Journal of Optimization in Industrial Engineering
Vol.13, Issue 2, Summer & Autumn 2020, 211- 226
DOI: 10.22094/JOIE.2019.1868275.1654

Research Article

A Risk-averse Inventory-based Supply Chain Protection Problem with Adapted Stochastic Measures under Intentional Facility Disruptions: Decomposition and Hybrid Algorithms

Sajjad Jalali [a], Mehdi Seifbarghy [b,*], Seyed Taghi Akhavan Niaki [c]

[a] *Department of Industrial Engineering,, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[b] *Department of Industrial Engineering, Alzahra University, Iran*
[c] *Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran*
Received 10June 2019; Revised 22 October 2019; Accepted 28 October 2019

Abstract

Owing to rising intentional events, supply chain disruptions have been considered by setting up a game between two players, namely, a designer and an interdictor contesting on minimizing and maximizing total cost, respectively. The previous studies have found the equilibrium solution by taking transportation, penalty and restoration cost into account. To contribute further, we examine how incorporation of inventory cost influences the players' strategies. Assuming risk-averse feature of the designer and fully optimizing property of the interdictor with limited budget, the conditional-value-at-risk is employed to be involved in total cost. Using special order sets of type two and duality role, the linearized tri-level problem is solved by column-and-constraint generation and benders decomposition algorithms in terms of small-sized instances. In terms of larger-sized instances, we also contribute to prior studies by hybridizing corresponding algorithms with bio-geography based optimization method. Another non-trivial extension of our work is to define adapted stochastic measures based on the proposed mean-risk tri-level formulation. Borrowing instances from prior papers, the computational results indicate the managerial insights on players' decisions, the model's efficiency and performance of the algorithms.

Keywords: Inventory-based protection problem; Tri-level Stackelberg game; Mean-risk formulation; Value of stochastic solution; Decomposition-based heuristic algorithm.

1. Introduction

Supply chain (SC) facilities are increasingly exposed to the massive intentional disruptions caused by malicious interdictors. From 2014 – 2016, the rate of intentional disruptions of the critical SC structures including agriculture, food, beverage, and pharmaceutical sectors alongside industrial and manufacturing materials have been doubled. To mitigate this, the designer of an SC structure should analyze the behavior and probable damages of a malicious interdictor by taking significant SC costs into account. The interdictor is a fully optimizing agent with a limited budget adopting his malicious strategies after analyzing and observing the decisions and precautions of the designer. Game theoretic approaches have been used to model this interactive environment (Yolmeh and Baykal-Gürsoy, 2017). The key point in a game between the designer and interdictor is to accurately consider and measure the prohibitive sources of SC costs. Another critical aspect of the game theory is related to the intractable computation for finding the equilibrium solution when the size of a problem grows. Here, the inventory term is added to the conventional cost components of the previous studies while the measurement is made through the risk-averse criterion. The efficiency of the proposed model is also assessed versus simpler counterparts. In terms of the solution methods, two decomposition algorithms are employed and further hybridized with a meta-heuristic algorithm to capture the corresponding intractability.

In detail, the game theory of the SC protection problem is mostly handled via a bi-level or tri-level programming to model the leader (designer) – follower (interdictor) Stackelberg game. Scaparra and Church (2008) formulated a specific variant of the SC protection problem, namely the *r*-interdiction median model, through bi-level programming. Their purpose was to protect a set of facilities in a way that the most detrimental attack over *r* unprotected facilities led to a minimum SC cost. Khanduzi and Sangaiah (2019) developed a new bi-level supply chain protection problem for a bio-medical network. On one hand, the defender looked for finding optimal protection decision for medical suppliers and distribution centers, assigning clients to medical devices and determining the throughput quantity between facilities. On the other hand, the attacker focused on reducing service quality by maximizing the capacity of medical devices. The *r*-interdiction median Stackelberg game is also used to be formulated in the context of a tri-level programming model using the min-max-min structure (Akbari-Jafarabadi et al., 2017; Liberatore et al., 2012). In fact, the first, the second and the third level

*Corresponding author Email address: m.seifbarghy@alzahra.ac.ir

problems were associated with the fortification, interdiction, and mitigation of the SC damages, respectively. In these *r*-interdiction median studies, the total SC cost includes the worst-case fixed, transportation and outsourcing expenditures.

