

Project-Driven Supply Chains: Integrating Safety-Stock and Crashing Decisions for Recurrent Projects

Xin Xu, Yao Zhao

Department of Supply Chain Management and Marketing Sciences

Rutgers University, Newark, NJ 07102

xin.xu.2009@rutgers.edu, yaozhao@andromeda.rutgers.edu

Ching-Yu Chen

Department of International Business, Tunghai University, Taiwan

amychen@thu.edu.tw

Abstract

We study a new class of problems – recurrent projects with random material delays, at the interface between project and supply chain management. Recurrent projects are those similar in schedule and material requirements. We present the model of project-driven supply chain (PDSC) to jointly optimize the safety-stock decisions in material supply chains and the crashing decisions in projects. We prove certain convexity properties which allow us to characterize the optimal crashing policy. We study the interaction between supply chain inventory decisions and project crashing decisions, and demonstrate the impact of the PDSC model using examples based on real-world practice.

Keywords: Recurrent projects, random material delays, dynamic programming, crashing, safety-stock.

1 Introduction

Today, complex projects frequently spend a significant portion of their budgets on material supplies sourced from an extensive supply network. The success of these projects depends critically on both the project and the supply chain operations. Examples can be found in construction and

airspace/defense industries in particular, and in engineering-procurement-construct (EPC) industries in general.

While some of these projects are unique and one of a kind, many of them are routine and recurrent, i.e., they share similar schedule and material supplies. Indeed, as companies standardize their processes and components to streamline their operations, recurrent projects are becoming increasingly popular in practice. In these cases, the project and supply chain operations are intertwined because the demand for the supply chain is driven by projects' material requirement and project progresses are constrained by random material delays. Despite the extensive literature of project and supply chain management, project operations and supply chain operations are rarely studied jointly in academia, and they are almost always managed separately in industry. This paper explores ways to improve the efficiency of recurrent projects by integrating project and supply chain operations.

Specifically, we study a new class of problems - recurrent projects with random material delays where the future projects' occurrence and material requirement cannot be fully predicted in advance. The project and supply chain decisions are naturally coupled because under-stocked supply chains lead to excessive material delays which may increase project crashing (i.e., expediting) and delay costs. To track down these problems, we develop a modeling framework to integrate project crashing and supply chain safety-stock decisions. Our objective is to validate a new approach - carrying safety stock for materials used in recurrent projects, in performance improvement for projects and their material supply chains.

One motivating example is the Intercontinental Construction Management (ICM) Inc., a construction management firm specialized in military buildings (the name is disguised to protect proprietary information). Although each construction project has a unique goal, they are recurrent - similar in schedule and material requirement. Structural steel is the most expensive material used in all projects, which is subject to a long and random lead time. Because projects are awarded through a bidding process, ICM cannot fully anticipate the occurrence of future projects and their material requirement, and thus cannot order materials before projects are awarded. Consequently, structural steel may be delayed behind schedule, in which case, ICM has to crash (i.e., expedite) the remaining tasks as needed to meet the due date.

ICM has never looked at the safety-stock of structural steel as a part of the solution because it assigns each project to a project manager who manages the project as a separate and unique

entity. ICM's practice does not stand alone - project and supply chain are rarely managed jointly in practice because of the distinct approaches to manage them. Projects are often managed on a one-for-one basis while supply chains are often managed continuously across demand from multiple projects. Such a one-for-one (or project-based) approach fails to take advantage of the similarity across recurrent projects.

Recurrent projects are widely seen in many industries, e.g., construction and build-to-order manufacturing, where standard components are combined in different ways to build complex and customized products (e.g., houses, ships, aircrafts). In addition to ICM, other examples of recurrent projects can be found in Walsh, et al. (2004), Brown, et al. (2004) and Elfving, et al. (2010). Specifically, Walsh, et al. (2004) presents a case study of a food company that is frequently engaged in projects of expanding an existing or adding a new facility. The key concern is on the critical material of stainless steel components that are used in all projects and are subject to the longest and most variable lead time. Brown, et al. (2004) presents the case study of Quadrant Homes Inc. which follows a standard schedule in construction of residential houses. Elfving, et al. (2010) conducts an empirical study on 180 projects by one Finnish construction company in five years and shows that many projects share make-to-stock standardized materials. Finally, Schmitt and Faaland (2004) shows that in addition to houses, projects of constructing airplanes and ships can be recurrent.

The industry reports of Kerwin (2005) and Xu and Zhao (2010) further confirm the popularity of recurrent projects in the construction industry. Kerwin (2005) shows that home-builders like Pulte Homes Inc. offer only the most popular floor plans to boost efficiency in fulfilling tens of thousand of new home orders in a year. Xu and Zhao (2010) surveys an important trend in construction industry – prefabricated housing, where houses have limited variety and are assembled by prefabricated materials. All these cases are characterized by repeating projects with limited variety and their supply chains of standardized materials (see Figure 1 for an illustration).

One key issue of the recurrent projects is the coordination of material delivery and project progress which is critical for on-time and on-budget completion of the projects. This is especially true when materials account for a significant portion of the budget (e.g., 65% for residential houses – Somerville 1999), and when material availability and delivery lead times are random, as illustrated by the examples of ICM and Walsh, et al. (2004), and confirmed by an investigation of time waste in construction (see Yeo and Ning 2002) which reveals that the site work-force spends a considerable

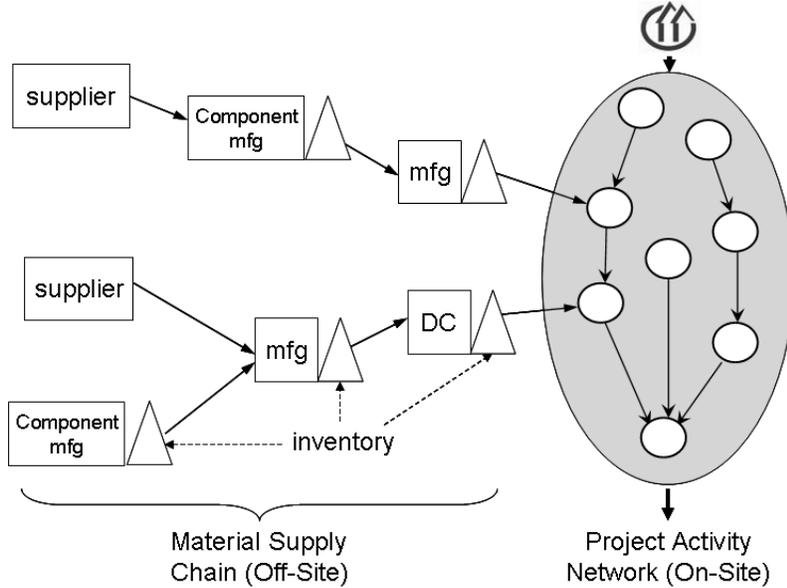


Figure 1: Recurrent projects with standardized material requirements.

amount of time waiting for approval or for materials to arrive on site.

One way to improve the efficiency of recurrent projects is to explore the similarity among these projects and plan for them on a continuing basis rather than for each project separately. Specifically, the supply-based options, such as inventory management and strategic alliance, have been considered to improve material supply management for recurrent projects. We refer the reader to Walsh, et al. (2004), Brown, et al. (2004), and Elfving, et al. (2010) for case studies, and Tommelein, et al. (2003), Tommelein, et al. (2009) for conceptual frameworks. These studies focus only on material supplies rather than the integration between project-based and supply-based decisions. However, the supply chain decisions are coupled with the project decisions: if materials are in shortage and the project due date cannot accommodate the delays, project tasks will have to be expedited which can be very costly, and the project is subject to delay. On the other hand, holding inventory to guarantee immediate availability of all materials may not be wise. While one can minimize the project crashing and delay costs, the supply chain inventory cost can be unaffordably high.

In this paper, we develop an integrated approach to optimize supply chain and project decisions simultaneously. Specifically, we provide a modeling framework, namely, the project-driven supply chain (PDSC) model, to jointly optimize supply chain safety-stock and project crashing decisions for recurrent projects with random material delays. In a nutshell, the PDSC model is a multi-stage mathematical model where we make supply chain inventory decisions in the first stage, and then

make crashing decisions dynamically as material delays are realized in subsequent stages. The objective is to minimize the total safety-stock and project cost per unit of time. We prove certain convexity properties of the cost functions and characterize the optimal crashing policy for each project. We study the interaction between supply chain inventory decisions and project crashing decisions. We also apply the PDSC model to a real-world example and demonstrate its potential by comparing to the current practice. Finally, we conduct an extensive numerical study to generate insights on when the PDSC model may provide significant savings.

The paper is organized as follows: after reviewing the related literature in §2, we present the modeling framework in §3. Then, we provide an analysis for the optimal crashing policy in §4, and study the interaction between inventory and crashing decisions in §5. In §6, we apply the model to ICM and quantify its impact. In §7, we conduct a numerical study to generate managerial insights. Finally, we conclude the paper in §8.

2 Literature Review

The project-driven supply chain (PDSC) model in particular and the integration of supply chain and project management in general are related to the literature of project management, supply chain management, and their interfaces such as project scheduling and material ordering (PSMO) and construction supply chain management. We shall review related literature in each area.

Project Management Literature. The project management literature focuses primarily on the planning and execution of a single project, which includes the classic results of critical path method (CPM), time-costing analysis (TCA), project evaluation and review techniques (PERT) and resource constrained project scheduling (RCPS). We refer to Ozdamar and Ulusoy (1995), Pinedo (2005) and Jozefowska and Weglarz (2006) for recent surveys. The time-cost analysis (TCA) or crashing analysis is a well developed technique in the project management literature to balance the duration and budget of a project. Most work on RCPS focuses on non-consumable and reusable resources such as machine and labor. For consumable resources (e.g., materials), the standard approach is to assume fixed lead times and then model material procurement processes as tasks.

