Available online at www.sciencedirect.com
SCIENCE DIRECT
Int. J. Production Economics 97 (2005) 227–240
international journal of production economics
www.elsevier.com/locate/dsw

The use of buffers in project management: The trade-off between stability and makespan

Stijn Van de Vonder*, Erik Demeulemeester, Willy Herroelen, Roel Leus

Department of Applied Economics, KU Leuven, Leuven, Belgium

Received 29 March 2004; accepted 20 August 2004
Available online 30 November 2004

Abstract

During execution projects may be subject to considerable uncertainty, which may lead to numerous schedule disruptions. Recent research efforts have focused on the generation of robust project baseline schedules that are protected against possible disruptions that may occur during schedule execution. The fundamental research issue we address in this paper is the potential trade-off between the quality robustness (measured in terms of project duration) and solution robustness (stability, measured in terms of the deviation between the planned and realized start times of the projected schedule). We provide an extensive analysis of the results of a simulation experiment set up to investigate whether it is beneficial to concentrate safety time in project and feeding buffers, or whether it is preferable to insert time buffers that are scattered throughout the baseline project schedule in order to maximize schedule stability.

Keywords: Project scheduling; Schedule stability; Quality robustness; Buffers

1. Problem description

The vast majority of the research efforts in project scheduling over the past several years have concentrated on the development of exact and suboptimal procedures for the generation of a *baseline schedule* (*pre-schedule*, *predictive schedule*) assuming a deterministic environment and complete information. During execution, however, a project may be subject to considerable uncertainty, which may lead to numerous schedule disruptions. The recognition that uncertainty lies at the heart of project planning has induced a number of research efforts in the field of project scheduling under uncertainty (for an extensive review of the literature we refer to Demeulemeester and Herroelen (2002) and Herroelen and Leus (2004b)).

Critical chain scheduling/buffer management (CC/BM)—the direct application of the Theory of Constraints (TOC) to project management (Goldratt, 1997)—has received a lot of attention in the project management literature. The

*Corresponding author.
E-mail address: stijnvandevonder@econ.kuleuven.be (S. Van de Vonder).

0925-5273/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2004.08.004

fundamental working principles of CC/BM have been reviewed by Herroelen et al. (2002). CC/BM builds a baseline schedule using aggressive[1] median or average activity duration estimates. The safety in the durations of activities that was cut away by selecting aggressive duration estimates is concentrated in the form of a *project buffer* (PB) that is positioned at the end of the so-called critical chain. The critical chain (CC) is defined as the longest chain[2] of precedence and resource dependent activities that determines the overall duration of a project. The project buffer should protect the project due date from variability in the critical chain activities. *Feeding buffers* (FB) are inserted whenever a non-critical chain[3] activity joins the CC. Clearly, the CC/BM idea is to protect the project due date against the disruptions that may occur during project execution. As such it can be viewed as a procedure for generating *makespan* (*due date*) *protective schedules*. Due date protection, however, is only one side of the coin and relates to the sensitivity of the project makespan to activity disruptions, i.e. to the *quality robustness* of the baseline schedule. For executing a project, on the other hand, the CC/BM approach does not rely on the buffered schedule but on a so-called *projected schedule*. This schedule is precedence and resource feasible and is to be executed according to the roadrunner mentality, i.e the so-called gating tasks (activities with no non-dummy predecessors) are started at their scheduled start time in the buffered schedule while the other activities are started as soon as possible. The projected schedule is recomputed when disruptions occur. Neither the buffered schedule nor the projected schedule are constructed with a view to stability (*solution robustness*, i.e. the insensitivity of planned activity start times to schedule disruptions).

An ideal schedule should combine solution robustness (i.e. be stable) and quality robustness (i.e. be makespan protective). The fundamental research issue we address in this paper is the potential trade-off between quality robustness (measured in terms of project duration) and solution robustness (stability, measured in terms of the deviation between the planned and realized start times) of the projected schedule. By means of simulation we investigate whether it is beneficial to concentrate safety time in project and feeding buffers as done by the *original* CC/BM approach and a *modified* CC/BM approach developed in this paper, or whether it is preferable to insert time buffers scattered throughout the project schedule, as done by the *adapted float factor model* (ADFF) developed by Leus (2003).

The remainder of the paper is organized as follows. The next section describes the set-up of our computational experiment. We describe the scheduling mechanism used by the *original* and *modified* CC/BM approach and the ADFF heuristic. The *original* and *modified* CC/BM approaches are used as representatives of scheduling algorithms that aim at generating makespan protective schedules. ADFF is used as a representative of scheduling algorithms that aim at generating solution robust schedules. We also describe the metrics used for measuring solution and quality robustness. Section 3 describes the experimental results obtained by the three scheduling heuristics. The last section is reserved for our overall conclusions.

2. Set-up of the computational experiment

We assume that projects are represented in activity-on-the-node representation, where the precedence constraints are of the finish-start type with zero time-lag. An example network with ten activities is given in Fig. 1. Nodes 0 and 9 are the dummy start and end nodes, respectively. We make abstraction of resource usage and assume that activity durations are random variables with known distribution. The first number above each node represents the corresponding mean activity duration, to be used in generating a baseline

[1] Goldratt (1997) proposes to build schedules by using activity durations that cover the time to do the work without any safety, based on a 50% confidence level rather than on the 80–90% confidence levels that he claims to be in common use in project management practice. These durations are called aggressive.

[2] If there is more than one critical chain, an arbitrary choice is made.

[3] A non-CC can be every chain of one or more activities that does not determine the project lead time.