A Two-stage Stochastic Mixed-integer Program for the Capacitated Logistics Fortification Planning under Accidental Disruptions

Xuwei Qin, X. Liu, Lixin Tang

PII: S0360-8352(13)00121-6
DOI: http://dx.doi.org/10.1016/j.cie.2013.04.006
Reference: CAIE 3446

To appear in Computers & Industrial Engineering

Received Date: 11 April 2011
Revised Date: 6 October 2012
Accepted Date: 6 April 2013


This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
A Two-stage Stochastic Mixed-integer Program for the Capacitated Logistics Fortification Planning under Accidental Disruptions

Xuwei Qin\textsuperscript{1,2}, X. Liu\textsuperscript{3*}, Lixin Tang\textsuperscript{4}

1. School of Business Administration, Northeastern University, P.R.China
2. Department of Production and Supply Chain Management, Technical University of Munich, Germany
3. Department of Industry Engineering, Shanghai Jiaotong University, P.R.China
4. School of Information Science and Engineering, Northeastern University, P.R.China

Corresponding author: Professor Xiao LIU
Department of Industrial Engineering management, Shanghai Jiao Tong University, 800 Dongchuan Road, Min-Hang District, Shanghai 200240, P.R.China

Email: x_liu@sjtu.neu.cn
Phone: +86-3420-7185
Fax: +86-3420-7185
A Two-stage Stochastic Mixed-integer Program for the Capacitated Logistics

Fortification Planning under Accidental Disruptions
Abstract

Vulnerability to service disruptions caused by accidents is one of the major threats in existing logistics systems. This paper presents a fortification planning model for capacitated logistics systems in a two-stage stochastic mixed-integer programming framework. Considering limited protection investment budget, the model can deal with locating fortified facilities, pre-positioning emergency inventory and assigning emergency transportation under scenario-based random parameters. The risk mitigation combination of facility protection and emergency inventory pre-positioning policies is proposed to hedge well against accidental disruptions in the capacitated logistics systems. The revised disjunctive decomposition-based branch-and-cut (D2-BAC) algorithm for the model is developed by integrating with two types of valid cuts and dynamical ‘truncation’ strategy of the branch-and-bound tree. Extensive computational results confirm the computational performance of the proposed method and indicate that this model can provide a powerful tool for identifying best possible fortification strategies. It is also demonstrated that the risk mitigation combination can significantly increase the reliability of capacitated logistics systems.

Keywords: fortification model; D2-BAC algorithm; capacitated logistics systems
1. Introduction

As long as there have been logistics systems, there have been disruptions, and no logistics networks are immune to them. Most of existing logistics systems was constructed on the assumption that facilities are always available in the design phase. These carefully constructed systems could be severely ruined due to some facility disruptions and the absence of countermeasures when disruptions do strike. Besides, the popularization of modern concepts such as lean, agile, outsourcing, customized, and global networks further results in these networks becoming more vulnerable to disruptions, because there tends to be very little inventory in more complex system to “buffer” any interruptions in supply. As a result, any disruptions can have a dramatic impact on the entire logistics system (Blackhurst et al. 2005).

Many potential threats in practice can lead to facility disruptions and even breakdown of logistics system, e.g., natural catastrophes such as earthquakes, hurricanes, and floods; accidental disruptions such as equipment failures, supplier discontinuities, plant fire, industrial accidents and so on; intentional disruptions such as terrorist attacks, sabotage, and labour strikes. These disruptions can cause not only serious operational consequences, such as higher transportation costs, order delays, inventory shortages, loss of market shares, and so on, but also extended negative financial effects. Several high-profile disruptions are easy to bring to mind: a strike at two General Motors parts plants in 1998 led to the shutdown of 26 assembly plants and ultimately
prevented the company from building over 500,000 vehicles; an eight-minute fire at a Philips semiconductor plant in 2001 brought one customer, Ericsson, to a virtual standstill, while another, Nokia, weathered the disruption (Sheffi, 2005). Most recently, a disastrous earthquake and the following tsunami halted the production of electronic components such as semiconductors, LCD panels and flash memory chips, the global electronics industry will be expecting a large near-term supply shortage (Clark and Takahashi, 2011).

Many studies of reliable/robust logistics design problems incorporating redundancy and flexibility strategies, reviewed by Klibi et al. (2010a), demonstrate that the impact of facility disruptions can be mitigated by considering the risk of disruptions when designing supply chain networks. However, redesigning entirely an existing system is not always reasonable given the prohibitive expense involved with relocating facilities, changing suppliers or reconfiguring networked systems. As an alternative, the fortification strategy can enhance efficiently the reliability of existing logistics system, which is risk mitigation strategies that the firm takes some action in advance of a disruption (and so incurs the cost of the action regardless of whether a disruption occurs). The fortification strategy is implemented through efficient investments in fortifying key facilities in existing logistics systems. Fortification measures include adding built-in redundancies, expanding capacity, installation of structural reinforcements and barriers, preventive maintenance, monitoring and so on. Except for logistics system protection, the fortification strategy have been applied for the
vulnerabilities analysis of electric power grids (Brown et al., 2006), for the optimal allocation of a security budget to a water supply network (Rico-Ramirez et al., 2007), and for the protection of hospital facilities in a seismically active region of Italy (Liberatore et al., 2012). Considering the interdependencies among faculties, the variety of disruptions, the prohibitive costs involved in securing large numbers of facilities, and the limitation of protection investment budget, the development of structured and analytical models able to capture these complexities has been widely perceived as an urgent priority, however the study of mathematical models for fortification planning is still in its infancy.

In this paper, we address a fortification planning problem for existing logistics system with capacitated facilities and the limited protection investment budget. The aim is to improve substantially in reliability of existing systems with minor increases in nominal cost by reasonable allocation of limited protection investment. A fortification strategy is carefully constructed by the risk mitigation combination of facility protection and pre-positioning emergency inventory policies. A model in two-stage stochastic mixed-integer programming framework is proposed to make the following decisions: which facilities should be fortified; where should emergency inventory be pre-positioned; how many emergency inventories should be held in fortified facilities to accommodate emergency demand after disruptions.

The remainder of this paper is organized as follows: In Section 2 we review relevant
literature for the topic; Section 3 focuses on formulating the fortification planning problem for existing capacitated logistics systems; Section 4 reviews briefly the original D2-BAC method and proposes two types of valid cuts, dynamic TB&B tree procedure and the scheme of the revised D2-BAC algorithm; Section 5 evaluates the computational performance of the revised D2-BAC algorithm and the impact of key parameters on the fortification strategy; Finally, we offer some concluding remarks in Section 6.

2. Literature Review

A variety of quantitative approaches have recently been developed to identify cost-effective ways of increasing the reliability of supply chain systems to external disruptions, showing the growing interest and the importance of optimizing reliability in supply chains. Roughly, the investigation of models can be split into two main streams. The first one concerns the development of design models to create an intrinsically reliable supply chain system with a high degree of performance both normally and when a disruption occurs. This stream is focused on by most of the literature (Snyder et al., 2005, Berman et al., 2007, Lim et al., 2010, Cui et al., 2010, and Peng et al., 2011). An excellent review of the literature on reliable supply chain design models can be found in the tutorial by Snyder et al. (2006) and Klibi et al. (2010a). The second one focuses on the development of fortification models for improving in reliability of supply chain system which are already in place and for which a complete reconfiguration would be prohibitively costly. This paper will pay an
emphasis on the fortification mathematical model and its solution.

