PSPSolver: An Open Source Library for the RCPSP

Javier Roca, Filip Bossuyt
Intelligent Software Company
Chaussée de Nivelles 121/2, 7181 Arquennes, Belgium

Gaetan Libert
Computer Science Department - Faculté Polytechnique de Mons
Rue de Houdain 9, 7000 Mons, Belgium

{javier.roca, filip.bossuyt}@planningforce.com, gaetan.libert@fpms.ac.be

Abstract

The Resource-Constrained Project Scheduling Problem (RCPSP) is a classical problem in project scheduling. The most common and successful approaches to solve the RCPSP are those applying heuristics, metaheuristics and sampling schemas, given their practicability and effectiveness. In most of the cases these approaches apply a Schedule Generation Schema (SGS) combined with a suitable solution representation and priority rules. Although there is a considerable research in these RCPSP solving methods and its theory there is a lack of software for supporting the research of new solving methods. In many cases, the RCPSP research requires the implementation of algorithms in order to validate or evaluate a solving method. We introduce PSPSolver (Project Scheduling Problem Solver), an extensible and practical heuristic-based library for supporting the research on solvers for the RCPSP.

1.The RCPSP Model

Informally, the single mode RCPSP model, simply referred as RCPSP, is a well known project scheduling problem (PSP) that seeks the answer to the following question: *"Given the limited availability of resources, what is the best way to schedule the activities in order to complete the project in the shortest possible time?"*. The RCPSP is of special interest in fields like construction and production scheduling. Conceptually, the RCPSP is a PSP with single mode activities, renewable constrained resources, finish-to-start precedence relationships with zero time lags, no pre-emption, and has the makespan minimization as the performance measure. According to Bucker et al. [1], this problem is denoted as *PS | prec | Cmax* (machine scheduling domain). Herroelen et al. [2] denotes this model as *m,1 | cpm | Cmax*.

Due to the fact that the RCPSP forms the core problem among the class of resource-constrained project scheduling problems [1], every improvement in its resolution can produce new advances in the resolution of the other models. The RCPSP instances are usually represented as A-O-N digraphs (Figure 1), while the RCPSP solutions (schedules) are represented as Gantt charts (Figure 2).

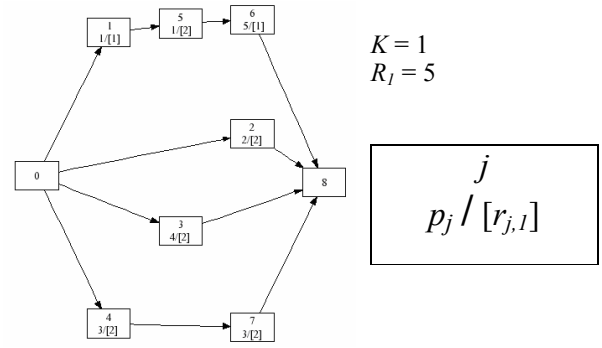

Figure 1: A RCPSP Instance

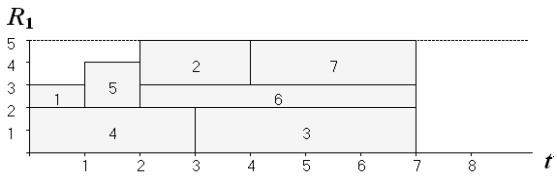

Figure 2: An optimal schedule for the instance in Figure 1.

2. Solving the RCPSP

The main approaches to solve the RCPSP are the optimal (exact) methods, heuristics, and the metaheuristics-based solution procedures. It has been shown by Blazewicz et al. [3] that the RCPSP, as a generalization of the classical job shop scheduling problem, belongs to the class of *NP*-hard optimization problems. Therefore, heuristic solution procedures are indispensable when solving large problem instances as they usually appear in practical cases [4].

2.1. Schedule Generation Schemes

Schedule generation schemes are the core of most of the heuristic/metaheuristic solution procedures for the RCPSP [4]. A SGS is a constructive technique that builds a feasible schedule by stepwise extension of a partial schedule (i.e. a schedule where only a subset of the activities have been scheduled). There are two different

types of SGS: serial SGS (S-SGS) and parallel SGS (P-SGS). The S-SGS is an *activity oriented* SGS that performs activity incrementation while building the schedule. The P-SGS, is a *time oriented* SGS that performs time incrementation in the schedule build. For a formal and detailed definition of the SGS the reader is referred to [4].