Supply Chain Management Literature. There is a vast literature of supply chain inventory management mainly grown out of applications in the manufacturing industry. We refer to Zipkin (2000), Porteus (2002) and Axsater (2006) for comprehensive reviews. For inventory place-

ment/positioning models in general structure supply chains, we refer to Axsater (2006), Graves and Willems (2003) and Simchi-Levi and Zhao (2011) for recent reviews. Graves and Willems (2005) presents a general model to optimize stock decisions and supply chain configurations simultaneously for new product introductions. All of these works focus on material supply chains without considering the project decisions and their interactions.

Our work is related to three models of stochastic inventory systems. The first model, see Hadley and Whitin (1963), assumes full backorder and that the system fulfills demand as soon as on-hand inventory becomes available. The second model, see Hariharan and Zipkin (1995), assumes also full backorder but that the system fulfills demand only on or after it is due. Clearly, it is possible to hold inventory and demand (not due) simultaneously in the second model. The third model, see Graves and Willems (2000), assumes guaranteed service-time (unsatisfied demand is filled by extraordinary measures other than on-hand inventory) and that the system fulfills demand only on or after it is due. Graves and Willems (2003) calls the first two models “stochastic service-time” models, and the third one the “guaranteed service-time” model.

In this paper, we cannot assume the guaranteed service-time model because the random material delay (due to stock-out and/or random processing times) is a necessary part of the problem. We shall use the stochastic service-time models – both the first and the second models for different stages of the material supply chain.

Interface – Project Scheduling and Material Ordering (PSMO). One approach to incorporate consumable resources in project management is the PSMO model which jointly plans for project schedule and material order quantities. This approach is based on the observation that a project may repetitively require the same material over time. Given the project schedule, the timing and size of material requirement are known, which serve as input to optimize material order quantities so as to balance the fixed ordering cost and inventory holding cost. Clearly, the project scheduling and material ordering decisions are coupled, and the question is how to jointly optimize both sets of decisions for a project. Aquilano and Smith (1980) initiates this approach by considering joint CPM and MRP planning with constant task durations. Smith-Daniels and Aquilano (1984) and Smith-Daniels and Smith-Daniels (1987) present various extensions. More recently, Dodin and Elimam (2001) considers varying task duration, early reward/late penalty and quantity discount.

The PDSC model complements the PSMO model by taking uncertainty and the safety-stock

issue into account. While the PSMO model focuses on the cycle stock issues for a single project facing economies of scale in ordering, the PDSC model focuses on the safety-stock issues for recurrent projects subject to random material delays. While the PSMO model jointly optimizes project schedule and material order quantities, the PDSC model jointly optimizes project crashing decisions and material safety-stock levels.

Interface – Construction Supply Chain Management. The literature of construction management has traditionally focused on the management of individual projects (Tommelein, et al. 2003). Since middle 1990s, the supply chain management concepts and methodologies have been introduced into this field and gained substantial attention. However, supply chain management is still relatively new in the construction industry (O’Brien, et al. 2002, Tommelein, et al. 2003), and most published results focus on qualitative and conceptual frameworks (Vrijhoef and Koskela 2000, Vaidyanathan and Howell 2007) or on case studies (Walsh, et al. 2004, Brown, et al. 2004). There is a lack of rigorous mathematical modeling that integrates the issues of projects and material supply chains, and resolves them jointly. We refer to O’Brien, et al. (2002) for a survey on construction supply chain management.

In this literature, Walsh, et al. (2004), Brown, et al. (2004) and Elfving, et al. (2010) are mostly related to our work. In particular, Walsh, et al. (2004) use simulation to determine the proper positions of safety-stock in the supply chain of stainless steel components, independently of any specific project, to reduce and stabilize the random lead time. These papers focus on supply chain operations only without considering the project scheduling issues.

3 The Modeling Framework

In this section, we present the model of Project-Driven Supply Chain.

3.1 Preliminaries

We consider recurrent projects that share the same schedule and the same type of materials which may be subject to random lead times. The starting times of the projects and the amount of each material required are random. The model is depicted in Figure 2 where each project consists of a set of tasks that are conducted sequentially. Each task (except the first one) requires a material which is provided by a supply chain. Specifically, there are $n + 1$ tasks and task i ($i > 0$) requires

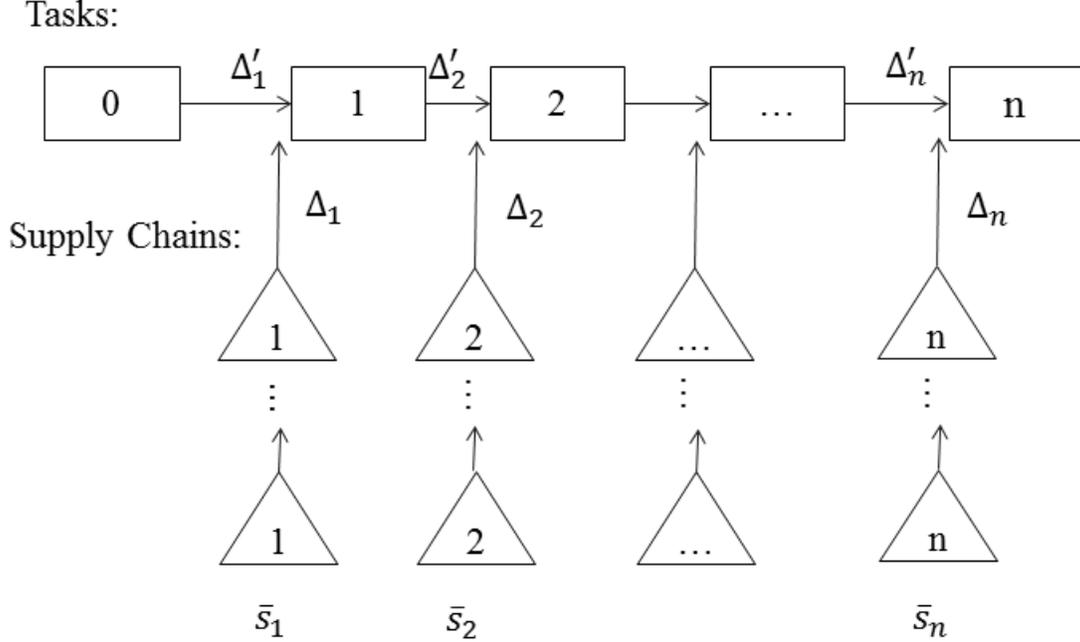


Figure 2: The model of project driven supply chain.

a material which is supplied by supply chain i . Task i can start only after the required material becomes available and task $i - 1$ completes. Orders for all materials are placed as soon as the project starts. All projects share a standard schedule. We point out that the model remains mathematically identical if task i is replaced by a group of tasks.

Assumption 1 *We make the following assumptions on the project and supply chain operations, as well as their interface:*

- **Project operations:** (1) each task has a known duration and a known crashing cost function. (2) The project delay penalty function is known. (3) There is no reward for completing the project earlier than the due date. (4) No task can start prior to the planned starting time by the standard schedule. (5) The standard schedule assumes no crashing for each task and no time buffer between consecutive tasks.
- **Supply chain operations:** (1) every stage in the supply chains operates under a periodic-review base-stock policy. (2) Processing times at all stages of the supply chain are sequential. (3) Unsatisfied demand is fully backordered at each stage. (4) Orders are fulfilled on a first-come-first-serve basis (FCFS) at each stage. (5) For any supply chain, delivery is made as soon as inventory becomes available in all stages except the last one where no early delivery

can be made to projects on-site. (6) The delivery lead time from the last stage of a supply chain to projects on-site is negligible.

- **Interface:** (1) the occurrences of projects are independent, and the amount of each type of material required are independent across projects. (2) All needed materials for one task of a project must be delivered in one set. (3) Material delivery status cannot be updated along with time. (4) Supply chains for materials required at different tasks operate independently. (5) Material inventory cannot be held at projects on-site.

While most assumptions are common sense, a few require more explanation. Project assumption (4) is standard in construction industry as equipment and personnel are typically not available prior to the planned schedule. Project assumption (5) is based on the fact that tasks i ($0 < i \leq n$) are critical, and the observed practice that no time buffer is planned for material delays. Supply chain assumption (5) and Interface assumption (5) are true when the project sites have limited space – often applicable in construction. Interface assumption (2) is true when there are significant economies of scale in transportation. We make Supply chain assumption (6) and Interface assumption (3) for simplicity. Relaxations are discussed at the end of this section.

Given the standard schedule, the project-driven supply chain (see Figure 2) operates as follows: task 0 is always on schedule because it is not subject to random material delays. Once task 0 is completed and material 1 is delivered, task 1 starts immediately. If material 1 is ready before the completion time of task 0, we hold it in inventory and pay a holding cost before delivering it at task 0's completion. Otherwise, material i is delivered immediately once it is ready. Depending on the material delay beyond the standard schedule, task 1 may be expedited (but not earlier than the completion time planned by the standard schedule) and incurs a crashing cost. Similar events take place for subsequent tasks and supply chains. Finally, if task n is delayed beyond the planned due date, a penalty is charged.

3.2 The Mathematical Model

To construct the model, we define the following notation:

- z_i : the duration crashed for task $i = 1, 2, \dots, n$.
- Z_i : an upper bound on z_i .

- D_i : the amount of material i required at task i for a project. Define $\bar{D} = \{D_1, D_2, \dots, D_n\}$. Let d_i (\bar{d}) be the realization of D_i (\bar{D} , respectively).
- \bar{s}_i : the base stock levels in the supply chain i , a vector.
- Δ_i : the time when material i is ready subtracts the planned starting time of task i . Let δ_i be its realization. If $\delta_i > 0$, it refers to the material delay of supply chain i beyond the planned starting time of task i ; otherwise, it indicates how much earlier material i is ready than the planned schedule.
- $\Delta'_i \geq 0$: the delay of task $i - 1$'s completion beyond the planned starting time of task i , $i = 1, 2, \dots, n$. Let δ'_i be the realization of Δ'_i .
- $W_i((\Delta'_i - \Delta_i)^+)$: the inventory holding cost of material i at the last stage of supply chain i if this material is ready before the completion time of task $i - 1$.
- $H_i(\bar{s}_i)$: the annual inventory holding cost in supply chain i excluding $W_i(\cdot)$.
- $C_i(z_i)$: crashing cost function of task i .
- Δ' : project delay. Let δ' be its realization.
- $\Pi(\Delta')$: penalty cost if a project is delayed by Δ' .
- λ : the average number of projects completed in one year.