The development of fortification models is a recent field of research. Most of the fortification models developed so far focus on the identification of efficient protection plans to reduce the impact of worst-case losses when intentional attacks occur. The class of problem can be represented in a leader-follower or Stackelberg game framework between a defender and an intelligent attacker, where the main problem models the defender decisions whereas the inner interdiction subproblem identifies the worst-case scenarios in response to a given protection strategy. An initial example is the \( r \)-interdiction median problem with fortification (RIMF) proposed by Church and Scaparra (2007). The main problem in the RIMF model deals with protection decisions for \( q \) facilities by efficient allocation of limited budget in order to minimize the maximum possible damage and the subproblem is embedded to identify the \( r \) most critical facilities within an existing \( p \)-median network. In subsequent works, two different solution approaches for this deterministic fortification model are discussed by Scaparra and Church (2008a) and Scaparra and Church (2008b). Variations on the basic fortification model have subsequently been proposed that include different model facets, such as a random number of possible facilities to be attacked instead of fixed losses (Liberatore and Scaparra 2010), shortest-path network systems (Cappanera and Scaparra 2011), the role of facility recovery time on system performance and the possibility of multiple disruptions over time (Losada et al. 2012), the propagation of disruptions over large areas and partial damage of capacitated facility (Liberatore et al.
The above fortification models can be case as multi-level defender-attacker models in which deliberate sabotage and terrorist attacks are a major concern.

For the most part of above models, other protection measures except fortified critical facilities, such as strategically-placed emergency inventory, have been completely disregard. Instead, strategically-placed emergency inventory measure is widely used to improve intrinsically reliability of systems in design models, which can accommodate rapidly insufficiency of system inventory and improve in the service level of customers after disruptions, especially for the capacitated supply chain (Schmitt et al. 2011). Besides, fortified facilities are perfect places to hold emergency inventory reserves, and pre-positioning emergency inventory reserves can also help reduce the number of facilities to be fortified.

Another class of fortification models focuses on critical infrastructure protection when accidental disruptions strike, which can be represented as the risk-neutral expected-cost model in the two-stage stochastic program formulation. Protecting against accidental disruptions is fundamentally different from protecting against intentional attack, because accidental disruptions strike diversely and highly randomly and do not adjust their offensive strategies to circumvent protection measures as intelligent adversaries do. Rico-Ramirez et al. (2007) address an optimal sensor placement problem in a municipal water network to detect injected contaminants, in which the first stage decides the number and the location of the sensors and the second
stage evaluates the expected value of the cost associated to the population suffering the attack. Liu et al. (2009) propose an approach of allocating limited retrofit resources over multiple highway bridges to improve the resilience and robustness of existing highway transportation systems. Fortunately, the above models in a two-stage stochastic program formulation without integer variables in the lower-level can be solved in a reasonable time using traditional stochastic programming algorithms such as the L-shaped method, stochastic decomposition algorithm and their improvements (Birge and Louveaux, 1997). Compared with water networks or transportation systems, fortification planning for existing logistics systems is somewhat different, which not only ensures node-to-node connectivity in the face of component failures, but also reassigns customers to the survived facilities in a cost-optimal way. The reassignment decision brings inevitably integer variables into the lower-level of stochastic model and results in the failure of traditional stochastic programming algorithms.

Fortification models in two-stage stochastic programming formulation with integer variables in the lower-level are notoriously difficult to solve optimally. Recently, there have been significant efforts to solve these two-stage stochastic mixed-integer models (SMIP). Sherali and Fraticelli (2002) draw upon the theory of reformulation linearization techniques for sequent convexification of the 0-1 mixed-integer sub-problem (MIP) and develop a new method akin to Benders decomposition (NBD). The method is merely suitable to solve the model including binary variables in lower-level. Sen and Higle (2005) propose the disjunctive decomposition (D2)
algorithm, in which the MIP sub-problem is convexified by a sequence of D2 cuts based on common cut coefficient (C3) theorem. Yuan and Sen (2009) streamline the computation associated with the D2 cuts generation process for the D2 algorithm by taking advantage of the special structure of the cut generation linear program. Computational results for large-scale stochastic server location instances using the D2 method are reported by Ntaimo and Sen (2005, 2008). The disjunctive decomposition-based branch-and-cut algorithm (D2-BAC) is proposed by Sen and Sherali (2006). What makes the D2-BAC algorithm computationally unique and attractive is its ability to provide the second-stage value function convexification based on terminal nodal dual information from the ‘truncated’ branch-and-bound tree (TB&B). The ‘truncated’ strategy can avoid bogging down in attempts to solve an optimally NP-hard sub-problem for a given first-stage solution $x$, even while the particular solution $x$ may not be near a reasonably small neighbourhood of an optimal solution. However, computational investigation of the D2-BAC algorithm on solving realistic-scale instances is currently in a nascent stage, especially speed-ups of the algorithm by using special structure of model.

3. Problem Modelling

There exists a capacitated logistics system with optimal configuration. Let $I$ be the set of customers. Each customer $i \in I$ has a specific demand $h_i$ that can be satisfied by one facility. The set $J$ is the set of existing facilities, and each facility is characterized by a maximal capacity $u_j$. A facility cannot be assigned more demand than its current
capacity. The coefficient $d_{ij}$ represents the cost of delivering a unit demand of customer $i \in I$ from facility $j \in J$. In business-as-usual situations, it is assumed that products are transported from operating facilities to customers in a cost-optimal way in existing logistics system. These operating facilities are susceptible to accidental disruptions, unless protective measures are taken to prevent these disruptions. Once a facility is hit by disruptions, the disrupted facility is completely inoperable throughout its entire recovery time so that customers assigned to it must be emergently reassigned to any non-disrupted facility that has enough excess capacity to accommodate the additional demand. The set $S$ is the set of plausible future disruptions scenarios with a finite number. It is assumed that disruptions scenarios are independent and a disruptions scenario may hit one or more facilities. For example, a flood can destroy multiple facilities in practical situations. The probability of disruptions scenario $q$ can be estimated by historic data and forecast of experts. The parameter $a_j$ is used to indicate whether facility $j$ is hit in scenario $q$. If facility $j$ fails, the parameter $a_j$ is equal to 1 and otherwise 0.

The planner with limited protection investment budget $C$ intends to adopt facility protection and pre-positioning emergency inventory measures to mitigate the impact of some disrupted facilities on a capacitated logistic system. Protection cost $c_j$ is stipulated for facility $j$ to fortify and expand its partial capacity, which depends on the size and location of facility $j$. The fortified facilities are assumed to maintain normal service capability even though disruptions hit them. In addition, emergency inventory can be pre-positioned only in some fortified facilities to reduce the risk of emergency
inventory reserve to be destroyed. It is assumed here that the expanded capacity of the fortified facility does not exceed the original maximal capacity $u_j$. The unit cost of emergency inventory is denoted by $b_j$, which is stipulated high value with penalty to reduce total emergency inventory pre-positioned as much as possible. Based on these conditions, the decisions-making in the fortification planning are the location of facilities to be fortified, the location and quantity of emergency inventory to be pre-positioned and the emergent reassignment from surviving facilities to customers in any scenario realization. The parameters of the model are the following.

Notation and input parameters:

$I$ set of customers ( indexed by $i$ ).

$J$ set of existing facilities (indexed by $j$ ).

$S$ set of plausible disruptions scenarios.

$C$ maximal budget of protection investment.

