A CONSTRAINT PROGRAMMING APPROACH TO SOLVE RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM

A CONSTRAINT PROGRAMMING APPROACH TO SOLVE RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM

Jose Dula

University of Mississippi, Hearin Center for Enterprise Science

Karen Lewis

University of Mississippi, Hearin Center for Enterprise Science

Keith Womer

University of Mississippi, Hearin Center for Enterprise Science

Haitao Li[1]

University of Mississippi,

Hearin Center for Enterprise Science

This research is motivated by the opportunity to save personnel cost due to technological advancement. The idea is that automation permits the elimination of some tasks but the tasks remaining do not map nicely into the skill sets of existing people. Often the decision to invest in new technology involves large-scale shifts in the mix of skills required to perform both routine and extraordinary work. If the investment in new technology is to be justified then staffing must be reduced. One way to look at this problem is as a resource constrained project scheduling problem (RCPSP).

Resource Constrained Project Scheduling Problem

[1] Contact author: Haitao Li, P.O.Box 7524, University, MS 38677

To give this problem more definition consider the reduction in crew size associated with the new DDX class of Navy ships. This new design will require that many of the routine and non-routine tasks that are now performed by our forces at sea will need to be accomplished with fewer resources. Determining the correct crew size and the right mix of skills for the crew will require that we understand how to schedule the myriad of tasks that a ship's crew must accomplish every day. This means that we not only have to accomplish routine tasks but also the unique tasks associated with combat operations, damage control, replenishment at sea, and other special assignments. The collection of tasks for any day can be thought of as a project.

Project management has a long and colorful history with the Navy. The well known Program Evaluation and Review Technique (PERT) was developed to aid the nuclear submarine development program in the 1950's. There the optimal sequence of interrelated tasks was determined in order to finish a project or a group of projects by a certain time and within a cost constraint. In the environment where resources can be acquired at will the optimization involved with project management is now routine. But RCPSP in this context presents a formidable scheduling problem. The list of tasks to be accomplished on board ship during a day may be small in comparison to a major building project but the task list varies from day to day, and in emergency situations survival may depend on scheduling tasks in the correct order. Further more the job must be accomplished with a fixed crew.

The resource constrained project scheduling problem can be described as follows. A project consisting of tasks must be carried out to achieve a predefined goal, subject to temporal requirements and resource constraints. For example, let

$$S_{jk} - S_{j'k'} \geq \delta^{jk}_{j'k'}$$
$$j, j' \in J; k, k' \in K$$

Where S_{jk} is the starting time of task k one of K tasks in job j, one of J jobs. $\delta^{jk}_{j'k'}$ is the minimum delay between the stating time of task jk and task $j'k'$. Each is defined to be short enough to be carried out without interruption.

If the objective is to find an optimal schedule to minimize the make span then the general formulation of the problem is (Neumann & Zimmermann, 2001):

$$Min S_{n+1}$$
$$s.t. r_k(S,t) \leq R_k \,(k \in R, 0 \leq t \leq \bar{d})$$
$$S_{jk} - S_{j'k'} \geq \delta^{jk}_{j'k'} \,(< jk, j'k' > \in E)$$
$$S_{jk} \geq 0 (j \in J, k \in K)$$
$$S_0 = 0$$

Where $r_k(S, t)$ refers to amount of resource k used at t, R_k is amount of resource k available at any point in time, $\bar{d}$ is an upper bound on the project duration, S_0 and S_{n+1} are the starting times of dummy tasks, and S is a schedule set of each task. R is set of renewable resources is required for carrying out the tasks of the project.

This is a difficult problem because even finding a feasible solution for a given instance is NP-hard. It is a versatile problem because it has many extensions (Sprecher 1994). Problems with generalized temporal constraints, resource requirements varying with time, renewable and non-renewable resource constraints, and alternative objectives such as minimization of weighted finish times, weighted tardiness, weighted flow time and maximization of net present value are all special cases of the problem (Hartmann 1999). Also it is an interesting problem since it can be applied in construction, manufacturing, transportation areas and it can be reduced to many well-know scheduling problems, flow shop, job shop and open shop problems as special cases.