We construct a multi-stage mathematical model to determine the optimal base-stock levels for all supply chains and the crashing decisions for all tasks of projects. In the first stage, we set the base-stock levels for the supply chains. In subsequent stages, we consider each project and determine the optimal crashing policy for each task as material delays are realized over the current and subsequent tasks. Our objective is to minimize the annual supply chain and project cost.

Specifically, we utilize the flow-unit method (see Axsater 1990, Zipkin 1991, Zhao and Simchi-Levi 2006) by keeping track of a specific project and its material requirement. To identify the optimal crashing policy given a set of inventory decisions, we use a dynamic programming model where we denote $G_i(\delta'_i; \bar{s}_i, \bar{s}_{i+1}, \dots, \bar{s}_n, d_i, d_{i+1}, \dots, d_n)$ to be the minimum expected cost incurred from task i on to the end of the project, including both supply chain and project costs, given the delay from task $i - 1$ being δ'_i , the base-stock levels in supply chains $i, i + 1, \dots, n$ being $\bar{s}_i, \bar{s}_{i+1}, \dots, \bar{s}_n$, and the amount of materials required from stages i to stage n being d_i, d_{i+1}, \dots, d_n .

Consider $i = n$, because the probability distribution of Δ_n depends on \bar{s}_n and d_n ,

$$\begin{aligned} G_n(\delta'_n; \bar{s}_n, d_n) &= E_{\Delta_n}[W_n((\delta'_n - \Delta_n(\bar{s}_n, d_n))^+)] + \\ &E_{\Delta_n}[\min_{0 \leq z_n \leq Z_n} \{C_n(z_n) + \Pi((\max\{\delta'_n, \Delta_n(\bar{s}_n, d_n)\} - z_n)^+)\}], \end{aligned} \quad (1)$$

where the notation E_{Δ_n} indicates that the expectation is taken with respect to Δ_n .

Similarly, for any $i = 2, 3, \dots, n-1$, we have,

$$\begin{aligned} &G_i(\delta'_i; \bar{s}_i, \bar{s}_{i+1}, \dots, \bar{s}_n, d_i, d_{i+1}, \dots, d_n) \\ &= E_{\Delta_i}[W_i((\delta'_i - \Delta_i(\bar{s}_i, d_i))^+)] + \\ &E_{\Delta_i}[\min_{0 \leq z_i \leq Z_i} \{C_i(z_i) + G_{i+1}((\max\{\delta'_i, \Delta_i(\bar{s}_i, d_i)\} - z_i)^+; \bar{s}_{i+1}, \dots, \bar{s}_n, d_{i+1}, \dots, d_n)\}], \end{aligned} \quad (2)$$

and the transition function $\Delta'_{i+1} = (\max\{\delta'_i, \Delta_i\} - z_i)^+$.

For $i = 1$, because task 0 is not subject to any random material delay, $\Delta'_1 = 0$ and we drop Δ'_1 from the notation. Following a similar logic as Eqs. (1)-(2), $G_1(\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n, \bar{d})$ can be written as,

$$\begin{aligned} &G_1(\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n, \bar{d}) \\ &= E_{\Delta_1}[W_1((-\Delta_1(\bar{s}_1, d_1))^+)] + \\ &E_{\Delta_1}[\min_{0 \leq z_1 \leq Z_1} \{C_1(z_1) + G_2((\max\{0, \Delta_1(\bar{s}_1, d_1)\} - z_1)^+; \bar{s}_2, \dots, \bar{s}_n, d_2, \dots, d_n)\}]. \end{aligned} \quad (3)$$

Consider all projects conducted per unit of time, our first-stage optimization problem is,

$$\begin{aligned} &\min_{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n} \sum_{i=1}^n H_i(\bar{s}_i) + \lambda E_{\bar{D}}[G_1(\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n, \bar{D})] \\ &s.t. \quad \bar{s}_i \geq 0, \forall i = 1, 2, \dots, n. \end{aligned} \quad (4)$$

In this model, we first determine the optimal crashing decisions upon each scenario of material delays, then we optimize the inventory decisions accordingly.

4 Convexity and The Optimal Crashing Policy

In this section, we characterize the optimal crashing policy given the inventory decisions.

Assumption 2 *We make the following assumptions:*

1. $C_i(\cdot)$ and $\Pi(\cdot)$ are convex and increasing.

2. $W_i(\cdot)$ is convex and increasing.

The convexity of the crashing cost, $C_i(\cdot)$, of a single task is justified by Nahmias (2004) Chapter 9. The crashing cost of a set of tasks is convex in the time crashed because one would first crash the cheapest tasks. The assumption on the delay penalty $\Pi(\cdot)$ and inventory holding cost $W_i(\cdot)$ includes the commonly seen practice of linear functions as a special case.

The following observations are straightforward, we omit the proof.

Observation 1 $f(x, y) = (\max\{x, x_0\} - y)^+$ is jointly convex in $(x, y) \in \mathbb{R}^2$; if $g(z)$ is convex and increasing in z and $f(x, y)$ is jointly convex in (x, y) , then $g(f(x, y))$ is also jointly convex in (x, y) .

By Observation 1, $(\max\{\delta'_i, \delta_i\} - z_i)^+$ is jointly convex in (δ'_i, z_i) . By Assumption 2 and Observation 1, $\Pi((\max\{\delta'_n, \delta_n\} - z_n)^+)$ is jointly convex in (δ'_n, z_n) . These results are summarized in the following observation.

Observation 2 $(\max\{\delta'_i, \delta_i\} - z_i)^+$ is jointly convex in (δ'_i, z_i) for $1 \leq i \leq n$. $\Pi((\max\{\delta'_n, \delta_n\} - z_n)^+)$ is jointly convex in (δ'_n, z_n) .

Proposition 1 $G_i(\delta'_i; \bar{s}_i, \dots, \bar{s}_n, d_i, \dots, d_n)$ is convex and increasing in δ'_i for $i > 1$.

Proof. We use induction by first considering $i = n$. By Assumption 2, $C_n(z_n)$ is convex in z_n . By Zipkin (2000) Proposition A.3.10 and Observation 2, $\min_{0 \leq z_n \leq Z_n} \{C_n(z_n) + \Pi((\max\{\delta'_n, \delta_n\} - z_n)^+)\}$ is convex in δ'_n . By Eq. (1),

$$\begin{aligned} G_n(\delta'_n; \bar{s}_n, d_n) &= E_{\Delta_n}[W_n((\delta'_n - \Delta_n(\bar{s}_n, d_n))^+)] + \\ &\quad E_{\Delta_n}[\min_{0 \leq z_n \leq Z_n} \{C_n(z_n) + \Pi((\max\{\delta'_n, \Delta_n(\bar{s}_n, d_n)\} - z_n)^+)\}]. \end{aligned}$$

To show $G_n(\delta'_n; \bar{s}_n, d_n)$ is convex in δ'_n , we only need $W_n((\delta'_n - \delta_n)^+)$ to be convex in δ'_n , which is true by Assumption 2. $G_n(\delta'_n; \bar{s}_n, d_n)$ is also increasing in δ'_n because $W_n((\delta'_n - \delta_n)^+)$ is increasing in δ'_n (Assumption 2), and $\Pi((\max\{\delta'_n, \delta_n\} - z_n)^+)$ is increasing in δ'_n .

Now we consider $1 < i < n$. By the induction assumption, G_{i+1} is convex and increasing in δ'_{i+1} . By Eq. (2),

$$\begin{aligned} &G_i(\delta'_i; \bar{s}_i, \dots, \bar{s}_n, d_i, \dots, d_n) \\ &= E_{\Delta_i}[W_i((\delta'_i - \Delta_i(\bar{s}_i, d_i))^+)] + \\ &\quad E_{\Delta_i}[\min_{0 \leq z_i \leq Z_i} \{C_i(z_i) + G_{i+1}((\max\{\delta'_i, \Delta_i(\bar{s}_i, d_i)\} - z_i)^+; \bar{s}_{i+1}, \dots, \bar{s}_n, d_{i+1}, \dots, d_n)\}]. \end{aligned}$$

By Observations 1-2 and the induction assumption, $G_{i+1}((\max\{\delta'_i, \delta_i\} - z_i)^+; \bar{s}_{i+1}, \dots, \bar{s}_n, d_{i+1}, \dots, d_n)$ is increasing in δ'_i and jointly convex in δ'_i and z_i . Following the same logic as for $i = n$, we can show that $G_i(\delta'_i; \bar{s}_i, \dots, \bar{s}_n, d_i, \dots, d_n)$ is increasing and convex in δ'_i . The proof is completed. \square

Theorem 1 *There exists a threshold level, z_i^* (depending on δ'_i and δ_i), for task i ($0 < i \leq n$), such that the optimal crashing policy for task i , is to crash its duration by z_i^* if feasible, otherwise not crash if $z_i^* < 0$ or crash Z_i if $z_i^* > Z_i$.*

Proof. We first consider $i = 1$. By Eq. (3),

$$G_1(\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n, \bar{d}) = E_{\Delta_1}[W_1((-\Delta_1(\bar{s}_1, d_1))^+)] + E_{\Delta_1}[\min_{0 \leq z_1 \leq Z_1} \{C_1(z_1) + G_2((\max\{0, \Delta_1(\bar{s}_1, d_1)\} - z_1)^+; \bar{s}_2, \dots, \bar{s}_n, d_2, \dots, d_n)\}].$$

By Assumption 2, Proposition 1 and Observation 1, $C_1(z_1) + G_2((\max\{0, \delta_1\} - z_1)^+; \bar{s}_2, \dots, \bar{s}_n, d_2, \dots, d_n)$ is convex in z_1 . Let its global minimum be achieved at z_1^* , thus it is optimal to crash task 1 by z_1^* if $0 \leq z_1^* \leq Z_1$, otherwise not crash if $z_1^* < 0$ or crash Z_1 if $z_1^* > Z_1$.