$h_i$ demand of customer $i$.

$u_j$ capacity of existing facility $j$.

$d_{ij}$ unit cost for severing customer $i$ from facility $j$.

$q$ probability of a disruptions scenario $S$.

$a_j$ 0-1 indicated parameter if facility $j$ is hit in scenario $S$.

$c_j$ protection cost of facility $j$.

$b_j$ unit cost of emergency inventory pre-positioned in facility $j$.

Decision variables:
\( x_j \) \begin{align*}  & 1 \text{ if facility } j \text{ is fortified and 0 otherwise.} \end{align*}

\( z_j \) emergency inventory reserves if facility \( j \) is fortified.

\( y_{ij} \) \begin{align*}  & 1 \text{ if customer } i \text{ is emergently assigned to facility } j \text{ in scenario }, \text{ and 0 otherwise.} \end{align*}

The fortification planning problem can be formulated in a two-stage stochastic mixed-integer programming model (FPSMIP) as follows.

\[
\text{Minimize} \sum_{j \in J} c_j x_j + E[f(x, \omega)] \tag{1}
\]

Subject to \[ \sum_{j \in J} c_j x_j \leq C \tag{2} \]

\[ x_j \in \{0,1\} \quad \forall j \in J \tag{3} \]

Where \[ f(x, \omega) = \min \sum_{j \in J} \sum_{j \in J} h_d y_{ij} + \sum_{j \in J} b_j z_j \tag{4} \]

Subject to \[ \sum_{j \in J} y_{ij} = 1 \quad \forall i \in I \tag{5} \]

\[ \sum_{i \in I} h_{ij} y_{ij} - z_j \leq (1 - a_{j\omega}) u_j + a_{j\omega} x_j \quad \forall j \in J \tag{6} \]

\[ z_j \leq x_j u_j \quad \forall j \in J \tag{7} \]

\[ z_j \geq 0 \quad \forall j \in J \tag{8} \]

\[ y_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \tag{9} \]

The objective function of the first-stage problem (1) minimizes current cost of facility protection and expected cost \( (E[\cdot]) \) of emergency inventory reserves and emergency
transportation over all possible disruptions, where the number and the location of the facilities to be fortified are identified by allocating effectively limited protection investment budget. Constraint (2) requires protection investment allocation not to exceed the maximal budget. Constraint (3) represents the integrality requirement for protection variables. The decisions in the second-stage are the location and quantity of emergency inventory to be pre-positioned and the emergent assignment plan. The objective in the second stage (4) is to minimize the cost of the emergency inventory and the emergency transportation from the non-disrupted facilities to customers, which is a random variable dependent on the first-stage protection decisions and realization of disruptions scenarios. Constraint (5) ensures that each customer must be emergently reassigned to a surviving facility. Capacity constraint (6) must reflect the fact that the facility \( j \), which is fortified or is not interdicted, can serve up to its maximal capacity. A variable \( z_j \) is introduced to serve for emergency inventory pre-positioned in the facility \( j \), which can provide feasibility that all customers can be reassigned to surviving facilities after any disruptions strike; simultaneously, it results in an augmentation of operation cost at a rate of \( b_j \) in business-as-usual situations. Constraint (7) links the upper- and lower-level problems by stipulating that only fortified facility can expand its emergency inventory up to \( u_j \). Constraint (8) requires that emergency inventory variable is not negative. Finally, Constraint (9) represents the integrality requirements for emergent assignment variables.

The deterministic equivalent program (DEP) of this model is a mixed-integer program.
As the scale of the logistics system and the number of scenarios increase, the DEP can become prohibitively large. Difficulties in solving large-scale problems through direct usage of commercial solvers (e.g. CPLEX 11.0) motivate us to use alternative methods based on decomposition (e.g. D2-BAC) with reasonable computing and memory requirements. In order to make the presentation clearer, it is useful to rewrite the model (FPSMIP) in a compact form. Let $A$ denote the resource matrix of the first-stage variables $x$. The second-stage variable vector denoted by $y$ is composed of variables $y_i$ and $z_j$. The vector $g^T$ in the second-stage objective function consists of the coefficients of variables $y_i$ and $z_j$. The matrix $W$ and $T(a)$ denote the fixed recourse matrix and the technology matrix, respectively. The vector $r(a)$ is composed of the right-hand constants in the second-stage constraints. So, the FPSMIP model can be trivially reformulated as follows:

\[
\begin{align*}
\min & \quad c^T x + E[f(x, \omega)] \\
\text{s.t.} & \quad Ax \geq B \\
\text{where} & \quad f(x, a) = \min g^T y \\
\text{s.t.} & \quad Wy \geq r - T(a)x
\end{align*}
\]

4. Solution Methodologies

4.1 Background

The original disjunctive decomposition-based branch-and-cut algorithm (D2-BAC), initiated by Sen and Sherali (2006), allows for the convexification of the second-stage
value function based on the subproblem terminal nodal dual information from the branch-and-bound tree using a strategy from reverse convex programming. The D2-BAC algorithm involves three main tasks per algorithmic iteration: solving a 0-1 integer master program, approximating the value function of mixed-binary subproblems via a sequential convexification process and generating a Benders-type optimality cut to add to the master program. We summarize the convexification process in the second task as follows:

Firstly, solve partially one MIP subproblem using the ‘truncation’ branch-and-bound tree (TB&B) with the given number of nodes.

Secondly, obtain a lower bounding function of the subproblem associated with a terminal node. Let \( Q(\cdot) \) denote the set of terminal nodes of the TB&B tree generated for the subproblem associated with scenario \( \omega \). For any node \( q \in Q(\cdot) \), let \( z_{ql}(\cdot) \) and \( z_{qh}(\cdot) \) denote vectors whose elements are used to define lower and upper bounds, respectively, on the second-stage (integer) variables. Let \( (\theta_q^k(\omega), \psi_{ql}^k(\omega), \psi_{qh}^k(\omega)) \) denote optimal dual multiplier for node \( q \in Q(\cdot) \) at \( k \) iteration. Then a lower bounding function of the subproblem in form (11) is valid when the restrictions associated with a terminal node \( q \) hold true.

\[
\eta \geq \theta_q^k(\omega)^T[r_q(\omega)-T_k(\omega)x]+\psi_{ql}^k(\omega)^Tz_{ql}(\omega)-\psi_{qh}^k(\omega)^Tz_{qh}(\omega)
\]

for at least one \( q \in Q(\cdot) \) (11)
Thirdly, approximate the lower bound of the value function of each subproblem by disjunctive programming. Let $\Pi_k(\alpha)$ denote the union of epigraph for all $q \in Q(\cdot)$. According to the disjunctive cut principle, the cut generation LP (ERP-LP) in form (13) is constructed and solved to derive a facet of the convex hull of the disjunctive set, which would essentially provide us with a convex lower bounding approximation of the value function.

$$\Pi_k(\omega) = \{(\eta, x) \in \bigcup_{q \in Q(\omega)} E^k_q(\omega)\} \quad (12)$$

Where

$$\begin{align*}
E^k_q(\omega) &= \{(\eta, x) \mid \eta \geq v^k_q(\omega) - \gamma^k_q(\omega)^T x, Ax \geq b, x \geq 0, \eta \geq 0\} \\
v^k_q(\omega) &= \theta^k_q(\omega)^T r^k_q(\omega) + \psi^k_q\omega(\omega)^T z^k_q(\omega) - \psi^k_q(\omega)^T z^k_q(\omega) \\
\gamma^k_q(\omega) &= T^k(\omega)^T \theta^k_q(\omega) \\
Max \{-\eta^k(\omega)\sigma_q(\omega) - \sum_j x^j \sigma_j(\omega) + \xi(\omega)\mid \sigma_q(\omega), \sigma(\omega), \xi(\omega) \in \Pi^*_k(\omega)\} \quad (13)
\end{align*}$$