Our plan is to investigate constraint programming as an efficient solution technique for this important problem.

Constraint Programming (CP)

Constraint Programming is the study of computational systems based on constraints. The earliest ideas leading to CP may be found in the Artificial Intelligence area of the Constraint Satisfaction Problem (CSP), dating back to the 1960's. It is important to note that there is a difference between the word "programming" in mathematical programming and constraint programming (Lustig, I.J., and J.-F. Puget, 2001). In CP, the word "programming" refers its roots in the field of a computer programming language. Other programming paradigms are procedure programming, objective oriented programming, functional programming and logic programming. In mathematical programming, the word "programming" has roots in Dantzig 1963, which is associated with a specific mathematical problem. "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." (E. Freuder 1997).

The first modern constraint programming languages were extensions of logic programming languages, called constraint logic programming language (CLP). Three different groups independently developed CLP during late 1980's and early 1990's. In Melbourne, Joxan Jaffar et al, developed CLP[R]; In Marseilles, Alain Colmerauer et al developed Prolog-III and in Munich, at European Computer Industry Research Center, CHIP is developed. Another step toward CP came from the area of concurrent logic programming. One popular example is the Oz system developed by Smolka et al. Now there is the C++ and Java based object oriented constraint solver developed by ILOG, which integrates the constraint programming technique with the OPL programming language.

The main solving technology for constraint programming is constraint satisfaction, which deals with problems defined over finite domains. Probably more than 95% of all industrial constraint application use finite domains. There are three categories of techniques for solving the constraint satisfaction problem: problem reduction, search and solution synthesis. Problem reduction is often referred as consistency maintenance in the literature. This includes node consistency (NC), arc consistency (AC) and path consistency (PC) or k-consistency. There are efficient algorithms developed to achieve NC and AC (Tsang 1993). An important observation is that reducing a problem to a minimal problem, where no more redundant values or compound labels can be removed from the domain of the problem, is often NP-hard. In other words, only easy redundant values and compound labels are removed. This is why a search procedure is often needed. Thus the efficiency of constraint satisfaction algorithms will be greatly limited by the procedures of tree search.

The most widely used technique for domain reduction is called constraint propagation. The idea is that when a variable's domain is modified, the effects of this modification are then communicated to any constraint that interacts with that variable. In other words, the domain reduction algorithm modifies the domain of all the variables in that constraint, given the modification of one of the variables in that constraint.

Constraint programming algorithms can also be used to solve optimization problems by gradually tightening a bound on the objective function. Unfortunately, it is often not efficient to apply this scheme in practice. In order to successfully integrate constraint programming with optimization the analyst must take advantage of both commonalities and differences between the techniques. The differences, or complementary strengths, indicate the opportunity for profit from integration; while commonality often makes the integration natural and easier (Hooker 2002). There are two areas of commonality. First, both approaches rely heavily on branching search, at least when an exact solution is required and the problem is combinatorial in nature. Second, both use logical inference methods to accelerate search (but in different ways).

The differences between the techniques are also important. First, constraint programming formulates a problem within a programming language. Its declarative nature can substantially reduce the coding effort. More importantly, it separates the model and the search procedure, which makes the model more robust to the changing environment (. This is an advantage over some problem specific algorithms, which are sensitive to the structure of the problem. Second, they use logical inference in different ways. CP uses logical inference to reduce the search space directly through such techniques as domain reduction and constraint propagation, while optimization methods use inferences (as in cutting planes method) to create better relaxations.

The versatile modeling framework of constraint programming leads to models that are more readable and easier to develop and debug than optimization models. Conversely, powerful optimization methods can solve structured sub-problems more efficiently than constraint programming methods.