Here, to constitute the game, tri-level programming is considered in order to minimize the total cost before and after interdiction. The framework of our study is close to the studies of Bricha and Nourelfath (2013) and Jalali et al. (2018). Bricha and Nourelfath (2013) formed a two-stage game between SC's designer and interdictor to determine the level of protective and interdicting investment on facilities. They discretely dealt with the investment options on facilities by defining the protective and interdicting efforts associated with the designer and interdictor, respectively. In this way, the failure probability of a facility was in relation to the corresponding invested protective and interdicting efforts. Concerning the simple fixed-charge location problem, the location and distribution configurations were initially set. Then, a set of possible scenarios of the disrupted facilities were characterized. Under different realization of the scenarios, the designer and interdictor were looking for minimizing and maximizing the expected SC costs while weighing against associated protective and interdicting expenditures, respectively. The SC costs were made up of the expected fixed, transportation, restoration, and penalty expenditures. In the proposed simple fixed-charge location problem, the allocation pattern was based on the assignment of customers to the nearest open facility. They also assumed a fully optimizing interdictor in which one not only is interested in disturbing the facilities but also looks for optimizing his expenses. Using almost the same configuration except for the fully optimizing feature of the interdictor, Jalali et al. (2018) devised the game in the context of bi-level programming. They measured the damage cost in terms of the conditional value-at-risk (CVaR) instead of the prior expectation criterion.

The aforementioned studies have failed to address the inventory costs with respect to the protection SC problem. In fact, in many cases when the ordering and holding expenditures of a product are not negligible, the inventory costs may make up a significant portion of the total SC cost (Chen et al., 2011). This fact can change the optimal transportation pattern of the classical location problem. This concern can even be exacerbated under serious SC disruptions (Shen and Li, 2016; Tang, 2006).

The most challenging issue of modeling the inventory costs is associated with its nonlinear formulation (Zheng et al., 2019). Lagrangian relaxation and piece-wise linearization methods are among the common practices (Chen et al., 2011; Diabat and Theodorou, 2015). For instance, Zhang et al. (2016) used the special order sets of type two (SOS2) in order to linearize the inventory cost function. They showed that the maximum error of approximating the inventory function with respect to the SOS2 was below 2%.

Concerning the protection of SC facilities under intentional disruptions, we incorporate the inventory costs, being approximated by SOS2, into the classical fixed-charge location problem. Thus, in our study, the total SC cost includes transportation, penalty, restoration, and inventory expenditures. Due to the inherent randomness of the SC costs as a result of the facilities' vulnerabilities, the total cost is characterized with regard to a mean-risk formulation using the conditional-value-at-risk. Previous studies such as Fan et al. (2020) also suggested the necessity to consider conditional-value-at-risk with respect to the Stackelberg game. However, the mean-risk formulation, being regarded as a stochastic programming model, is generally known to be computationally challenging. Alternatively, there are simpler approaches such as the worst-case or robust programming approaches for capturing the uncertainty of a stochastic mean-risk formulation. Noyan (2012) defined two stochastic measures to justify the efficiency of her proposed single-level mean-risk formulation versus some simpler approaches. In this paper, we adapt the corresponding stochastic measures in terms of assessing the performance of the proposed tri-level mean-risk formulation.

To solve the proposed multi-level SC protection problem, one should focus on the different previous methodologies such as meta-heuristic, heuristic, decomposition, and hybrid algorithms. Lin et al. (2019), Konak et al. (2015) and Kim et al. (2009) utilized meta-heuristic algorithms to deal with the bi-level models of protecting the railway and telecommunication networks. In terms of the heuristic approaches, Bricha & Nourelfath (2013, 2014, 2015) considered the backward induction heuristic method to solve the proposed two-stage game-theoretic models. The choice of Jalali et al. (2018) was the Benders decomposition (BD) algorithm in the case of small-sized instances. Meanwhile, in the context of a tri-level location-transportation problem, Zeng and Zhao (2013) theoretically proved that the column-and-constraint generation (CCG) method is less complex compared to the BD algorithm. In addition, An et al. (2014) demonstrated the effectiveness of CCG in terms of a bi-level two stage robust programming. The results indicated that the generated cuts of CCG were always stronger than those generated by the BD algorithm. Meanwhile, Du et al. (2020) concluded that depending on the size of instance, either BD or CCG performed the best. Ghavamifar et al. (2018), Mahmoodjanloo et al. (2016), Jalali et al. (2018) and Setak et al. (2019) improved the performance of their decomposition algorithms over the multi-level problems with the aid of compromise programming, bio-geography based optimization and the genetic algorithm.