Where $\Pi^*_k(\omega) = \{\sigma_q(\omega) \in R, \sigma(\omega) \in R^{n_1}, \xi(\omega) \in R \mid \forall q \in Q(\omega), \exists \tau_q(\omega) \geq 0, \tau_{0q}(\omega) \in R,\}$

s.t. $c_0(\alpha) \geq \tau_{0q}(\alpha), \forall q \in Q(\alpha)$

$$\sum_{q \in Q(\omega)} \tau_{0q}(\omega) = 1$$

$$\sigma_j(\omega) \geq \tau_q(\omega)^T A_j + \tau_{0q}(\omega) \gamma^k_q(\omega), \forall q \in Q(\omega), j = 1,\ldots, n_1$$

$$\xi(\omega) \leq \tau_q(\omega)^T b + \tau_{0q}(\omega) v^k_q(\omega), \forall q \in Q(\omega)$$

$$\tau_q(\alpha) \geq 0, \tau_{0q}(\alpha) \geq 0, \forall q \in Q(\alpha)$$

Finally, generate Benders-type optimality cut by using the expectation operator for all scenarios as follows:
\[ \eta \geq E\left[ \frac{\xi^i(\omega)}{\sigma_0^i(\omega)} \right] - E\left[ \frac{\sigma^i(\omega)}{\sigma_0^i(\omega)} \right]^T x \]  

(14)

where \( \sigma_0^i(\omega), \sigma^i(\omega), \xi^i(\omega) \) is an optimal solution of (13). Then the Benders-type optimality cut is added to the master problem. The original D2-BAC algorithm iterates until the termination condition is satisfied.

### 4.2 Valid Disjunctive Decomposition Cut

The disjunctive decomposition cut (D2), proposed by Sen and Higle (2005), provides the sequential convexification ability of the second-stage feasible set with the form

\[ \pi^T y \geq \pi_0(x, \omega). \]

The common cut coefficient derived for one instance of the second-stage problem can be used for other scenario instances, which is obtained by solving a simple recourse stochastic linear program C3-LP. The piecewise-linear concave function \( \pi_0(x, \alpha) \) is derived by solving another linear program RHS-LP for each scenario. It is obvious that a large number of C3-LPs and RHS-LPs are solved to generate a sequence of D2 cuts in iteration process. Yuan and Sen (2009) streamline the computation of the D2 cut generation process and propose some weaker valid D2 cuts, instead of the original D2 cuts \( \pi^T y \geq \pi_0(x, \omega) \). To obtain the common cut coefficient, the revised Benders decomposition is designed to solve efficiently the C3-LP without using any LP solver by utilizing a very special structure of the C3-LP. The right-hand side \( \pi_0(x, \alpha) \) can be generated directly through its convexification \( \pi^k_0(x, \omega) \) without solving any RHS-LP as follows:

\[ \pi^k_0(x, \omega) \leq \pi_0(x, \omega) \]
\[
= \min \{\alpha_i(\omega), \alpha_2(\omega)\} + \sum_i (\min \{\beta_{ij}(\omega), \beta_{i,j}(\omega)x_i\})
\]

where \( \alpha_i(\omega) = (\lambda_{n1}^i)^T r_1(\omega); \quad \alpha_2(\omega) = (\lambda_{n1}^i)^T r_1(\omega) + \lambda_{n2}^i; \quad \beta_i(\omega)^T = -(\lambda_{n1}^i)^T T_k(\omega); \]

\( \beta_2(\omega)^T = -(\lambda_{n1}^i)^T T_k(\omega); \) The multiples \( \lambda_{n1}^k, \lambda_{n2}^k, \lambda_{n2}^k \) can be obtained by solving efficiently the C3-LP in iteration \( k \). This effort of simplified D2 cut generation provides a feasibility that the original D2-BAC method integrates with D2 cuts.

### 4.3 Facet-defining Cover Inequalities

We now develop facet-defining cover inequalities for capacity constraints (6). Given protection variable \( x_i \), facility \( j \) and scenario, the right-hand side of capacity constraints (6), viz. \((1-a_j)u_j + a_jx_ju_j\), is equal to \( u_j \) or 0. When \(((1-a_j)u_j + a_jx_ju_j)\) is \( u_j \), capacity constraint (6) is the 0-1 mixed knapsack inequality with a single non-negative continuous variable, from which some facet-defining cover inequalities can be derived by the lifting theory of continuous variable presented by Marchand et al. (1999).

Considering the set:

\[
Y_j^w(x_j) = \{ (y_{j\omega}, z_j) \in B^j \times R^l : \sum_i h_iy_{j\omega} \leq u_j + z_j \}
\]

Now let a \( k \)-cover pair \((k, C)\) be a cover for \( Y_j^w(x_j) \) if \( i \in C \) \( I, \sum_{i \in C} h_i = u_j + \lambda \) with \( \lambda > 0 \) and \( \sum_{i \in C \setminus k} h_i < u_j \). Let \( Y_{jc}^w(x_j) = \{ (y_{j\omega}, z_j) \in Y_j^w(x_j) : y_{j\omega} = 0; i \in I \setminus C \}\) and \( \tilde{C} = \{ i \in C : h_i > \lambda \} \). Supposing that \( \tilde{C} = \{ 1, \ldots, r \} \) with \( h_1, \ldots, h_r \), let \( A_i = \sum_{j=1}^{\tilde{C}} h_i \) for \( i \in r \), and \( A_0 = 0 \). Facet-defining inequalities of 0-1 mixed knapsack polytope are obtained using the following process: Firstly, fix all but one of the binary variables \( y_{j\omega} \) and derive a facet-defining inequality in the two-dimensional space.
consisting of one binary variable and the continuous variable; Secondly, lift in the variables have been set to $y_{kj} = 1$ for $i \in C \setminus \{k\}$ and obtain facet-defining for $\text{conv}(Y_{jC}^0(x_j))$ as in Proposition 1; Thirdly, lift in the variables from $y_{kj} = 0$ for $i \in I \setminus C$ and obtain facet-defining for $\text{conv}(Y_j^0(x_j))$ as in Proposition 2.

**Proposition 1** (Marchand et al. 1999). If $(k, C)$ is a $k$-cover for $Y_j^0(x_j)$, the inequality

$$
\sum_{i \in \hat{C}} \lambda y_{j\omega} + \sum_{i \in \hat{C}} h_i y_{j\omega} \leq \left(\mid \hat{C} \mid - 1\right) \lambda + \sum_{i \in \hat{C}} h_i + z_j \tag{16}
$$

is facet-defining for $\text{conv}(Y_{jC}^0(x_j))$.

