

# Chapter 74

## Scheduling Large and Complex IT Projects Using Sliding-Frame Approach

**Yuval Cohen**

*Open University and Afeka Academic College, Israel*

**Arik Sadeh**

*Holon Institute of Technology, Israel*

**Ofer Zwikael**

*Australian National University, Australia*

### **ABSTRACT**

*Many IT projects and software development projects are very complex and sophisticated involving a large coordinated team. Such projects are a constant part of the operations of software companies such as Microsoft, SAP, Oracle, Google, Yahoo, IBM, and others. Many other companies carry large software projects as part of their IT operations. As a result of the size and complexity of such projects, a rolling horizon approach for their planning and management is not only plausible but also desirable. For large projects, traditional project scheduling techniques cannot provide an optimal and timely solution to minimum project duration under precedence and resource constraints. This paper proposes a technique that allows utilizing non-polynomial (NP) algorithms in a heuristic manner – generating high quality schedules in reasonable time. This approach can be applied efficiently for solving most project scheduling problems. The main advantage of this approach is its ability to dissect the original problem into small sub-problems of a controllable size to which exact techniques can be applied. Thus, it neutralizes the complexity of the applied algorithms (and their non-polynomial growth). Moreover, the same technique could be used for implementing a rolling-horizon approach in project planning and management.*

DOI: 10.4018/978-1-4666-4301-7.ch074

## INTRODUCTION

Many IT Projects are large and complex giving rise to complicated scheduling problems characterized by the presence of both precedence constraints and resource constraints (Whitty & Maylor, 2009). This combination of constraints greatly complicates the scheduling and causes the project scheduling complexity to grow in a non-polynomial (NP) rate, effectively blocking the possibility of solving large problems optimally (Demeulemeester & Herroelen, 2002). The simplest form of these NP scheduling problems is the classical Resource Constrained Project Scheduling (RCPSP) (e.g., Kolisch & Hartman, 1999; Leung, 2004; Rivera & Duran, 2004) with one or many resources. Throughout the years, many complications have been added to the RCPSP, making it even more intricate: multi-mode resource availability (Wuliang & Chengen, 2009), uncertainty and stochasticity (Ballestín & Leus, 2009; Cohen & Zwikael, 2008; Kirytopoulos, Leopoulos, & Diamantias, 2008), multiple criteria (Maria, Teixeira, & Alencar, 2009), and resources leveling (Kastor & Sirakoulis, 2009) – to mention just a few.

The aim of these problem variations is to find the schedule with the shortest duration, setting the order and timing of each activity. This is more challenging than estimating the project duration by shortcuts such as the critical path (Zammori, Braglia, & Frosolini, 2009) or the critical chain (Ash & Pittman, 2008).

However, when it comes to large projects with hundreds or even thousands of activities it is impractical to use exact methods for finding the optimal solution (Demeulemeester & Herroelen, 2002; Ling & Tiong, 2008). The reason for this is that RCPSP is known to be non-polynomial (NP hard) (Demeulemeester & Herroelen, 2002; Kolisch & Sprecher, 1996; Kolisch & Hartman, 1999; Leung, 2004). This means that none of the exact optimal techniques can solve the problems faced by large projects in a timely manner. The current trend of managing multi-project envi-

ronments makes planning and scheduling even more challenging (Laslo & Goldberg, 2008). The aforementioned large complexity requires a new approach, and this paper aims to suggest and investigate one.

In addition to the size and complexity of IT projects, their dynamic nature makes a rolling horizon approach desirable for planning and management (Collyer & Warren, 2009). A similar conclusion for other complex projects was found by Zanelidin (2009). This paper presents a practical technique that enables planning and even managing a large, complex, and sophisticated IT project in a rolling horizon manner. The main idea is to build a sequence of activities (a rail) and a sliding frame. The rail serves the frame as rails serve a train – as a linear basis along which to move. Each time the frame is set, the focus is on optimizing the frame. The frame slides in steps small enough to overlap the previous step. The paper discusses how to build the rails of the sliding frame, and the pros and cons of solving the frames rather than solving the whole problem.

The idea of dissecting or slicing large problems into sub-problems and solving them, rather than solving the whole problem at once, has been very attractive in operations research (Evans & Minieka, 1992). Such an approach can effectively turn an exponential complexity into a polynomial complexity. A famous example of non-polynomial complexity is the Traveling Salesperson Problem (TSP): the problem of a sales person who must visit several cities and wants to choose the best order of cities for the shortest route that will bring him back home. The enumeration algorithm for solving the TSP has a factorial computational time of  $n!$ . If the set of TSP points could be sliced into equal pieces of, say, 10 in each set, then solving each piece separately would give an over all solution based on  $n/10$  local optimum points. In such an instance, the number of computations for solving each piece is  $(10!)$  (a constant) and the total complexity drops to:  $(n/10)(10!) = n(9!)$ . Since  $9!$  is a constant number:  $n(9!) \sim O(n)$ . The biggest obstacle

11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/scheduling-large-complex-projects-using/77769](http://www.igi-global.com/chapter/scheduling-large-complex-projects-using/77769)

## Related Content

---

**The Multi-Agents Architecture for Emotion Recognition: Case of Information Retrieval System**  
Mohamed Néji, Ali Waliand Adel M. Alimi (2014). *International Journal of Software Innovation* (pp. 73-85).  
[www.irma-international.org/article/the-multi-agents-architecture-for-emotion-recognition/111451](http://www.irma-international.org/article/the-multi-agents-architecture-for-emotion-recognition/111451)

**Open Source Software: A Key Component of E-Health in Developing Nations**  
David Parry, Emma Parry, Phurb Dorjiand Peter Stone (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1628-1640).  
[www.irma-international.org/chapter/open-source-software/29468](http://www.irma-international.org/chapter/open-source-software/29468)

**A Critical Systems View of Power-Ethics Interactions in Information Systems Evaluation**  
José-Rodrigo Córdoba (2010). *Emerging Systems Approaches in Information Technologies: Concepts, Theories, and Applications* (pp. 257-273).  
[www.irma-international.org/chapter/critical-systems-view-power-ethics/38184](http://www.irma-international.org/chapter/critical-systems-view-power-ethics/38184)

**Survey of Vulnerabilities and Mitigation Techniques for Mooc-Based Applications**  
Hossain Shahriar, Hisham M. Haddad, David Lebronand Rubana Lupu (2016). *International Journal of Secure Software Engineering* (pp. 1-18).  
[www.irma-international.org/article/survey-of-vulnerabilities-and-mitigation-techniques-for-mooc-based-applications/176398](http://www.irma-international.org/article/survey-of-vulnerabilities-and-mitigation-techniques-for-mooc-based-applications/176398)

**Katana: Towards Patching as a Runtime Part of the Compiler-Linker-Loader Toolchain**  
Sergey Bratus, James Oakley, Ashwin Ramaswamy, Sean W. Smithand Michael E. Locasto (2012). *Security-Aware Systems Applications and Software Development Methods* (pp. 217-233).  
[www.irma-international.org/chapter/katana-towards-patching-runtime-part/65850](http://www.irma-international.org/chapter/katana-towards-patching-runtime-part/65850)