For $1 < i < n$, it follows by Eq. (2) and the above logic that there exists a z_i^* (depending on δ'_i, δ_i) achieving the global minimum for $C_i(z_i) + G_{i+1}((\max\{\delta'_i, \delta_i\} - z_i)^+; \bar{s}_{i+1}, \dots, \bar{s}_n, d_{i+1}, \dots, d_n)$, and the optimal policy is to crash task i by z_i^* if feasible, otherwise not crash if $z_i^* < 0$ or crash Z_i if $z_i^* > Z_i$. The same logic applies to $i = n$. \square

We now discuss some relaxations of Assumption 1. Project assumption (5) can be relaxed by allowing time buffers between consecutive critical tasks. To accommodate this generality, we must introduce additional notation on the starting times of the standard and crashed schedule rather than relying only on their difference. Supply chain assumption (6) can be relaxed without changing the model and results by properly adding a constant to Δ_n . We can relax Interface assumption (3) by assuming that material delivery status can be updated as project processes. While Proposition 1 and Theorem 1 remain true, the optimal crashing policy of a task shall depend on the material delivery status for all subsequent tasks.

Finally, we should point out that the cost function $\sum_{i=1}^n H_i(\bar{s}_i) + \lambda E_{\bar{D}}[G_1(\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n, \bar{D})]$ is generally not convex in the base-stock levels. An example is given in §6.

5 Interaction of Supply Chain and Project Decisions

In this section, we study the interaction between supply chain inventory decisions and project crashing and delay decisions. Our key result is that if we reduce the inventory levels for material k which feeds task k and thus increase the delay of this material, Δ_k , with probability one (w.p.1), then the cumulative time crashed for all tasks preceding task k will reduce w.p.1 to match the delayed material k ; while for each task succeeding task k , the time crashed will increase w.p.1 to catch up with the delayed schedule. The project delay will also increase w.p.1. More specifically, we derive the following results:

- By increasing the delay of material k (feeding task k) w.p.1, we show that the cumulative time crashed for all tasks prior to task k decreases w.p.1. (Proposition 2). We prove it by first showing that the right derivatives of the optimal cost functions for all tasks prior to $k+1$ decrease for each cumulative delay (Lemma 1), and then we show that the cumulative delays for all tasks prior to k increase w.p.1 (Lemma 2).
- By increasing the delay of material k (feeding task k) w.p.1, we also show that the time crashed for task k and all the subsequent tasks increases w.p.1 (Proposition 3). We prove it by first showing that the time crashed at task k increases w.p.1 (Lemma 3), and then we show that the cumulative delays for task k and all subsequent tasks as well as the project delay increase w.p.1 (Lemma 4).

For the ease of exposition, we drop the notation of demand and stock levels the following analysis. We consider the right derivatives in the following proofs.

Impact On Tasks Preceding Task k

We first study the impact of a stochastically greater material k delay, Δ_k , on the optimal cost functions.

Lemma 1 *For any $j \leq k$, $G'_j(\delta'_j)$ decreases for each δ'_j as Δ_k increases w.p.1.*

Proof. When $j = k$, $G_k(\delta'_k) = E_{\Delta_k}[W_k((\delta'_k - \Delta_k)^+) + \min_{0 \leq z_k \leq Z_k}[C_k(z_k) + G_{k+1}((\max(\delta'_k, \Delta_k) - z_k)^+)]]$. Because it is not optimal to set $z_k > \max(\delta'_k, \Delta_k)$, we remove $()^+$ from G_{k+1} for simplicity in the rest of this section. The results still hold without this simplification.

Suppose Δ_k increases to $\Delta_k + E$, where E is a non-negative random variable and its realization is ϵ , let $\tilde{G}_k(\delta'_k) = E_{\Delta_k + E}[W_k((\delta'_k - \Delta_k - E)^+) + \min_{0 \leq z_k \leq Z_k}[C_k(z_k) + G_{k+1}(\max(\delta'_k, \Delta_k + E) - z_k)]]$. Consider a sample path of $\Delta_k = \delta_k$ and $E = \epsilon$, we have three cases:

1. $\delta'_k < \delta_k$. $\tilde{g}_k(\delta'_k) = \min_{0 \leq z_k \leq Z_k}[C_k(z_k) + G_{k+1}(\delta_k + \epsilon - z_k)]$ and $g_k(\delta'_k) = \min_{0 \leq z_k \leq Z_k}[C_k(z_k) + G_{k+1}(\delta_k - z_k)]$. Since neither of them is a function of δ'_k , $\tilde{g}'_k(\delta'_k) = g'_k(\delta'_k) = 0$.
2. $\delta_k \leq \delta'_k < \delta_k + \epsilon$. $\tilde{g}_k(\delta'_k) = \min_{0 \leq z_k \leq Z_k}[C_k(z_k) + G_{k+1}(\delta_k + \epsilon - z_k)]$ and $g_k(\delta'_k) = W_k(\delta'_k - \delta_k) + \min_{0 \leq z_k \leq Z_k}[C_k(z_k) + G_{k+1}(\delta'_k - z_k)]$. Here $\tilde{g}'_k(\delta'_k) = 0$ and $g'_k(\delta'_k) \geq 0$ ($g_k(\delta'_k)$ is convex increasing in δ'_k by Proposition 1), so $\tilde{g}'_k(\delta'_k) \leq g'_k(\delta'_k)$.
3. $\delta'_k \geq \delta_k + \epsilon$. $\tilde{g}_k(\delta'_k) = W_k(\delta'_k - \delta_k - \epsilon) + \min_{0 \leq z_k \leq Z_k}[C_k(z_k) + G_{k+1}(\delta'_k - z_k)]$. Because $W'_k(\delta'_k - \delta_k - \epsilon) \leq W'_k(\delta'_k - \delta_k)$ for each δ'_k ($W_k(\cdot)$ is convex increasing by Assumption 2) and the second terms of $\tilde{g}_k(\delta'_k)$ and $g_k(\delta'_k)$ are the same, so $\tilde{g}'_k(\delta'_k) \leq g'_k(\delta'_k)$.

By Assumption 2, it is easy to see that for any δ'_k , the right derivatives, $g'_k(\delta'_k)$ and $\tilde{g}'_k(\delta'_k)$, exist for all sample paths; furthermore, the right derivatives, $g'_k(\delta'_k)$ and $\tilde{g}'_k(\delta'_k)$, are bounded from above. By Rubinstein and Shapiro (1993), Lemma A2, p. 70, the sample path derivatives are unbiased. Thus, $G'_k(\delta'_k)$ decreases for each δ'_k when Δ_k increases w.p.1.

To prove the same result for task $j < k$, we use induction by making the following induction assumption: for a given j ($j < k$), $G'_j(\delta'_j)$ decrease for each δ'_j when Δ_k increases w.p.1. Then, we consider $G'_{j-1}(\delta'_{j-1})$, where

$$G_{j-1}(\delta'_{j-1}) = E_{\Delta_{j-1}}[W_{j-1}((\delta'_{j-1} - \Delta_{j-1})^+) + \min_{0 \leq z_{j-1} \leq Z_{j-1}}[C_{j-1}(z_{j-1}) + G_j(\max(\delta'_{j-1}, \Delta_{j-1}) - z_{j-1})]].$$

Consider a sample path of $\Delta_{j-1} = \delta_{j-1}$, we first note that $W_{j-1}((\delta'_{j-1} - \delta_{j-1})^+)$ does not depend on Δ_k , so we omit it in the following analysis. For the second part of G_{j-1} , we replace z_{j-1} by $\max(\delta'_{j-1}, \delta_{j-1}) - \delta'_j$ and ignore the boundary of z_{j-1} for the ease of exposition (the same result holds with the boundaries), we arrive at $\min_{\delta'_j}[C_{j-1}(\max(\delta'_{j-1}, \delta_{j-1}) - \delta'_j) + G_j(\delta'_j)]$. Let δ_j^{*} be the δ'_j that minimizes this expression, δ_j^{*} should satisfy the first order condition (FOC):

$$C'_{j-1}(\max(\delta'_{j-1}, \delta_{j-1}) - \delta_j^{*}) = G'_j(\delta_j^{*}).$$

Note that $C_{j-1}(\cdot)$ and $G_j(\cdot)$ are both increasing convex functions (by Assumption 2 and Proposition 1) and $G'_j(\delta'_j)$ decreases for each δ'_j when Δ_k increases w.p.1 by the induction assumption, we conclude that the optimal δ_j^{*} increases when Δ_k increases w.p.1.

We now take the derivative of $C_{j-1}(\max(\delta'_{j-1}, \delta_{j-1}) - \delta'_j) + G_j(\delta'_j)$ with respect to δ'_{j-1} and use the FOC for δ'_j to arrive at,

$$d(C_{j-1}(\max(\delta'_{j-1}, \delta_{j-1}) - \delta'_j) + G_j(\delta'_j))/d\delta'_{j-1} = \begin{cases} 0, & \text{if } \delta'_{j-1} < \delta_{j-1} \\ C'_{j-1}(\delta'_{j-1} - \delta'_j), & \text{otherwise.} \end{cases}$$

Because the optimal δ'_j increases when Δ_k increases w.p.1, $C'_{j-1}(\delta'_{j-1} - \delta'_j)$ decreases for each δ'_{j-1} when Δ_k increases w.p.1. Consequently, $g'_{j-1}(\delta'_{j-1})$ decreases for each δ'_{j-1} when Δ_k increases w.p.1. Using a similar analysis and by Rubinstein and Shapiro (1993), Lemma A2, p. 70, we can show that the sample path derivatives are unbiased. Thus, we conclude that $G'_{j-1}(\delta'_{j-1})$ decreases for each δ'_{j-1} when Δ_k increases w.p.1. By induction, we have proven that $G'_j(\delta'_j)$ decreases for each δ'_j for all $j \leq k$ when Δ_k increases w.p.1. \square

We then study the impact of a stochastically greater Δ_k on the cumulative delays for the tasks preceding task k . For the ease of exposition, we ignore the boundaries on z_k in the following analysis. The results stay the same if we include these boundaries.