**Proposition 2** (Marchand et al. 1999). If $(k, C)$ is a $k$-cover for $Y_j^0(x_j)$, the inequality

$$
\sum_{i \in \hat{C}} \lambda y_{j\omega} + \sum_{i \in \hat{C}} h_i y_{j\omega} + \sum_{i \in I \setminus C} \phi_C(h_i) y_{j\omega} \leq \left(\mid \hat{C} \mid - 1\right) \lambda + \sum_{i \in \hat{C}} h_i + z_j \tag{17}
$$

is facet-defining for $\text{conv}(Y_j^0(x_j))$. Where $C$ is superadditive on $R_1$ as follows:

$$
\phi_C(\alpha) = \begin{cases} 
(i-1)\lambda & \text{if } A_i \leq \alpha \leq A_i - \lambda, \quad i = 1, \ldots, r \\
(i-1)\lambda + [\alpha - (A_i - \lambda)] & \text{if } A_i - \lambda \leq \alpha \leq A_i, \quad i = 1, \ldots, r-1 \\
(r-1)\lambda + [\alpha - (A_r - \lambda)] & \text{if } A_r - \lambda \leq \alpha 
\end{cases} \tag{18}
$$

When $((1-a_j)u_j + a_jx_j u_j) = 0$, viz. $\sum_{i \in I} h_i y_{j\omega} - z_j \leq 0$, it means that facility $j$ is disrupted without fortification, so $z_j$ and $y_{j\omega}$ are equal to 0 due to constraint (7) $z_j: x_j u_j$. In order to obtain a uniform form of valid inequalities of constraint (6) whatever is $(1-a_j)u_j + a_jx_j u_j$, we only need revise slightly continuous cover inequality (17) as follows:

$$
\sum_{i \in \hat{C}} \lambda y_{j\omega} + \sum_{i \in \hat{C}} h_i y_{j\omega} + \sum_{i \in I \setminus C} \phi_C(h_i) y_{j\omega} - z_j \leq \left(\mid \hat{C} \mid - 1\right) \lambda + \sum_{i \in \hat{C}} h_i (1-a_j\omega + a_j\omega x_j) \tag{19}
$$
Note that this cover inequality (19) remains valid for any value of $x_j$ and for any scenario. Once the $k$-cover pair $(k, C)$ is given, its right-hand side is determinate, so the cover inequality (19) can be viewed as a global constraint. The procedure to obtain facet-defining continuous cover inequalities of $\text{conv}(Y_j(x_j))$ is summarized as follows:

**Procedure FDCCI**

Step 1. Generate greedily a fixed number of minimal cover of set $Y_j(x_j)$ by a greedy approach and obtain every $k$-cover pair $(k, C)$, and $\tilde{C}$.

Step 2. Obtain the facet-defining inequality (16) for $\text{conv}(Y_{jC}(x_j))$ by Proposition 1.

Step 3. Build superadditive lifting function $c$ (18) and calculate $c(h_i)$ for $i \in I \setminus C$.

Step 4. Obtain the facet-defining inequality (17) for $\text{conv}(Y_j(x_j))$ by Proposition 2.

Step 5. Repeat Steps 1 to 4 until all $k$-cover pairs are dealt with.

### 4.4 Implementation of Dynamical TB&B tree

The TB&B tree procedure is an important part of the D2-BAC algorithm. For each scenario, the TB&B tree procedure is used to solve partially one subproblem and acquires the optimal dual solution from the terminal nodes of the TB&B tree in order to form Benders-type optimality cut (14). A scheme of the TB&B tree procedure is designed as follows:
TB&B tree procedure

Step 1. Initialize.

1.1 Input: the first-stage current solution $x^k$, scenario and maximum number of nodes to explore $\text{max\_numnd}$.

1.2 Initialize fathomed nodes $fathomed\_numnd$, current nodes not to be explored $\text{non\_numnd}$, and current best bound $\text{best\_bound}$; set the vacant queue of currently active open nodal problems $L$.

1.3 Form and solve the root node by $\text{ILP}_0$. If the solution is feasible and integral, fathom the root node to form inequality (11) and return to the main procedure of the D2-BAC algorithm; otherwise set the root node to be $L = \{\text{ILP}_0\}$ and proceed to Step 2.

Step 2. Termination criteria.

If $L = \emptyset$ or $\text{non\_numnd} + fathomed\_numnd > \text{max\_numnd}$, return to the main procedure of the D2-BAC algorithm with inequality (11) for all terminal nodes, $\text{best\_bound}$ and the best solution; otherwise proceed to Step 3.

Step 3. Fathom terminal nodes.

3.1 Select a node $P_q$ from the head of queue $L$ and set $L/L/ P_q$.

3.2 If the solution $y^k(\cdot)$ of the current node is feasible and integral, then the current node is denoted terminal node; and if the objective value $\text{objvalue} \leq \text{best\_bound}$, then update the $\text{best\_bound} = \text{objvalue}$ and set $y^k(\cdot)$ to be the incumbent solution.

3.3 If the solution $y^k(\cdot)$ of the current node is fractional feasible and $\text{objvalue} > \text{best\_bound}$, then the current node is denoted terminal node; otherwise proceed to Step
4.

3.4 Fathom the terminal node to acquire the optimal dual multiplier and form the node inequality (11); \( fathomed\_numnd = fathomed\_numnd + 1 \) and go to Step 2.

3.5 If the solution \( y^k(.) \) of the current node is fractional feasible and \( objvalue \) \( best\_bound \), then proceed to Step 4.

**Step 4. Branch with breadth-first strategy.**

4.1 Create two nodes \( P_{q0} \) and \( P_{q1} \) using a disjunction variable index \( j \) determined by \( \{j = \arg\min \{|y_j( ) - 0.5|\} \} \).

4.2 If two nodes \( P_{q0} \) and \( P_{q1} \) are both feasible, then the problem with the smaller objective value is added to the tail of \( L \) prior to the other; go to Step 2.

4.3 If only one node is feasible, then the feasible problem is added to the tail of \( L \); go to Step 2.

4.5 The Revised D2-BAC Algorithm

The revised D2-BAC algorithm is improved in two aspects. One improvement represents that two types of valid cuts are integrated into the original D2-BAC algorithm, viz. facet-defining cover inequalities for facility capacity constraints and simplified D2 cuts. The facet-defining cover inequalities (19), derived from facility capacity constraints, are generated \textit{a priori} and then added into the FPSMIP model to help overcome computational barrier by obtaining tighter LP relaxation of subproblems. These simplified D2 cuts, instead of the original D2 cuts, are generated sequentially and added into the FPSMIP model during the execution process of the
revised D2-BAC algorithm, which can accelerate the revised D2-BAC algorithm by the combination of value function convexification and integer-feasible solution set convexification of the second-stage inherited from the D2 method. Another improvement represents that the dynamical ‘truncation’ strategy of the TB&B tree, instead of the original fixed number of nodes for the TB&B tree, will be implemented by controlling dynamically the number of nodes to explore and the branch strategy. In the initial stage of the revised D2-BAC process where the first-stage solutions are generally not anywhere near the neighbourhood of an optimal solution, the number of nodes is specified as small integer in order to avoid bogging down in attempts to ‘dive’ deep into the TB&B tree. As iterations proceed and the gap between the lower and upper bound is reduced, the number of nodes increases suitably. Especially when the gap between the lower and upper bound remains constant for a given number of iterations or the first-stage solution stops changing, the branch-and-bound tree will be explored completely to obtain the optimal solution of the subproblem. The branch strategy of the TB&B tree follows a breadth-first strategy with node selection always favouring the node with the best objective value of the subproblem. The flowchart of the revised D2-BAC algorithm for the FPSMIP is designed in Figure 1.

