformulation algorithms for global constraints and automated solving and it also presents the metrics by which success and progress can be measured. The motivation here is to reduce the burden on constraint programmers and to increase the scope of problems that can be handled alone by domain experts. Modeling defines the problem, in terms of variables that can take different values. Progress is evaluated empirically in constraint programming. A model for practical problem as a constraint satisfaction problem (CSP) is preferred and available constraint programming tools are used to solve it. Generic methods from the machine learning field can be applied to learn an appropriate formulation of the target problem as a CSP. The filtering algorithm is difficult to design and this is considered the major challenge that one faces when designing a new global constraint.

Christian Bessiere, R. Coletta, T. petit [4] have presented a framework for learning implied global constraints which is presented in a constraint network assumed to be provided by a non-expert user. As global constraints are key feature of constraint programming learning global constraints is important. A motivation example is considered and it is shown that if it is required that the model is to be solved with more tasks then the need to improve model is needed. Constraint network is defined by a set of variables and a set of domains of values for the variables. The tighter the learned constraint is, the more promising its filtering power is. A general process to learn the parameters of implied global constraints is given. The focus is made on global constraints and set of parameters. Efficient algorithm exists to propagate when the cardinalities of the value are parameters that take values in a range. A model was generated to minimize the sum of preference variables. This was considered the first approach that derives implied global constraints according to the actual domains. Experiments show that a very small effort

spent learning implied constraints with this technique can improve the solving time.

Steven J. Miller [5] has described linear programming as an important generalization of linear algebra. Various real world situations are modeled successfully using programming. The problems that can be solved by linear programming are discussed. Binary integer linear programming is also discussed which is an example of a more general problem is called Integer Linear Programming. The difficulty here due to the fact that a problem may have optimal real solutions and optimal integer solutions but both the solutions need not be closed to each other. The simplex method is used for solving the linear problems to find the optimal solutions. It has two phases, one is to find a basic feasible solution and other one is to find a basic optimal solution, given a basic feasible solution. If no optimal solution exists this phase produces a sequence of solutions that are feasible with their cost tending to minus infinity. Algorithms are defined for them. The time for finding the optimal solution is also considered as a major factor here.

Nicholas Nethercte, P J. Stukcey, R. Becket, S. Brand, G J. Duck and Guido Tack [6] have presented MiniZinc as a simple and expressive CP modeling language. It is known that there is no standard modeling language for constraint programming problems so most solvers have their own language for modeling. The experimentation and comparison between different solvers is encouraged with a standard language for modeling CP. This MiniZinc problem has two parts- model and data which may be in separate files. The assignments to parameters declared in the model are contained in the data file. The model file is not attached to any particular data file. Boolean, integers, and floats are the three scalar types provided and sets and arrays are two compound types provided. The MiniZinc is translated to FlatZinc in two parts as flattening and the rest. Flattening is done in a number of steps to reduce the model and data as much as possible. The order of the steps is not fixed. After flattening, post flattening steps are applied. Different MiniZinc to FlatZinc converters are used. The main goal here was to define a language which is not too big but expressive.

Alan M. Frisch, M. Grum, C. Jefferson, B.M. Hernandez, Ian Miguel [7] have discussed a new formal language ESSENCE for specifying combinatorial problems which provides a high level of abstraction. This language was a result of attempt to design a formal language that enables abstract problem. For this language no expertise in CP should be needed, it is accessible to anyone with knowledge of discrete mathematics as it is based on the notation and concepts of discrete mathematics. It provides high level of abstraction stating that the language should not force a specification to provide unnecessary information. This language provides an exceptionally rich set of constructs for expressing quantification. It also supports complex, nested types and also its result can be specified without modeling them.

Adrian Petcu [8] has discussed in brief about efficient optimization techniques that are essential to coordinate to business companies and distributed solution processes are desirable as they allow the participating actors to keep control on their data and also offer privacy.

Many key issues are presented that are present in this domain like the actors involved in the distributed decision processes do not have the global knowledge and overview. The goal of constraint optimization is to find the best assignment of values to the variables so that utilities are maximized and cost is minimized. A new technique based on dynamic programming was developed for distributed optimization which was a utility propagation mechanism and works on constraint problems.