Lemma 2 Δ'_j increases w.p.1 for any $j \leq k$ if Δ_k increases w.p.1.

Proof. For task 1, we consider a sample path of $\Delta_1 = \delta_1$. We must have $\min_{\delta'_2} [C_1(\max(\delta'_1, \delta_1) - \delta'_2) + G_2(\delta'_2)]$. By definition, $\delta'_1 = 0$. By Lemma 1, $G'_2(\delta'_2)$ decreases for each δ'_2 as Δ_k increases w.p.1, thus it follows from the first order condition (FOC) that the optimal δ'_2 increases (hence, Δ'_2 increases w.p.1) as Δ_k increases w.p.1 because $C_1(\max(\delta'_1, \delta_1) - \delta'_2)$ is convex decreasing in δ'_2 and $G_2(\delta'_2)$ is convex increasing in δ'_2 .

To prove the same result for tasks $1 < j \leq k$, we use induction and make the following induction assumption: Δ'_j ($j < k$) increases w.p.1 as Δ_k increases w.p.1. Consider task j and a sample path of $\Delta'_j = \delta'_j$ and $\Delta_j = \delta_j$, we must have $\min_{\delta'_{j+1}} [C_j(\max(\delta'_j, \delta_j) - \delta'_{j+1}) + G_{j+1}(\delta'_{j+1})]$. By Lemma 1, $G'_{j+1}(\delta'_{j+1})$ decreases for each δ'_{j+1} as Δ_k increases w.p.1. Meanwhile, $C'_j(\max(\delta'_j, \delta_j) - \delta'_{j+1})$ decreases in δ'_{j+1} . By the induction assumption, $\max(\delta'_j, \delta_j)$ increases as Δ_k increases w.p.1, and so $C'_j(\max(\delta'_j, \delta_j) - \delta'_{j+1})$ increases for each δ'_{j+1} as Δ_k increases w.p.1. It follows from the FOC that the optimal solution δ'_{j+1} also increases as Δ_k increases w.p.1. This concludes the induction and yields the desired result. \square

We are now ready to study the impact of a stochastically greater Δ_k on project crashing decisions.

Proposition 2 $z_1^* + z_2^* + \dots + z_{k-1}^*$ decreases w.p.1 if Δ_k increases w.p.1.

Proof. We consider a sample path of $\Delta_i = \delta_i$ for all $i \leq k$. By definition, we have

$$\delta_1 - z_1^* + (\delta_2 - \delta_2^*)^+ - z_2^* + (\delta_3 - \delta_3^*)^+ - \dots - z_{k-1}^* + (\delta_k - \delta_k^*)^+ = \max(\delta_k', \delta_k).$$

Rewrite the equation,

$$\begin{aligned} z_1^* + \dots + z_{k-1}^* &= \delta_1 + (\delta_2 - \delta_2^*)^+ + \dots + (\delta_k - \delta_k^*)^+ - \max(\delta_k', \delta_k) \\ &= \delta_1 + (\delta_2 - \delta_2^*)^+ + \dots + (\delta_{k-1} - \delta_{k-1}^*)^+ - \delta_k^*. \end{aligned}$$

Because $\delta_2^*, \dots, \delta_k^*$ increase as Δ_k increases w.p.1, we conclude, from above equation, $z_1^* + z_2^* + \dots + z_{k-1}^*$ decrease for any sample path, and thus decrease w.p.1 as Δ_k increases w.p.1. \square

Impact On Task k and Succeeding Tasks

We first focus on the time crashed at task k .

Lemma 3 z_k^* increases w.p.1 if Δ_k increases w.p.1.

Proof. We consider a sample path of $\Delta_i = \delta_i$ for all $i \leq k$. At task k , the optimization problem is $\min_{z_k} [C_k(z_k) + G_{k+1}(\max(\delta_k', \delta_k) - z_k)]$, and the FOC is,

$$C_k'(z_k) = G_{k+1}'(\max(\delta_k', \delta_k) - z_k).$$

Because $C_k(z_k)$ is a convex increasing function on z_k , $C_k'(z_k)$ is positive and increasing on z_k . We also note that because $G_{k+1}(\cdot)$ is a convex increasing function, then $G_{k+1}'(\max(\delta_k', \delta_k) - z_k)$ is positive and decreasing on z_k . By Lemma 2, it follows by the assumption that δ_k increases, $\max(\delta_k', \delta_k)$ shall increase as Δ_k increases w.p.1. As a result, $G_{k+1}'(\max(\delta_k', \delta_k) - z_k)$ increases for each z_k as Δ_k increases w.p.1, and thus the optimal z_k^* increases for any sample path as Δ_k increases w.p.1. \square

We then study the cumulative delays of task k and all subsequent tasks as well as the project delay.

Lemma 4 Δ_j^* (for any $j > k$) and the project delay Δ^* increase w.p.1 if Δ_k increases w.p.1.

Proof. For task k , we consider a sample path of $\Delta_i = \delta_i$ for all $i \leq k$. The optimization problem is $\min_{\delta_{k+1}'} [C_k(\max(\delta_k', \delta_k) - \delta_{k+1}') + G_{k+1}(\delta_{k+1}')]$. We note that $C_k(\max(\delta_k', \delta_k) - \delta_{k+1}')$ is a convex decreasing function in δ_{k+1}' and $G_{k+1}(\delta_{k+1}')$ is a convex increasing function in δ_{k+1}' . By Lemma 2, $\max(\delta_k', \delta_k)$ increases as δ_k increases, consequently, the optimal δ_{k+1}' increases when δ_k increases.

For task $j > k$, we use induction and assume that, for $j > k + 1$, Δ_j^* increase if Δ_k increases. Using a sample path of $\Delta_i = \delta_i$ for all $i \leq j$, we consider the optimization problem at task j , $\min_{\delta'_{j+1}} [C_j(\max(\delta_j^*, \delta_j) - \delta'_{j+1}) + G_{j+1}(\delta'_{j+1})]$. We first note that $C_j(\max(\delta_j^*, \delta_j) - \delta'_{j+1})$ is a convex decreasing function in δ'_{j+1} and $G_{j+1}(\delta'_{j+1})$ is a convex increasing function in δ'_{j+1} . By the induction assumption, $\max(\delta_j^*, \delta_j)$ increases as δ_k increases, consequently, the optimal δ'_{j+1} increases when δ_k increases. This concludes the induction for tasks $j > k$.

The project delay $\Delta' = \Delta'_{n+1}$ if we extend the definition of Δ'_j to $n + 1$. Because the penalty function is also a convex increasing function in Δ' , by a similar proof, we can show that the optimal project delay increases w.p.1 when Δ_k increases w.p.1. \square

Finally, we study the impact of a stochastically greater Δ_k on the crashing decisions.

Proposition 3 z_j^* increases w.p.1 for all $j > k$ if Δ_k increases w.p.1.

Proof. For task j ($j > k$), we consider a sample path of $\Delta_i = \delta_i$ for all $i \leq j$ and the optimization problem

$$\min_{z_j} [C_j(z_j) + G_{j+1}(\max(\delta_j^*, \delta_j) - z_j)].$$

We note that $C_j(z_j)$ is convex increasing in z_j and $G_{j+1}(\max(\delta_j^*, \delta_j) - z_j)$ is convex decreasing in z_j . By Lemma 4, $\max(\delta_j^*, \delta_j)$ increases when δ_k increases, and $G_{j+1}(\max(\delta_j^*, \delta_j) - z_j)$ increases for each z_j as $\max(\delta_j^*, \delta_j)$ increases. As a result, the optimal z_j^* increases if δ_k increases. Thus, z_j^* increases w.p.1 for all $j \geq k$ if Δ_k increases w.p.1. \square

6 An Illustrating Example

In this section, we apply the PDSC model and analysis to a real-world example, ICM, to demonstrate its potential.

6.1 ICM Overview

ICM is a U.S. based construction management firm that keeps internally only design, engineering, bidding, project planning and management functions. The company follows a standard bidding process for each project, and the outcome of which is not predictable. Once a project is awarded, ICM assigns it to a project manager who puts the plan into action by securing subcontractors (for labor) and ordering materials from suppliers. The project manager oversees the entire project

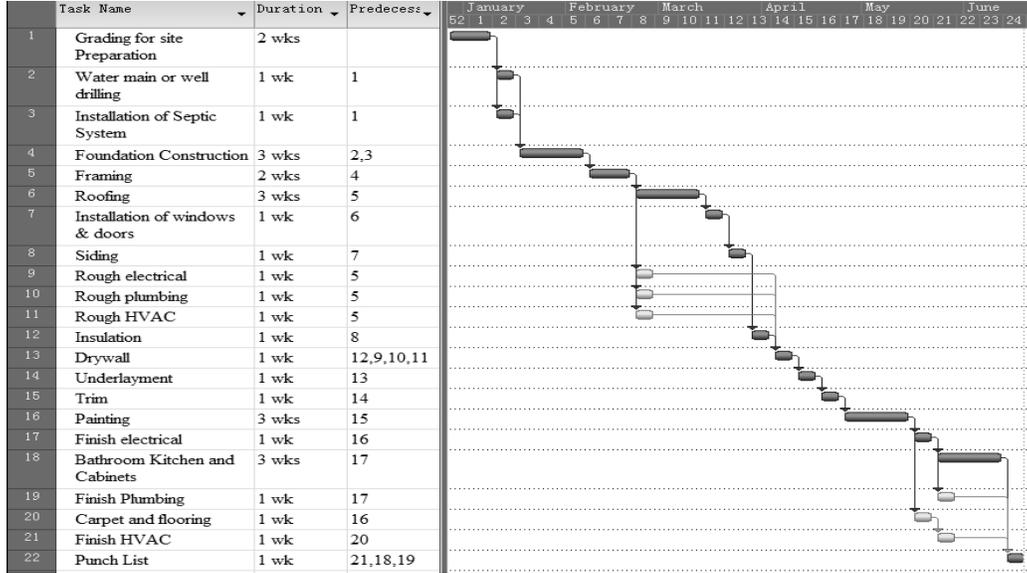


Figure 3: Standard project schedule at ICM.

execution and is not connected in any formal way to other project managers. We refer the reader to Shah and Zhao (2009) for a detailed case study.