5.  Computational Results

5.1 Problem Instance Generation

In this section we report on the computational instances. First, the test beds for
fortification planning are generated, which are the optimal solution of the capacitated location problems. Data of the capacitated location problems are randomly generated as follows: (a) Points representing demand nodes and potential facilities are uniformly randomly generated in a unit square. Unit transportation cost from facilities to customers is then stipulated by multiplying the Euclidean distances by 100; (b) Demand of a customer is randomly generated from uniform distribution in the interval \( U[5, 35] \) and facility capacity \( u_k \) is generated in the interval \( U[10, 160] \); (c) Fixed cost of a facility is generated according to the formula \( f_k = U[0, 90] + U[100, 110] \sqrt{u_k} \) to take into account the economies-of-scale. Second, scenario data are generated as follows: Set of disruptions scenarios is generated randomly using different seeds so that one or multiple facilities could fail in a scenario. Each scenario is assigned the same probability, but the failure probability of the facilities is perhaps different, because the same facility maybe exists in multiple scenarios. Besides the experiments with different probability of scenarios are done to further confirm the results. The sets of scenarios are checked out to make sure that there are no duplicated scenarios in any instance. Third, the coefficient of the emergency inventory cost \( b_j \) is fixed at 1,000, which is a sufficiently high penalty cost to minimize emergency inventory pre-positioned in fortified facilities in the optimal solution.

As a mnemonic, test beds are named according to the convention \( \text{LS}_m.n.S.r.t \), where \( m \) is the number of capacitated facilities, \( n \) is the number of customers, \( S \) is the number of disruptions scenarios, \( r \) is the ratio of all facilities capacities to all customers
demands and \( t \) is the ratio of protection investment budget to maximal investment demand used to fortify all facilities. According to above-mentioned generation method of instances, the following test beds are obtained to simulate existing capacitated logistic systems in a business-as-usual situation: LS_5.15, LS_10.30 and LS_15.50. Based on three test beds, the fortification strategy and the performance of the revised D2-BAC algorithm are explored by varying the value of \( S, r \) and \( t \) in the next section.

5.2 Results Analysis

In the experimental plan, the first objective is to study the computational performance of the revised D2-BAC algorithm by comparing with CPLEX and the original D2-BAC algorithm; the second objective is to explore how change in key parameters affects the fortification strategy of existing capacitated logistic systems.

5.2.1 Computational Performance

The D2-BAC algorithm is implemented in Java integrated with CPLEX 11.0 concert technology. All the experiments are conducted on a computer with a Pentium D CPU @ 2.8 GHz, 2 GB of RAM. The 2,400 seconds of CPU time is imposed as one of the termination criteria. The instances are run to optimality or stopped when a CPU time limit is reached. If possible, as a benchmark, the CPLEX MIP solver is used to solve the large-scale DEP formulation for each instance. To get the best CPU time for the DEP, the CPLEX parameters are set at the following values based on preliminary testing: “MIPEmphasis 1” (emphasize feasibility over optimality), and “MIPOrdInd
on $x$” (use branches priority order first on $x$). At each iteration of the D2-BAC algorithm, the 0-1 master problem is solved to optimality using CPLEX MIP solver.

The preferable dynamical ‘truncation’ strategy of TB&B tree is found out on the basis of preliminary analysis of many computational experiments as follows: at the initial stage of the revised D2-BAC algorithm, $max_{numnd}$ is set at 2, which simply means that only one decision variable is selected for the branch, thus not more than two nodal subproblems are solved to get the duals and create Benders-type cut for this subproblem; at the medium stage of the revised D2-BAC algorithm, $max_{numnd}$ is set as the integer multiple of iterations and $fathomed_{numnd}$ is stipulated at not more than 3, simultaneously; at the final stage of the revised D2-BAC algorithm, such as the gap between the lower bound and upper bound below 8%, the branch-and-bound process is activated completely. Furthermore, whenever there is no significant improvement in the lower bound ($<0.01\%$) for two consecutive iterations, the branch-and-bound process is also activated completely.

Insert Table 1

The computational results of three fortification planning instances using the CPLEX, the original D2-BAC algorithm and the revised D2-BAC algorithm, respectively, are reported in Table 1. The number of binary variables and constraints are that of the DEP. We also report the minimum time (Min), the maximum time (Max) and the average time for CPU in seconds for the five replications. Three observations should be made from the tabulated results in Table 1: first, the revised D2-BAC algorithm has the
better performance compared with the CPLEX for all the instances except the smaller instance. With the scale of instance increasing, the revised D2-BAC algorithm takes much more time to solve, but can still solve all the instances in a reasonable time; the CPLEX performs better only on smaller instances but fails to solve most of the instances; second, the average percentage of improvement measured by CPU time savings is around 19% and the percentages in saving for LS_15.50 are more significant, compared with the original D2-BAC algorithm; third, an increase in the scale of the master problem and the subproblems do have an adverse effect while an increase in the number of scenarios does not result in a visible effect for the revised D2-BAC algorithm. As a whole, the performance of the revised D2-BAC algorithm is approximately linear with increasing problem size, which is a desired algorithmic behaviour for scalability.

**Insert Table 2**

Table 2 gives the computational results of two fortification planning instances with the different values of the ratio $r$ and $t$ to study how the two parameters affect the performance of the revised D2-BAC algorithm. Two trends should be made from the tabulated results in Table 2: first, CPU time of the revised D2-BAC algorithm will decrease significantly when the value of the ratio $r$ increases for the fortification planning instance with the same value of the ratio $t$; second, CPU time of the revised D2-BAC algorithm will increase significantly when the value of the ratio $t$ increases for the fortification planning instance with the same value of the ratio $r$. Despite the
revised D2-BAC algorithm can solve all the instances, it takes more time to solve the instance with the larger value \( t \) and the smaller value \( r \). The larger value of the ratio \( t \) possibly means more first-stage feasible solutions and the smaller value of ratio \( r \) means more infeasible nodes in the TB&B tree due to tighter capacity constraints in the subproblems, which jointly result in increased computational time of the revised D2-BAC algorithm.

5.2.2 Analysis of Fortification Strategy

We now discuss the impact of protection investment budget, disruptions scenarios, the scale of logistic system and the ratio \( r \) on the fortification strategy of the capacitated logistic systems, respectively.

First, we analyze the impact of protection investment budget on the fortification strategy. For each fortification planning instance with the same disruptions scenarios, we vary the ratio \( t \) from 0% to 100% and then observe some change in fortified facilities, emergency inventory and the efficiency gain. The optimal fortification strategies for the LS_10.30 and the LS_15.50 with the protection investment budget increasing under different scenarios are shown in Table 3. When the ratio \( t \) is less than 25% for the LS_10.30.10, there exists no feasible fortification strategy due to deficiency of total capacity in the surviving system after disruptions. It means that the logistic system breaks down once the disruption scenarios are realized. Increasing the ratio \( t \) from 25% to 40%, the efficiency of the surviving system is improved
remarkably, which is usually achieved with the two or three fortified facilities and the decreasing emergency inventory. Subsequently, the more protection investments with the ratio $t$ between 45% and 60% cannot bring higher efficiency gain but progressively lower efficiency gain; especially with the ratio $t$ between 65% and 100%, there is no benefit in increasing protection expenditure. Besides, when the ratio $t$ is more than 60%, the investment of protection cost may be so much that the fortification strategy is an infeasible risk mitigation strategy for existing logistics systems. Because the risk mitigation strategy incurs the cost of the protection action regardless of whether a disruption occurs. Furthermore, there are three facilities (5, 7 and 9) which appear in most optimal fortification strategies and which obviously should be the core sets of key facilities. The emergency inventory pre-positioned in the fortified facilities is decreasing with the protection investment budget increasing, which has an important effect on total efficiency. This analysis can be used not only to identify core sets of facilities to be fortified, but also to find suitable emergency inventory pre-positioned in fortified facilities. It is clear that there is an optimal trade-off point between fortified facilities and emergency inventory. These results also confirm the significance of our risk mitigation measures combining facility protection and pre-positioning emergency inventory for existing capacitated logistic systems in the fortification planning.