It requires only a linear number of messages for finding the optimal solution. These algorithms for distributed constraint optimization have not been applied to large scale due to complexity reason.

Brahim Hnich, S.D. Prestwich, E. Selensky, B.M. Smith[9] have developed models for constraint programming for finding an optimal covering array. It is shown that the compound variables that represent tuples of variables in the original model, allow the constraints of the problem to be represented more easily, propagating better. The optimality of existing bounds is proved for finding the optimal solutions for moderate size array. In covering test problems instances are used with coverage strengths. Number of parameters here is varied. It has shown that for moderate problem size one can find provably optima solution using CP approach. One of the advantages of CP is easy handling of side constraints i.e. simply by adding them to the model.

C. Bessiere, J. Quinqueton, G. Raymond [10] have proposed an automated model to generate different viewpoints for the problem we are to model. The main idea here is to build a viewpoint enough to describe many different solutions of problems also describes a solution of the target problem. Historical data is with which it is started and historical data is used as solutions to problems close to the target problems. From this data candidate variables are extracted. So this can be seen that these viewpoints are capable of describing the historical solutions and also the solutions of our target problem. The goal here is to build viewpoints which match the given historical data. For this candidate variables are determined according to the history. A set of potential viewpoints are obtained out of which more relevant is selected to build constraint models efficiently.

P.E. Hladik. A.M. Deplanche, N. Jussien, H. Cambazard [11] has presented an approach to solve hard real time allocation problem i.e. to assign periodic tasks to processors in context of fixed priority preemptive scheduling. Bendors decomposition is also used as a way of learning when the allocation yields a valid solution. The problem is distributed in systems that belongs to a class. The authors

presents a decomposition based method which separates the allocation problem from the scheduling one. The three classes that the constraint allocation problem must respect are timing, resource, and allocation constraints. For solving a master problem using constraint programming, the problem needs to be translated into CSP. The subproblem is considered as to check whether a valid solution produced by master problem is schedulable or not. If no data is sent then deadlines can correspond to non-communicating tasks. The overall problem is split into a master problem for allocation and resource constraints and a subproblem for timing constraints. The learning technique is used in an effort to combine the various issues into a solution that satisfies all constraints.

Julia L.Higle [12] has presented an introduction to stochastic programming models. Stochastic linear programming is resulted when some of the data elements in a linear program are appropriately described using some random variables. An example is illustrated giving the reason why SP model is preferred and some essential features of a stochastic program are identified. Stochastic programs are difficult to solve and formulate. When the size of the problem increases we can easily see that the solution difficulties increase as well. Sensitivity analysis is done which provides a sense of security and is important. It is used to study the robustness of the solution to a linear programming model. It is done for the accuracy of the data to check whether the solution changes or not on changing the data. If the solution remains same it is believed that the solution is appropriate and vice versa. All the uncertainties should be included in the model.

Philippe Refalo [13] has presented a new general purpose strategy for constraint programming which is inspired from integer programming technique. The importance of a variable for the reduction of the search space is measured by the impact. Designing the search strategy is difficult in integer programming whereas the concept of domain reduction is easier to understand and the use design of a search strategy is easier in constraint programming. In the impact based search strategy, by storing the observed importance of variables impacts permit us to benefit from the search effort made up to a certain node. With some standard strategies some instances remain unsolved which are solved by this technique. Certain principles are defined here for reducing the search effort. When a value is assigned to a variable in constraint programming, constraint propagation reduces the domains of other variables defined.

Y.C Law, J.H.M. Lee [14] has introduced model induction which is a systematic transformation of constraints in an existing model to constraints in another viewpoint. Three ways of combining redundant models are proposed using model induction, another way is model channeling, and the last is model intersection. It is also investigated how the problem formulation and reformulation affect execution efficiency of constraint solving algorithms. For the formulation process the variables and the domain of the variables is to be determined. The induced model is result of the model induction. The three ways of combining the redundant models are proposed so as to utilize the redundant information in enhancing constraint propagation. Alternate ways of generating models in a different viewpoint from existing model are made.