Project operations. All construction projects follow a standard schedule (see Figure 3) where the tasks in darker color are on the critical path. The total duration of a project is 25 weeks. The structural steel is needed for Task 5 (Framing) at the beginning of the 7th week. One project only requires one set of structural steel at this point in time. If this material is delayed, the tasks that can be crashed (expedited) to bring the schedule back on track are 5 (Framing), 6 (Roofing), 16 (Painting), and 18 (Bathroom, Kitchen and Cabinets). Crashing a task reduces its duration but must maintain the total labor hours. Crashing cost comes from the over-time wage which is 50% higher than the regular wage. The delay penalty per week for a project is 1% of the project revenue (less materials).

Structural steel supply chain. Structural steel is the most expensive material and is used in all projects. The structural steel supply chain consists of three stages: producer, service center and fabricator. The producer manufactures standardized shapes. The service center serves as a warehouse before fabricator. The fabricator customizes the structural steel according to engineering drawings. We refer to Figure 4 for a detailed description. These companies are the only government authorized suppliers in proximity and thus cannot be switched. Currently, the entire structural steel supply chain makes to order and the total lead time ranges from 5 to 8 weeks.

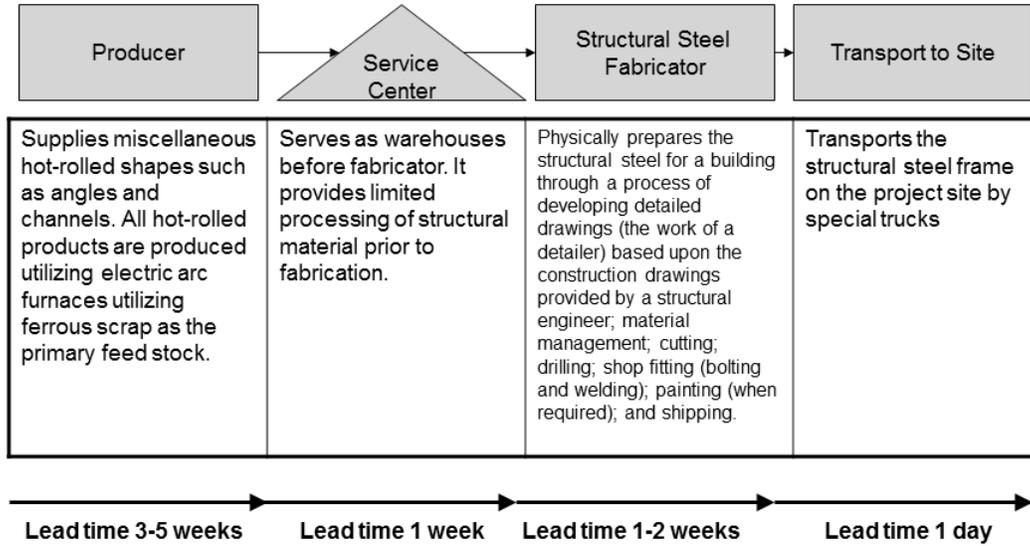


Figure 4: The structural steel supply chain.

Because projects may start at different times and require different amount of structural steel, ICM's weekly material requirement is highly sporadic and possibly zero if no projects require framing at a certain week.

6.2 PDSC Model for ICM

Assumption 1 holds for ICM. In addition, we need,

Assumption 3 For ICM's projects and its structural steel supply chain, we assume

- *Project operations: (1) There is at most one project requiring structural steel at any given period. (2) Project delay penalty is a linear function of the period delayed.*
- *Supply chain operations: (1) Safety-stock can only be held at the service center in the form of standardized shapes; (2) All stages in the structural steel supply chain share an identical review period – one week.*

The assumption on projects is justified by the real-world data and the fact that ICM receives about 20 projects a year. Safety-stock cannot be held at the fabricator because customized steel is project-specific and thus very risky to hold before the project is awarded.

For the structural steel supply chain, we assume the standard sequence of events at each stage (see, e.g., Hadley and Whitin 1963): At the beginning of a period, the stage receives replenishment,

then reviews inventory and places orders if appropriate. At the end of the period, all demand is realized, deliveries are made, and all costs are calculated for the period.

The PDSC model of §3 can be applied to ICM as follows: we group all tasks prior to Task 5 (where the structural steel is required) as task 0. We also group Task 5 and all subsequent tasks as task 1. The structural steel supply chain is denoted as supply chain 1. We define the following additional notation for the structural steel supply chain:

- D_1^t, d_1^t : the demand for structural steel at the t^{th} week, and its realization. Note that each project only requires one set of structural steel at one point in time and thus this demand process is formed by material requirement of consecutive projects. By the same nature of lead time demand, we define $D_1[t, t+k] = \sum_{i=0}^k D_1^{t+i}$ for $k \geq 0$. If $k < 0$, then $D_1[t, t+k] = 0$. We also define $D_1(l)$ be demand during l periods of time.
- L : the total lead time of the service center that includes the processing times at the producer and the service center.
- S : the base-stock level at the service center.
- X : the service time provided by the service center.
- Y : the processing time at the fabricator.
- h_s, h_f : The annual inventory holding cost per unit at the service center and fabricator respectively.
- T : the due date – the time when structural steel is needed for a project since the beginning of the project.

By definition,

$$\Delta_1 = X + Y - T. \quad (5)$$

By Eq. (3), $G_1(\cdot)$ for ICM is

$$G_1(S, d_1) = E_{\Delta_1}[W_1((-\Delta_1(S, d_1))^+)] + E_{\Delta_1}[\min_{0 \leq z_1 \leq Z_1} \{C_1(z_1) + \pi(\max\{0, \Delta_1(S, d_1)\} - z_1)^+\}], \quad (6)$$

where the first term is the inventory holding cost at the Fabricator for materials that are ready before they are needed at a project.

For ICM, the first-stage optimization problem is,

$$\begin{aligned} \min_S \quad & H_1(S) + \lambda E_{D_1}[G_1(S, D_1)] \\ \text{s.t.} \quad & S \geq 0, \end{aligned} \quad (7)$$

where $H_1(S)$ is the annual inventory holding cost at the service center.

We now characterize how the base-stock level, S , determines the distribution of the service time (X) at the service center, the random material delay (Δ_1) to projects, and the cost of the supply chain. Define $R_k(d_1^t)$ to be the probability that demand d_1^t (> 0) is satisfied within the period $t + k$ at the service center. For a constant lead time L , $R_k(d_1^t) = \Pr\{S - D_1[t - L + k, t] \geq 0 | d_1^t > 0\}$ for $k \leq L$ (Hausman, et al. 1998). For stochastic and sequential lead time L (see Zipkin (2000) Chapter 7), we let the probability of $L = L_i$ be $\Pr\{L = L_i\}$, then $R_k(d_1^t) = \sum_i \Pr\{S - D_1[t - L_i + k, t] \geq 0 | d_1^t > 0\} \Pr\{L = L_i\}$ for $0 \leq k \leq \max_i\{L_i\}$. Let $X(d_1^t)$ be the service time of d_1^t (> 0), then $R_0(d_1^t) = \Pr\{X(d_1^t) = 0\}$ and $R_k(d_1^t) = \Pr\{X(d_1^t) \leq k\}$ for $k > 0$. Consequently, $\Pr\{X(d_1^t) = 0\} = R_0(d_1^t)$ and $\Pr\{X(d_1^t) = k\} = R_k(d_1^t) - R_{k-1}(d_1^t)$ for $k > 0$.

With the distribution of $X(d_1^t)$ (for $d_1^t > 0$), we can calculate the distribution of Δ_1 by Eq. (5). The expected inventory holding cost at the fabricator for a project that requires d units of structural steel can be calculated by,

$$E_{\Delta_1}[W_1((-\Delta_1(d))^+)] = E_{\Delta_1}[W_1((T - X - Y)^+)] = h_f \times d \times E[(T - X(d) - Y)^+]. \quad (8)$$

The net inventory at the end of period t at the service center is $N(t) = S - D_1[t - L, t]$, and the on-hand inventory at the service center is $I_s(t) = N(t)^+$. In steady state, the annual inventory holding cost at the service center is

$$H_1(S) = h_s E[I_s] = h_s E[(S - D_1(L + 1))^+]. \quad (9)$$

6.3 Impact of the PDSC Model

Using historical data, we construct an empirical distribution for the weekly demand of structural steel, D_1^t , as shown in Table 1. Here we use the approximation of discrete demand by rounding up, for instance, all values in $[41, 50]$ to 50.