Second, we analyze the impact of disruptions scenarios on the fortification strategy. As shown in Table 3, when we vary the number of disruptions scenarios from 10 to 30 for LS_10.30, the minimal ratio $t$, which can get a feasible fortification strategy, increases
from 25% to 28%; synchronously, more emergency inventory needs to be pre-positioned in fortified facilities. For an example, the total emergency inventories for the LS_10.30.30 with the ratio $t$ up to 40% are 6 times more than those for the LS_10.30.10 with the same ratio $t$; even for the ratio $t$ up to 60%, emergency inventory still exists in the fortification strategy. Comparing the fortified facilities set of the LS_10.30.30 with those of the LS_10.30.10, most fortified facilities are occurring in all of them. So it is clear that a fortification strategy hedging against more disruptions scenarios means more protection investments to improve the reliability of the logistics systems.

Third, we analyze the impact of the scale of the logistic system and the ratio $r$ on the fortification strategy. Comparing the fortification strategy of the LS_10.30.10 with those of the LS_15.50.10 in Table 3, the trend is evident that more cost is invested with the scale of the logistic system increasing, which represents as higher minimal ratio $t$, more fortified facilities and more emergency inventory pre-positioned. To investigate the effect of different ratio $r$ on the fortification strategy, we vary the ratio $r$ from 1.19 to 1.51 for the LS_10.30.10 and achieve the different fortification strategies in Table 4. For the LS_10.30.10 with the same ratio $t$, the operation cost of the fortified logistics system and the emergency inventory pre-positioned in fortified facilities decrease markedly as the ratio $r$ increases, whereas the fortified facilities sets are very similar to each other and some key facilities occur repeatedly in them. This is because the capacitated logistics system with the higher ratio $r$ originally holds more excess
capacity to accommodate the loss of facilities capacity after disruptions.

Insert Table 3-4

Finally, we find the similar analysis results from computational experiments with different probability of scenarios, though there are a bit of difference in some details. The above analysis of fortification strategy can lead to the following recommendations helping us to develop cogent fortification plan for a capacitated logistics system: the ratio $r$ along with the scale of the logistic system should been investigated to understand fully the inherent reliability of the existing system. For example, a large-scale logistics system with the small ratio $r$ is more vulnerable to disruptions; all important plausible disruption scenarios including type and probability must been selected carefully, for they have an important effect on the fortification plan; the trade-off between fortified facilities and emergency inventory should be explored to minimize the operation cost and protection investments. From a practical perspective, emergency inventory pre-positioned in fortified facilities should be reduced as much as possible if protection investment is relatively inexpensive, for more emergency inventory usually results in more operation cost in business-as-usual situations. A natural question to ask is: how is a fortification plan implemented to improve the reliability of a capacitated logistics system? For an example of the durable goods logistics system, the manufacturer needs to fortify only a few key distribution centres in-house and pre-position suitable emergency inventory in them; the other distribution centres can be outsourced to the third party. Thus, even if part of the third-party
distribution centres suddenly fail, its emergency inventory pre-positioned in the fortified distribution centres still makes the surviving logistics system perform well. Furthermore, this strategy can also promote the third-party distribution centres to perform better, because manufacturers can credibly claim to be able to operate without badly performing distribution centres, at least in the short term.

6. Conclusions

In this paper, the important risk mitigation combination of facility protection and emergency inventory pre-positioning measures has been incorporated into a fortification model that identifies the optimal allocation of limited protection investment budget in an existing capacitated logistics system in order to hedge against the impact of accidental facility disruptions. The resulting formulation is a two-stage stochastic mixed-integer program, which it is notoriously difficult to solve optimally due to the presence of integer variables in the second-stage. To improve the efficiency of the original D2-BAC algorithm, two types of valid cuts and dynamical ‘truncation’ strategy of the TB&B tree are proposed to embed into the D2-BAC algorithm. Example results indicate that the revised D2-BAC algorithm is generally more efficient than the original D2-BAC algorithm, especially for realistic instances of fortification planning problems. Finally, a parametric analysis is conducted to reveal how key parameters, including protection investment, disruption scenarios, the scale of the logistics system and the ratio $r$, affect the fortification plan of existing capacitated logistic systems. Empirical results suggest that there is a value of protection budget
after the marginal improvement in efficient becomes negligible for additional protection resources. Furthermore, this analysis highlights that the trade-off between fortified facilities and emergency inventory reserves should be designed carefully in order to minimize protection investments and the operation cost in business-as-usual situations. These managerial insights from parametric analysis help to identify the optimal fortification plan for an existing capacitated logistics system under the accidental risk of facility disruptions.

The attempt of multiple risk mitigation measures incorporating redundancy and facility protection policies are undoubtedly important to improve in reliability of capacitated logistics systems. We hope this model will set the groundwork for the development of other fortification models considering other elements and decision objective, such as partial loss of facility capability, multi-echelon logistics networks, worst-case losses, and the conditional value at risk objective. Besides, the revised D2-BAC algorithm is general and can be tailored to solve other models in two-stage stochastic mixed-integer program framework.

References


hedging against disruptions with ripple effects in location analysis. Omega, 40 (1), 21–30

Appendices:

1. Figure 1. The flowchart of the revised D2-BAC algorithm for FPSMIP
2. List of Tables

Table 1. Computational result of three test-beds using respectively CPLEX, the original D2-BAC algorithm and the revised D2-BAC algorithm

<table>
<thead>
<tr>
<th>Problem</th>
<th>Bins.</th>
<th>Cons.</th>
<th>DEP</th>
<th>The original D2-BAC (s)</th>
<th>The revised D2-BAC (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg</td>
<td>Min</td>
</tr>
<tr>
<td>LS_5.15.30</td>
<td>2255</td>
<td>606</td>
<td>1.10</td>
<td>0.59</td>
<td>0.33</td>
</tr>
<tr>
<td>LS_5.15.50</td>
<td>3755</td>
<td>1006</td>
<td>2.45</td>
<td>0.75</td>
<td>0.52</td>
</tr>
<tr>
<td>LS_10.30.5</td>
<td>1510</td>
<td>211</td>
<td>42</td>
<td>53</td>
<td>35</td>
</tr>
<tr>
<td>LS_10.30.10</td>
<td>3010</td>
<td>411</td>
<td>560</td>
<td>169</td>
<td>108</td>
</tr>
<tr>
<td>LS_10.30.30</td>
<td>9010</td>
<td>1211</td>
<td>&gt;2400</td>
<td>0.77%</td>
<td>277</td>
</tr>
<tr>
<td>LS_10.30.50</td>
<td>4373</td>
<td>391</td>
<td>1457</td>
<td>&gt;2400</td>
<td>3.68%</td>
</tr>
<tr>
<td>LS_10.30.100</td>
<td>48020</td>
<td>5085</td>
<td>&gt;2400</td>
<td>8.19%</td>
<td>537</td>
</tr>
<tr>
<td>LS_15.50.5</td>
<td>4373</td>
<td>391</td>
<td>1457</td>
<td>&gt;2400</td>
<td>3.68%</td>
</tr>
<tr>
<td>LS_15.50.10</td>
<td>7515</td>
<td>666</td>
<td>&gt;2400</td>
<td>2.46%</td>
<td>814</td>
</tr>
<tr>
<td>LS_15.50.15</td>
<td>11265</td>
<td>991</td>
<td>&gt;2400</td>
<td>4.95%</td>
<td>1428</td>
</tr>
<tr>
<td>LS_15.50.20</td>
<td>15015</td>
<td>1316</td>
<td>&gt;2400</td>
<td>6.97%</td>
<td>2081</td>
</tr>
</tbody>
</table>