H.Y. Benson, D.F. Shanno, R.J. Vanderbei [15] have analyzed the performance of several optimization codes on large-scale nonlinear optimization problems. The size of problem is defined to the number of variables and the number off constraints. Some of the codes are tested and presented available for solving large scale NLP's. To identify the features of these codes that are efficient is the goal. Infeasibilities and unboundedness are detected in the problem as early as possible. Performance of the algorithms running on the same set of problems is compared to simple compute an estimate of the probability that an algorithm performs. A number of conclusions concerning specific algorithm details exits if various algorithms are compared. Numerical result for solving large scale nonlinear optimization problems is presented. The performance of each solver is explained easily and predicted based on the characteristics.

4. CONCLUSION

There are many challenges faced by mathematical engineering approaches to find the optimal solution. The common weakness to all of the approaches is the assumption that the input data are perfectly accurate. Many benefits of using this approach are discussed as visualization of results using activities and resources. From an existing model another model of a different viewpoint can be generated in a systematic way. Experiments show that we can improve the solving time by very small effort is spent in learning. But at the same time it is too expensive. There are many challenges as: these rely heavily on supervision of an expert and also they are not capable of acquiring a description of the problem class.

5. REFERENCES

[1] Willem-Jan van Hoeve, "Developing Constraint Programming Applications with AIMMS," in CP,2013.

[2] Arnaud Lallouet, Matthieu Lopez, Lionel Martin, Christel Vrain, "On Learning Constraint Problems," in ICTAI, 2010.

[3] Barry O'Sullivan, "Automated Modelling and Solving in Constraint Programming," in proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence(AAAI-10), 2010.

[4] C. Bessi`ere, R. Coletta, and T. Petit, "Learning implied global constraints," in IJCAI, 2007, pp. 44–49.

[5] Steven J. Miller," An Introduction to Linear Programming," in mathematics,2007.

[6] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, and G. Tack, "Minizinc: Towards a standard cp modelling language," in CP, 2007, pp. 529–543.

[7] A. M. Frisch, M. Grum, C. Jefferson, B. M. Hern´andez, and I. Miguel,"The design of essence: A constraint language for specifying combinatorial problems," in IJCAI, M. M. Veloso, Ed., 2007, pp. 80–87.

[8] Adrian Petcu," Recent Advances in Dynamic, Distributed Constraint Optimization," in infoscience, 2006.

[9] Brahim Hnich, Steven D. Prestwich, Evgeny Selensky, Barbara M. Smith, "Constraint Models for the Covering Test Problem," in CP, 2006, pp. 199-219.

[10] C. Bessiere, J. Quinqueton, and G. Raymond, "Mining historical data to build constraint viewpoints," in Proceedings CP'06 Workshop on Modelling and Reformulation, 2006, pp. 1–16.

[11] Pierre-Emmanuel Hladik, Hadrien Cambazard, Anne-Marie Deplanche, Narendra Jussien," in ECRTS, 2005.

[12] Julia L. Higle," Stochastic Programming: Optimization When Uncertainty Matters," in operations research informs-New Orleans 2005, 2005.

[13] Philippe Refalo," Impact-Based Search Strategies for Constraint Programming," in peasant IBS, 2004.

[14] Y.C. Law, J.H.M. Lee," Model Induction: a New Source of CSP Model Redundancy," in AAAI, 2002.

[15] Hande Y. Benson, David F. Shanno, Robert J. Vanderbei," A Comparative Study of Large-Scale Nonlinear Optimization Algorithms," in NLP, 2002.

[16] Roman Bartak," Constraint-Based Scheduling: An Introduction for Newcomers," in SOFSEM, 2002.

[17] R. Bart´ak. Constraint programming: In pursuit of the holygrail. In *Proc. of WDS99*, 1999.

[18] J. Charnley, S. Colton, and I. Miguel, "Automatic generation of implied constraints," in ECAI, 2006, pp. 73–77.

[19] J.-F. Puget, "Constraint programming next challenge : Simplicity of use," in International Conference on Constraint Programming, ser. LNCS, M. Wallace, Ed., vol. 3258. Toronto, CA: Springer, 2004, pp. 5–8, invited paper.

[20] A. M. Frisch, M. Grum, C. Jefferson, B. M. Hern´andez, and I. Miguel, "The design of essence: A constraint language for specifying combinatorial problems," in IJCAI, M. M. Veloso, Ed., 2007, pp. 80–87.