3. The PSPSolver Library

The main motivation for the development of our PSPSolver is the lack of freely available software for the RCPSP. The main goal is to build an extensible environment for abstracting RCPSP instances and solutions with the implementation of SGS-based solving methods by using modern programming concepts. PSPSolver provides an extensible object-oriented application programming interface (API) for the visualization, representation, and solving of RCPSP instances. The library is currently implemented in C# and can be freely downloaded from *http://www.planningforce.com/wiki/*. The code distribution includes detailed documentation and ready-to-use code snippets. The reader is referred to this documentation for a detailed description of PSPSolver's features (e.g. easy of use, extensibility, performance, limitations, comparison with other software, etc.).

3.1. Problem and Solution Representation

The library provides classes to represent single mode RCPSP instances and schedules, nevertheless, other PSP models could be easily extended as well (i.e. multi-mode RCPSP). PSPSolver provides mechanisms for handling RCPSP logical instances by supporting common file formats (i.e. the formats proposed by PSPLIB[5]), and also defines a new normalized XML-based representation, best suitable for data exchange between applications. The library can model and be extended with user defined priority rules as well.

3.2. Visualization

PSPSolver is able to render RCPSP instances as A-O-N digraphs and also instance solutions as Gant Charts. In order to implement a clear visual representation of the instance, the network rendering relies on features of the GraphViz graph rendering engine [6]. This feature is of great utility especially when we want to visualize a complex topology on large RCPSP instances. The diagrams illustrated in Figure 1 and Figure 2 were rendered by using the PSPSolver visualization API.

3.3. Solving

PSPSolver provides an API for solving RCPSP instances by implementing the S-SGS and the P-SGS applying user defined heuristics (priority rules). Additionally, one of the most important features in the Solving API is the possibility to easily integrate or be integrated in custom scheduling metaheuristics (e.g. ACO, TS, PSO, SA, etc.). In brief, PSPSolver is able to solve a PSP using user-defined heuristics (priority rules and/or metaheuristics) by implementing the SGS approach. The library code distribution includes a self-contained example (the PSPViewer application) that illustrates the main features of the PSPSolver API by implementing an instance renderer, a solution benchmarker, a solver, an illustrated custom priority rule, and a solution renderer. As a reference, PSPViewer was able to solve the 480 instances from the J30-SM set (PSPLIB) in less than 2s. (an average of 4ms. per instance), using a S-SGS and the SPT (Shortest Processing Time) heuristic in a Pentium 2.0 GHz with 1MB of RAM using Visual C# Express 2005 and .Net Framework 2.0.

4. Conclusions

We consider that PSPSolver is a basic but powerful free library for solving the RCPSP model, and we consider it a valuable and practical tool for the PSP research community. The library can be easily adapted to the researcher's needs, in order to implement new SGS-based heuristics. We plan to improve the library with the implementation of a lower-bound calculation, double justification (schedule optimization)[7] and extending it to the multi-mode RCPSP. We are currently working in porting the library to the JAVA programming language as well.

References

[1] Brucker, P., Drexl, A., Mohring, R., Neumann, K., and Pesh, E., 1999. *Resource-constrained project scheduling: Notation, classification, models, and methods*, European Journal of Operational Research 112

[2] Herroelen, W., De Reyck, B., and Demeulemeester, E., 1998. *Resource-constrained project scheduling: a survey of recent developments*, Computers Ops Res. Vol. 25

[3] Blazewicz, J., Cellary W., Slowinski R., and Weglarz J., 1986. *Scheduling under Resource Constraints: Deterministic Models*, in: *Annals of operations research, vol. 7*., J.C. Baltzer

[4] Kolisch, R. and Hartmann, S. eds. 1999. *Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis*, in: *Project scheduling: Recent models, algorithms and applications*, Kluwer

[5] Kolisch, R., and Sprecher, A. 1997. *PSPLIB A Project Scheduling Problem Library*, European Journal of Operational Research 96

[6] Gansner, E., and E., North, 2000. *An open graph visualization system and its applications to software engineering*, Software Practice and Experience Vol.30, 11

[7] Valls, V., Ballestín, F., Quintanilla, S., 2006. *Justification Technique Generalizations,* in: *Perspectives in Modern Project Scheduling*, Kluwer