Table 2. Computational results for two fortification planning test-beds with the different values of the ratio \(t\) and \(r\)
<table>
<thead>
<tr>
<th>Problem</th>
<th>Ratio $t$</th>
<th>Ratio $r$</th>
<th>Average nodes</th>
<th>The revised $D^2$-BAC(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>LS_10.30.50</td>
<td>1.19</td>
<td>8402.3</td>
<td>174</td>
<td>290</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>6792.9</td>
<td>114</td>
<td>174</td>
</tr>
<tr>
<td></td>
<td>1.51</td>
<td>4284.1</td>
<td>78</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>1.19</td>
<td>13647.4</td>
<td>482</td>
<td>878</td>
</tr>
<tr>
<td></td>
<td>1.51</td>
<td>10432.0</td>
<td>215</td>
<td>284</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>7581.2</td>
<td>157</td>
<td>229</td>
</tr>
<tr>
<td>LS_15.50.20</td>
<td>1.2</td>
<td>10743.6</td>
<td>1493</td>
<td>2021</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>8392.0</td>
<td>763</td>
<td>1783</td>
</tr>
<tr>
<td></td>
<td>1.7</td>
<td>7228.3</td>
<td>698</td>
<td>1532</td>
</tr>
<tr>
<td></td>
<td>1.51</td>
<td>11210.0</td>
<td>1523</td>
<td>2328</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>9897.6</td>
<td>1334</td>
<td>2132</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>8695.4</td>
<td>832</td>
<td>1693</td>
</tr>
</tbody>
</table>

Table 4. The fortification strategy for the LS_10.30.10 with different ratio $r$

<table>
<thead>
<tr>
<th>Ratio $t$</th>
<th>Ratio $r$</th>
<th>Optimal value</th>
<th>Fortified facility</th>
<th>Emergency inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>1.19</td>
<td>failure</td>
<td>failure</td>
<td>failure</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>193125.4</td>
<td>6 9</td>
<td>$Z_6=73.0; Z_9=107.0$</td>
</tr>
<tr>
<td></td>
<td>1.51</td>
<td>146841.1</td>
<td>6 9</td>
<td>$Z_6=72.0; Z_9=62.0$</td>
</tr>
<tr>
<td>40%</td>
<td>1.19</td>
<td>92399.6</td>
<td>5 7 9</td>
<td>$Z_5=29.0; Z_7=50.0$;</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>32871.1</td>
<td>5 7 9</td>
<td>$Z_5=6.0; Z_7=13.0$</td>
</tr>
<tr>
<td></td>
<td>1.51</td>
<td>13242.1</td>
<td>7 8 9</td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td>1.19</td>
<td>22403.5</td>
<td>1 5 6 7 9</td>
<td>$Z_1=2.0; Z_5=7.0$</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>12586.9</td>
<td>0 1 5 6 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.51</td>
<td>12091.6</td>
<td>0 2 6 7 8</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. The fortification strategy for the LS_10.30 and the LS_15.50 with the increasing protection invest under different scenarios
<table>
<thead>
<tr>
<th>Problem</th>
<th>Ratio</th>
<th>Optimal value</th>
<th>Fortified facility</th>
<th>Emergency inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;25%</td>
<td>failure</td>
<td>failure</td>
<td>failure</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>193125.4</td>
<td>6 9</td>
<td>Z_6=73.0; Z_9=107.0</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>147369.2</td>
<td>5 9</td>
<td>Z_5=109.0; Z_9=23.9</td>
</tr>
<tr>
<td></td>
<td>35%</td>
<td>106287.6</td>
<td>0 7 9</td>
<td>Z_0=44.0; Z_7=47.9; Z_9=1.0</td>
</tr>
<tr>
<td>LS_10.30.10</td>
<td>40%</td>
<td>32871.1</td>
<td>5 7 9</td>
<td>Z_5=6.0; Z_7=13.0</td>
</tr>
<tr>
<td></td>
<td>45%</td>
<td>32851.4</td>
<td>5 7 9</td>
<td>Z_5=6.0; Z_7=13.0</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>20235.3</td>
<td>5 6 7 9</td>
<td>Z_5=0.0; Z_6=6.0; Z_7=1.0</td>
</tr>
<tr>
<td></td>
<td>55%</td>
<td>13363.9</td>
<td>1 2 5 7 9</td>
<td>Z_7=1.0</td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>12586.9</td>
<td>0 1 5 6 7 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>70%</td>
<td>12280.9</td>
<td>0 1 2 5 6 7 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80%</td>
<td>12280.9</td>
<td>0 1 2 5 6 7 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>12280.9</td>
<td>0 1 2 5 6 7 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>12280.9</td>
<td>0 1 2 5 6 7 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;28%</td>
<td>failure</td>
<td>failure</td>
<td>failure</td>
</tr>
<tr>
<td>LS_10.30.30</td>
<td>28%</td>
<td>174517.9</td>
<td>5 9</td>
<td>Z_5=120.0; Z_9=42.0</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>124531.9</td>
<td>3 5 6</td>
<td>Z_3=4.0; Z_5=36.0; Z_9=73.0</td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>29241.3</td>
<td>0 3 5 7 9</td>
<td>Z_3=3.0; Z_7=13.0</td>
</tr>
<tr>
<td></td>
<td>&lt;36%</td>
<td>failure</td>
<td>failure</td>
<td>failure</td>
</tr>
<tr>
<td>LS_15.50.10</td>
<td>36%</td>
<td>378338.3</td>
<td>2 5 7 8 11</td>
<td>Z_2=91.0; Z_8=85.0; Z_9=56.0; Z_11=50.0</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>347093.0</td>
<td>1 2 5 7 11</td>
<td>Z_5=85.0; Z_7=20.0; Z_11=50.0</td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>202997.5</td>
<td>1 2 5 6 8 9 11 13</td>
<td>Z_2=77.0; Z_5=2.0; Z_8=40.0; Z_9=2.0; Z_11=49.0; Z_13=7.0</td>
</tr>
</tbody>
</table>
This paper presents a fortification planning model for capacitated logistics systems in a two-stage stochastic mixed-integer programming framework. Firstly, the risk mitigation combination of facility fortification and emergency inventory pre-positioning policies is proposed to hedge well against accidental disruptions in the capacitated logistics systems. Secondly, a new disjunctive decomposition-based branch-and-cut (D2-BAC) algorithm for the model is developed by integrating with two types of valid cuts and dynamical ‘truncation’ strategy of the branch-and-bound tree. Thirdly, the proposed method indicates the model can provide a powerful tool for identifying best possible fortification strategies. Finally, the risk mitigation combination can significantly increase the reliability of capacitated logistics systems.