Constraint programming was designed for constraint satisfaction problems but it is also especially useful in solving scheduling problems. The reason can be traced to advantages and drawbacks of constraint satisfaction techniques. To be specific, when a constraint contains a cost or profit function of many variables, domain reduction (or consistency maintenance) becomes ineffective (usually there are effective algorithms for binary constraints, which contains only two variables in a constraint). The optimization community overcomes this difficulty by using relaxation techniques. The constraint satisfaction community, by contrast, developed to a large degree through binary constraint problem (or 2-SAT problem), where more efficient AC algorithms can be applied. There are at least two characteristics of scheduling problem that cater to constraint satisfaction technique's specialty. First, the temporal relations are usually binary constraints. For example, when we specify that activity b1 must precede activity c3, there are exactly two variables involved in this constraint. Second, the objective is often minimizing make-span, where there are not many variables involved. Jain and Grossmann (2001) proposed algorithms integrating constraint programming and MILP model to solve a scheduling problem and reported very good results.

We propose a constraint programming approach to solve the RCPSP because of its following advantages. First, constraint satisfaction technique is good at handling binary constraints, which populate the RCPSP problem. In fact, binary constraints are ubiquitous in scheduling problems and since constraint programming is efficient at dealing with binary constraints we expect it to substantially reduce the solving time in this class of problems. Preliminary investigations show reductions by factors of 10 in small problems.

In particular we plan to make use of a product called OPL in this investigation. OPL is a computer application for constraint programming (Hentenryck 1999). OPL is the newest of these tools and promises to be effective in precisely the kinds of problem we propose to solve here. OPL is a product sold by ILOG, an innovative company in the forefront of software applications for mathematical and constraint programming.

OPL's declarative nature makes the model readable and easy to develop. Hence it becomes a particularly attractive programming tool. In addition, with OPL the search procedure can be separated from the model. This feature makes the model robust with the changing environment. In other words, we can add, delete or modify the temporal constraints, task durations and resource availability freely without hampering the validity of the model. This is an obvious advantage compared with some problem specific algorithms whose performance is sensitive to the problem at hand. Finally, through OPL's connectivity feature with databases, such as Microsoft Access and Oracle we can easily develop user-friendly applications that require good management of databases, as in the case of RCPSP. Furthermore, OPL has API's for embedding OPL models with traditional programming language such as C++, Java, VB and VBA, which makes the development of independent application software solving optimization problems possible.

This research effort will:

1. Construct a scheduling problem with characteristics similar to the problem of scheduling sailors to tasks on board a small combatant.
2. Implement the problem in OPL as a constraint programming problem and in CPLEX as a mixed integer programming problem.
3. Compare the results of solving the problem under a variety of circumstances.

References:

Freuder E. 1997, Constraints: An International Journal, 2, pp.57-61

Hartmann Sonke 1999, Project Scheduling under Limited Resources, 478 Lecture Notes in Economics and Mathematical Systems, Springer

Hentenryck P.V., 1997, "Constraint Programming for Combinatorial Search Problems", Constraints: An International Journal, 2, 99-101.

______________, 1999, "The OPL -- Optimization Programming Language", The MIT Press, Cambridge, Massachusetts.

Hooker J., 2000, "Logic-Based Methods for Optimization", Wiley-Interscience Series in Discrete Mathematics and Optimization.

_______, 2002, “Logic, Optimization, and Constraint Programming”, INFORMS Journal on Computing, Vol.14, No.4, Fall 2002 pp.295-321.

Jain V., I. E. Grossmann, 2001, “Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems”, INFORMS Journal on Computing, Vol.13, No.4, Fall 2001 pp.258-276.

Lustig, I.J., and J.-F. Puget, 2001, “Program does not equal program: Constraint programming and its relationship to mathematical programming”, *Interfaces* Vol. 31, pp. 29-53.

Neumann K., C. Schwindt, J. Zimmermann, 2001,“Project Scheduling with Time Windows and Scarce Resources”, 508 Lecture Notes in Economics and Mathematical Systems, Springer-Verlag.

Sprecher A., 1994, “Resource-Constrained Project Scheduling”, 409 Lecture Notes in Economics and Mathematical Systems, Springer-Verlag.

Tsang, 1993, Foundations of Constraint Satisfaction