With $S = 0$ at the service center, the structural steel supply chain can deliver an order to project on-site at the beginning of 7th, 8th, 9th, and 10th week due to the review period and the

Value	Frequency	Probability	Value	Frequency	Probability
0	32	0.604	60	4	0.075
10	0	0	70	1	0.019
20	0	0	80	2	0.038
30	1	0.019	90	0	0
40	7	0.132	100	2	0.038
50	4	0.075	110	0	0

Table 1: The empirical distribution of D_1^t .

sequence of events of the structural steel supply chain. Thus it may cause a delay of 1, 2, or 3 weeks because it is needed at the beginning of 7th week. We assume that the processing time at the producer has an equal chance to be 3, 4, and 5 weeks, and the processing time at the fabricator has an equal chance to be 1 and 2 weeks. The monthly inventory holding cost is \$16.6 per ton, and thus the annual inventory holding cost $h_s = h_f = \$199.2$ per ton per year.

For ICM, the average number of construction projects conducted annually, λ , is 20. $T = 6$ weeks. The crashing cost function is piece-wise linear where,

$$C_1(z_1) - C_1(z_1 - 1) = \begin{cases} \$4,500 & z_1 = 1, \\ \$6,500 & z_1 = 2, 3, 4, 5, \\ \$8,000 & z_1 = 6, 7. \end{cases} \quad (10)$$

The project delay penalty per week, $\pi = \$13,700$. Clearly the delay penalty is much more expensive than the crashing costs and thus should be avoided.

We now compare the PDSC model to the current practice of ICM which holds zero inventory ($S = 0$) in the structural steel supply chain. Figure 5 summarizes the numerical result, where the vertical axis stands for the annual operating costs while the horizontal axis presents the base-stock level at the service center, S . We make the following observations:

- The total cost function is not convex in S . It first increases slightly (due to the bulky material requirement of projects, a small amount of inventory does not reduce lead time but adds to cost) and then decreases as S increases before it reaches the global minimum at $S = 140$. Then it slowly increases as S further increases.
- The components of the total cost behave as we expected in §1: as we increase the base-stock level from $S = 0$, the inventory cost increases slowly but the project crashing cost decreases

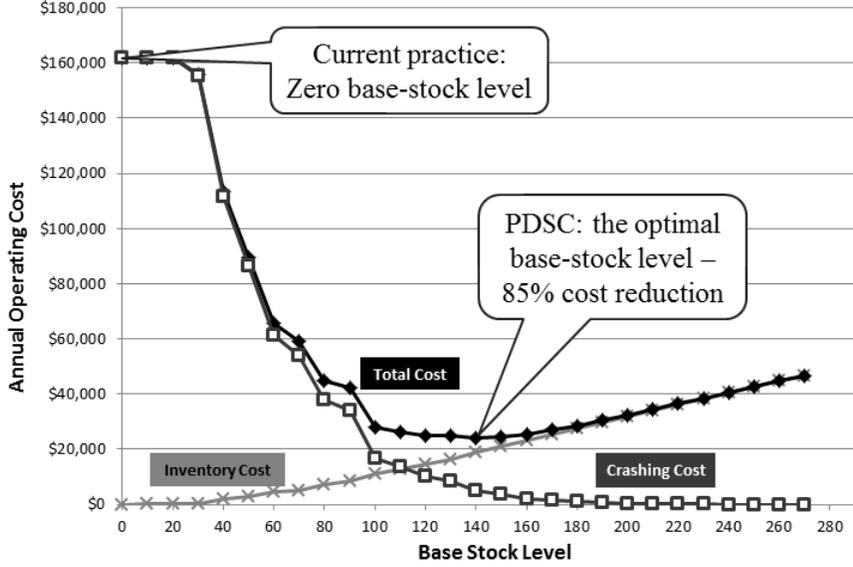


Figure 5: The annual operating costs.

	$P\{\Delta_1 = 0 \text{ week}\}$	1	2	3	$E[\Delta_1]$
$S = 0$	0.167	0.333	0.333	0.167	1.5
$S = 140$	0.954	0.037	0.008	0	0.054

Table 2: Delays of structural steel.

sharply, and thus the total cost decreases sharply. Beyond $S = 140$, the project crashing cost almost reaches zero but the inventory cost becomes significant, and so further increasing the base-stock level becomes un-economical. Comparing to the current practice with $S = 0$ (ignoring material supply in project decisions), the PDSC model (integrated supply chain and project decisions) brings the cost down from \$161,667 to \$24,164, with a 85.1% saving.

- The delay of structural steel, Δ_1 , at the optimal base-stock level, $S = 140$, is much smaller (stochastically) than Δ_1 at $S = 0$, see Table 2. Thus the safety-stock at the service center reduces the random material delays and stabilizes the schedule of the projects, such benefits far outweigh the holding cost of the safety-stock.

7 Numerical Study

In this section, we conduct an extensive numerical study to gauge the potential savings of the PDSC model by solving various environments that companies may face in practice. We use ICM

as a base case and conduct a sensitivity analysis with respect to its parameters, such as demand variability, inventory holding cost, lead time and due date.

Demand Variability and Project Arrival Rate λ . We first study the impact of demand variability on the savings. We define \tilde{D} to be the nominal weekly demand which follows a normal distribution. Since demand cannot be negative, we set the weekly demand distribution to be discrete truncated normal where $P\{D_1(1) = 0\} = P\{\tilde{D} \leq 0\}$, and $P\{D_1(1) = k\} = P\{\tilde{D} \leq k\} - P\{\tilde{D} \leq k - 10\}$ for $k = 10, 20, \dots$. In the numerical study, we set $E[\tilde{D}] = 30$ tons and change the standard deviation of \tilde{D} from 3 tons to 48 tons while keeping everything else the same as in §6.

Consistent to intuition, our numerical results (not reported here) shows that when the demand variability increases, more inventory is needed to buffer against demand uncertainty and to reduce material delays down to the same level. Thus, the optimal base stock level and the total cost increase. Given the cost of the benchmark, $S = 0$, remains unchanged, the percentage saving decreases as demand variability increases.

We also studied the impact of project arrival rate, λ , on the percentage savings. Here we set $E[\tilde{D}] = 30$ tons and the standard deviation of \tilde{D} to be 15 tons but vary λ from 10 to 50. Our numerical study (not reported here) shows that the percentage savings increase as λ increases. Intuitively, it is more beneficial to integrate inventory and project decisions if projects occur more frequently.

Inventory Cost. Second, we study the impact of inventory holding cost on the savings from the PDSC model. We set the coefficient of variation of the nominal demand to be 0.8 and change the inventory holding cost per ton per week (at the service center and fabricator) according to $\$3.82 \times$ an inventory cost factor which varies from 0.6 to 1.4. The result (not reported here) shows that when holding inventory becomes more expensive, the percentage savings decrease. The result is intuitive because more expensive inventory cost means less valuable the option of holding inventory, and so the smaller the percentage saving from integrating safety-stock and crashing decisions.

We also study the sensitivity of the optimal base-stock level with respect to the inventory holding cost. To test the robustness of the solution, we compare the cost of a benchmark base-stock level, $S = 150$, to the optimal cost at various inventory holding costs for which this benchmark base-stock level is not optimal. Table 3 shows that when the holding cost increases, the optimal base-stock level tends to decrease but not by much. Even if the optimal base-stock level differs

Inventory cost factor	Optimal base-stock level	% Cost increment at $S = 150$ over the optimal cost
1.4	140	2.99%
1.3	140	2.39%
1.2	140	1.7%
1.1	140	0.91%
1	150	0
0.9	150	0
0.8	150	0
0.7	150	0
0.6	160	1.34%

Table 3: Robustness of the optimal base-stock level.

Lead Time Range	(3,5)	(4,6)	(5,7)	(6,8)	(7,9)
% Cost Savings	85.6%	89.1%	93.2%	94.9%	93.7%

Table 4: Impact of the lead time.

from the benchmark, the cost of benchmark solution is only slightly higher than the optimal cost. In summary, these studies show that inventory holding cost may have a significant impact on the percentage savings of the PDSC model but much less an impact on the optimal base-stock level.

Lead Time. To study the impact of lead time on the effectiveness of the PDSC model, we assume that the lead time for the service center follows a uniform distribution with different ranges, see Table 4. We set the coefficient of variance of the nominal demand to be 0.8, inventory holding cost per ton per week to be \$3.82, and everything else remains unchanged.

The impact of lead time is more complex than that of demand variability and inventory holding cost because the latter only affects the supply chain inventory cost but the former affects both the project crashing and supply chain inventory costs. Table 4 shows that as the lead time increases, the % saving tends to increase but not always. This is true because as lead time increases, the project crashing cost may increase significantly in all solutions, and thus lead to smaller percentage savings.

Due Date. Finally, we study the impact of the project due date, T , on the effectiveness of

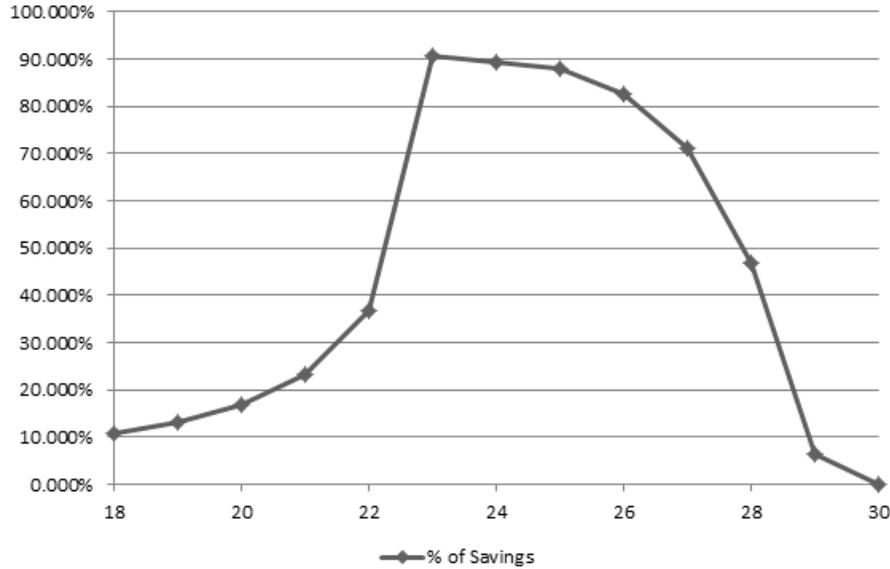


Figure 6: The impact of project due date.

the PDSC model. For this purpose, we set the lead time for the service center to be uniformly distributed from 3 to 8 weeks, the coefficient of variance of the nominal demand to be 0.8, and inventory holding cost per ton per week to be \$3.82. We choose a wider ranged lead time than previous studies because it allows us to study a wider range of the due date.

