

Chapter 2.19

Optimal Crashing and Buffering of Stochastic Serial Projects

Dan Trietsch

American University of Armenia, Armenia

ABSTRACT

Crashing stochastic activities implies changing their distributions to reduce the mean. This can involve changing the variance too. Therefore, crashing can change not only the expected duration of a project but also the necessary size of its safety buffer. We consider optimal crashing of serial projects where the objective is to minimize total costs including crashing cost and expected delay penalty. As part of the solution we determine optimal safety buffers. They allow for activities that are statistically dependent because they share an error element (e.g., when all durations have been estimated by one person, when weather or general economic conditions influence many activities, etc). We show that under plausible conditions the problem is convex and thus it can be solved by standard numerical search procedures. The purpose of the paper is to encourage software development that will include valid stochastic analysis for scheduling and crashing using current estimates and historical performance records.

INTRODUCTION

Historically, crashing project activities has been associated with CPM, using deterministic activity duration assumptions. By contrast, PERT took into account stochastic activity durations but with few exceptions crashing was not considered. One

possible reason for this state of affairs may be that PERT was never a truly effective platform for assessing stochastic activity durations. Baker & Trietsch (2009) list several PERT deficiencies, of which the stochastic independence assumption is the most pernicious. To address this issue, Trietsch (2005) presents the *systemic error model*, which employs a multiplicative stochastic error that is constant for each project but varies across

DOI: 10.4018/978-1-61350-456-7.ch2.19

projects. This component of the model represents estimation bias, and it entails positive correlation between the activity durations of each project as compared to the original project schedule. One advantage of the model is that it is straightforward to use historical data to achieve reliable distributions without requiring practitioners to provide three estimates for each activity. That model has recently been validated by field data from two Armenian NGOs (Gevorgyan, 2008). The validation also suggested that the lognormal distribution with a consistent coefficient of variation is useful for modeling activity durations (although in one NGO it was also necessary to account for the Parkinson effect, where activities are often tardy but rarely early). The lognormal distribution is especially convenient if we wish to crash an activity by allocating more capacity to it, because that can be modeled by division of work content by capacity. If both work content and capacity are modeled as lognormal random variables, their ratio is also lognormal. No other widely-used distribution shares this useful property. Furthermore, although the exact distribution of a sum of lognormal variables is unknown, the lognormal distribution itself provides reasonable approximate convolutions (Robb & Silver, 1993).

Those results open the way not only to effective stochastic scheduling but also to modeling and implementing optimal crashing of a stochastic project. Baker & Trietsch (2009) discuss this issue in generic terms, and the purpose of this paper is to explore it in more depth. We do so specifically in terms of a simple but important building block: a serial project. We study optimal or near-optimal crashing of stochastic activities in basic projects with n (not necessarily independent) activities in series, and without intermittent idling. We assume continuous crashing with a linear cost per unit as in classical CPM (Fulkerson, 1961; Kelley, 1961). A due date, D , is given and tardiness is penalized at a proportional rate, P . The objective is to minimize the total cost of crashing plus expected delay penalty. (We refer to this as

the *default* objective.) If we define the difference between the due date and the mean completion time as the *project buffer*, then optimal crashing yields optimal project buffers. For this reason we consider the two issues together. Our purpose is to provide a theoretical basis for enhancing project scheduling and control software packages, to handle stochastic analysis properly and without demanding more input from the user than is customary for deterministic analysis.

To review the literature related to stochastic crashing of project activities, the following statements are implied (unless stated otherwise): (1) General PERT networks with statistically independent stochastic activities are addressed. (2) Crashing is continuous. (3) The default objective is pursued (with the “cost of crashing” interpreted according to the context). An alternative, to which we refer explicitly as the *expectation* objective, is to either minimize the expected duration given a crashing budget or achieve a required expected duration most economically (the ability to solve one suffices for the other). The expectation objective is a special case of the default objective, obtained if we set a due date of zero and adjust the penalty rate. Finally, (4) no intentional idling occurs—each activity starts as soon as it becomes feasible. In these terms, our own problem is “serial networks with dependent activity distributions.”

Arisawa & Elmaghraby (1972) considered GERT networks where only exclusive-or nodes are allowed (thus excluding general PERT but including the serial case) with an expectation objective, and presented a tractable fractional LP solution. Britney (1976) explicitly addressed both buffers and crashing with a sum of several default objectives as the combined objective, and with possible idling between all activities. His buffers apply to activities one-by-one and planned idling occurs unless the buffer is exceeded. Wollmer (1985) pursued an expectation objective where crashing applies only to activity means. He modeled stochastic variation by discrete random variables, and obtained a convex programming

10 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/optimal-crashing-buffering-stochastic-serial/62460

Related Content

A Cloud Broker for Service Adaptation Based on Interface Localization

Claus Pahland Luke Collins (2018). *Multidisciplinary Approaches to Service-Oriented Engineering* (pp. 96-120).

www.irma-international.org/chapter/a-cloud-broker-for-service-adaptation-based-on-interface-localization/205295

Science Communication with Dinosaurs

Phillip L. Manningand Peter L. Falkingham (2012). *Handbook of Research on Computational Science and Engineering: Theory and Practice* (pp. 587-611).

www.irma-international.org/chapter/science-communication-dinosaurs/60376

Object-Oriented Cognitive Complexity Measures: An Analysis

Sanjay Misraand Adewole Adewumi (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 917-940).

www.irma-international.org/chapter/object-oriented-cognitive-complexity-measures/192907

Shaping Competitive Strategies for the Computer Industry

Shameem Akhter, Nayem Rahman, Mahmud Ullahand Mohammad Nirjhar Rahman (2019). *Handbook of Research on Technology Integration in the Global World* (pp. 189-207).

www.irma-international.org/chapter/shaping-competitive-strategies-for-the-computer-industry/208799

True Color Image Segmentation by MUSIG Activation Function Using Self-Supervised QMLSONN Architecture With Context-Sensitive Thresholding

Pankaj Paland Siddhartha Bhattacharyya (2018). *Quantum-Inspired Intelligent Systems for Multimedia Data Analysis* (pp. 213-261).

www.irma-international.org/chapter/true-color-image-segmentation-by-musig-activation-function-using-self-supervised-qmlsonn-architecture-with-context-sensitive-thresholding/202549