In the standard schedule of ICM, task 0 (task 1, see definition in §6.2) has a duration of 6 (19) weeks. So the project duration is 25 weeks. Because task 0 cannot be expedited and task 1 can be expedited by seven weeks at most, the shortest duration of the project is 18 weeks without considering material delays. Considering the worst case of material delay and no expediting of any task, the longest duration of the project is 30 weeks.

Figure 6 illustrates the following insight,

- When the project due date is very tight (approaching the shortest duration), the saving of the PDSC model diminishes because the high penalty cost and the tight due date force us to crash all tasks to their minimum in all solution approaches. In this case, safety-stock only helps to balance the project delay penalty and inventory holding cost.
- When the project due date is very loose (approaching the longest duration), the saving again diminishes. This is true because there is plenty of time to accommodate material delays, and thus it is not necessary to consider material supply while planning for projects.
- When the due date is in between the shortest and longest durations, the savings can be quite

significant because one has the flexibility to coordinate safety-stock and crashing decisions so as to balance the inventory and project costs. Specifically, the percentage saving reaches its peak value at 23 weeks. We note that if the due date is less than 23 weeks, the percentage saving is low because the total cost is high due to project delay penalty paid in some events. If the due date is greater than 23 weeks, we can manage not to pay delay penalty in any event, which reduces the total cost and increases the percentage savings.

In summary, the saving of integrating safety-stock and crashing decisions reaches its peak value when the due date is moderate, which is most likely the case in practice.

8 Conclusion

In this paper, we study a new class of problems – recurrent projects with random material delays, at the interface between supply chain and project management. We present the model of project-driven supply chains to jointly optimize safety-stock and project crashing decisions. We prove certain convexity properties for the model which facilitate fast computation of the optimal crashing decisions. We also study the dependence of the project crashing decisions on the supply chain inventory decisions. We demonstrate the impact of the model by a real-world example and study its sensitivity with respect to several system parameters. The model, although ignoring the economies of scale in production as well as material/project customization, captures the trade-off between supply chain operations and project operations for recurrent projects under uncertainty. It demonstrates that a certain amount of material inventory, if placed at the right location(s) of the supply chain, can help stabilizing the schedule of projects and reducing system-wide cost substantially.

One variation of the problem not considered in this paper is recurrent projects with different schedules. For instance, the same material may be required by different types of projects at different tasks. While the general trade-off and idea of the PDSC model still apply, we must modify the model to account for different project types. In this paper, we only test the effectiveness of the model in examples with one critical material. It would be interesting to see the impact of the PDSC model in practice with multiple materials. Finally, the structure of the projects considered in this paper is simplified into a sequence of critical tasks. In reality, projects can be more complex with parallel non-critical paths. The development and application of the PDSC model in these areas

remain as our future research directions.

Acknowledgments

We are grateful for the editor and referees for their constructive comments and suggestions that have helped us improve this paper. This work is supported by grant CMMI No. 0747779 from the National Science Foundation.

References

- [1] Aquilano, N. J., D. E. Smith. 1980. A formal set of algorithms for project scheduling with critical path method-material requirements planning. *Journal of Operations Management*, **1**, 57-67.
- [2] Axsater, S. 1990. Simple solution procedures for a class of two-echelon inventory problems. *Operations Research*, **38**, 64-69.
- [3] Axsater, S. 2006 *Inventory Control*, 2nd Edition. Springer, New York, NY.
- [4] Brown, K., T. G. Schmitt, R. J. Schonberger, S. Dennis. 2004. Quadrant Homes applies lean concepts in a project environment. *Interfaces*, **34**, 442-450.
- [5] Dodin, B., A. A. Eliman. 2001. Integrated project scheduling and material planning with variable activity duration and rewards. *IIE Transactions*, **33**, 1005-1018.
- [6] Elfving, J.A., G. Ballard, U. Talvitie. 2010. Standardizing logistics at the corporate level towards lean logistics in construction. *Proceedings IGLC-18*, July 2010, Technion, Haifa, Israel, page 222-231.
- [7] Graves, S. C., S. P. Willems. 2000. Optimizing strategic safety stock placement in supply chains. *Manufacturing and Service Operations Management*. **2**, 68-83.
- [8] Graves, S. C., S. P. Willems. 2003. Supply chain design: safety stock placement and supply chain configuration, in A. G. de Kok and S. C. Graves, eds, *Handbooks in Operations Research and Management Science Vol. 11, Supply Chain Management: Design, Coordination and Operation*. North-Holland Publishing Company, Amsterdam, The Netherlands.

- [9] Graves, S. C., S. P. Willems. 2005. Optimizing the Supply Chain Configuration for New Products. *Management Science*, **51**, 1165-1180.
- [10] Hadley, G. & T. M. Whitin. 1963. *Analysis of Inventory Systems*. Prentice-Hall, Englewood Cliffs, NJ.
- [11] Hariharan, R., P. Zipkin. 1995. Customer-order information, leadtimes, and inventories. *Management Science*, **41**, 1599-1607.
- [12] Hausman, W. H., H. L. Lee, & A. X. Zhang. 1998. Joint demand fulfillment probability in a multi-item inventory system with independent order-up-to policies. *European Journal of Operational Research*, **109**, 646-659.
- [13] Jozefowska, J., Jan Weglarz. 2006. *Perspectives in Modern Project Scheduling*. Springer; 1st edition.
- [14] Kerwin, K. 2005. BW 50: A new blueprint at Pulte Homes. *BusinessWeek*, October 3rd, 2005.
- [15] Nahmias, S. 2004. *Production and operations analysis*, 5th Edition. MCGraw-Hill Irwin. Boston.
- [16] O'Brien, W. J., K. London, R. Vrijhoef. 2002. Construction supply chain modeling: a research review and interdisciplinary research agenda. *Proceedings IGLC-10*, 1-19. August 2002, Gramado, Brazil.
- [17] Ozdamar, L., G. Ulusoy. 1995. A survey on the resource-constrained project scheduling problem. *IIE Transactions*, **27**, 574-586.
- [18] Pinedo, M. 2005. *Planning and Scheduling in Manufacturing and Services*. Springer New York, NY.
- [19] Porteus, E. L. 2002. *Foundations of Stochastic Inventory Theory*. Stanford University Press, California.
- [20] Rubinstein, R. Y. and A. Shapiro (1993). *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. John Wiley and Sons, New York, NY.

- [21] Schmitt, T. G., B. Faaland. 2004. Scheduling recurrent construction. *Naval Research Logistics*, **51**, 1102-1128.
- [22] Shah, M., Y. Zhao. 2009. Construction Resource Management – ICM Inc. Rutgers Business School Case Study. Newark, NJ.
- [23] Simchi-Levi, D., Y. Zhao. 2011. Performance evaluation of stochastic multi-echelon inventory systems: a survey. *Advances in Operations Research*, Volume 2012, Article ID 126254, 34 pages.
- [24] Smith-Daniels, D. E., N. J. Aquilano. 1984. Constrained resource project scheduling subject to material constraints. *Journal of Operations Management*, **4**, 369-388.
- [25] Smith-Daniels, D. E., V. L. Smith-Daniels. 1987. Optimal project scheduling with materials ordering. *IIE Transactions*, **19**, 122-129.
- [26] Somerville, C. T. 1999. Residential Construction Costs and the Supply of New Housing: Endogeneity and Bias in Construction Cost Indexes. *The Journal of Real Estate Finance and Economics*, **18**, 43-62.
- [27] Tommelein, I. D., G. Ballard, P. Kaminsky. 2009. Supply chain management for lean project delivery. Chapter 6 in *Construction Supply Chain Management Handbook*, eds. W. J. O'Brien, C. T. Formoso, R. Vrijhoef, K. A. London. CRC Press – Taylor & Francis Group, New York.
- [28] Tommelein, I. D., K. D. Walsh, J. C. Hershauer. 2003. Improving Capital Projects Supply Chain Performance. Research Report No. 172-11, Construction Industry Institute, Austin, TX, 241 pp.
- [29] Vaidyanathan, K., G. Howell. 2007. Construction supply chain maturity model – conceptual framework. *Proceedings IGLC-15*, 170-180. July 2007, Michigan, USA.
- [30] Vrijhoef, R., L. Koskela. 2000. Roles of supply chain management in construction. *European Journal of Purchasing and Supply Chain Management*. **6**, 169-178.
- [31] Walsh, K. D., J. C. Hershauer, I. D. Tommelein, T. A. Walsh. 2004. Strategic positioning of inventory to match demand in a capital projects supply chain. *Journal of Construction Engineering and Management*, ASCE / November/December, 818-826.

- [32] Xu, X., Y. Zhao. 2010. Some economic facts of the prefabricated housing. Industry Report, Rutgers Business School, Newark, NJ.
- [33] Yeo, K. T., J. H. Ning. 2002. Integrating supply chain and critical chain concepts in engineering-procure-construct (EPC) projects. *International Journal of Project Management*, **20**, 253-262.
- [34] Zhao, Y., D. Simchi-Levi. 2006. Performance analysis and evaluation of assemble-to-order systems with stochastic sequential lead-times. *Operations Research*, **54**, 706-724.
- [35] Zipkin, P. 1991. Evaluation of base-stock policies in multiechelon inventory systems with compound-Poisson demands. *Naval Research Logistics*, **38**, 397-412.
- [36] Zipkin, P. 2000. *Foundations of Inventory Management*. McGraw Hill, Boston.