A brief review of the relevant works in the literature and the place the current research takes in comparison with them is given in Table 1, where both CCG and BD algorithms are used to solve two small-sized instances. Then, to solve a medium-sized instance, a novel solution method is developed by hybridizing CCG, BD and BBO algorithms.

In the rest of the paper, theoretical and practical implications of the current study are illuminated in Section 2. Section 3 elucidates the model formulation whereas Section 4 discusses the solution methodologies. Section 5 assesses the efficiency of the proposed model formulation and solution algorithms. Finally, Section 6 provides a brief conclusion and a number of directions for future research.

Table 1

Concise literature review

Studies	No of levels [a]	Cost components [b]	Location problem [c]	Fully optimizing feature of interdictor [d]	Optimization approach [e]	Stochastic measures [f]	Methodologies [g]
Scaparra and Church (2008)	2L	T	M	-	WC	-	H
Kim et al. (2009)	2L	T	FC	-	EC	-	H
Liberatore et al. (2012)	3L	T,B	M	-	WC	-	E,H
Noyan (2012)	1L	F,T,B,I	FC	-	MR	+	E
Zeng and Zhao (2013)	1L	F,T,C	FC	-	WC	-	E
Bricha and Nourelfath (2013)	2L	F,T,R,B,P,A	FC	+	EC	-	H
An et al. (2014)	2L	T,B	M	-	WC	+	E,H
Bricha and Nourelfath (2014)	2L	F,T,R,B,P,A,C	FC	+	EC	-	H
Bricha and Nourelfath (2015)	2L	F,T,R,B,P,A,C	FC	+	EC	-	H
Konak et al. (2015)	2L	P,A	M	-	WC	-	A,H
Zhang et al. (2016)	1L	F,T,B,I	FC	-	EC	-	A,E
Akbari-Jafarabadi et al. (2017)	3L	F,T,B	M	-	WC	-	H
Mahmoodjanloo et al. (2016)	3L	T,B,P	M	-	WC	-	E,H
Ghavamifar et al. (2018)	2L	F,T,B,I	FC	-	EC	-	E,H
Jalali et al. (2018)	2L	F,T,R,B,P,A	FC	-	MR	-	A,E,H
Our research	3L	F,T,R,B,P,A,I	FC	+	MR	+	A,E,H

[a] 1L, 2L and 3L indicate a single-level, bi-level and tri-level programming, respectively.
[b] F: Fixed, T: Transportation, B: Penalty, R: Restoration, P: Protection, A: Interdiction, C: Capacity acquisition, I: Inventory.
[c] FC and M refer to the Fixed-charge and Median location problems, respectively.
[d] It shows whether to consider the fully optimizing feature of the interdictor (+) or not (-).
[e] Optimization approach includes Expected cost (EC), Worst-case (WC) and Mean-risk (MR) formulations.
[f] +: Considering stochastic measures, -: Ignoring stochastic measures.
[g] Methodologies are categorized in terms of Exact (E), Heuristic (H) and Approximation (A) approaches.

2. Contributions and Practical Implications

This paper can be distinguished from the relevant studies available in the literature in terms of three significant aspects. First, despite the aforementioned studies, the current research embeds the inventory cost and its linearized approximation into the SC protection framework under intentional disruptions. Second, conventional stochastic measures are adapted for this work to justify the proposed tri-level mean-risk formulation. Third, a unique solution methodology is developed by hybridizing CCG, BD and BBO algorithms.

Through the quick review of Table 1, the differences between the current work and the other studies are highlighted based on the key features of the literature. In particular, our research extends Bricha and Nourelfath (2013) in terms of deeming the inventory costs, mean-risk formulation, stochastic measures, and a new hybridized solution method. In addition, the current research is different from Jalali et al. (2018) due to the considerations of the inventory costs, fully optimization feature of the interdictor, stochastic measures, and the corresponding hybridized method.

In terms of the practical implications, the consideration of the inventory costs makes the current work capable of handling a wider scope of real cases of intentional disruptions. Specifically, this paper is applicable to disaster relief and management context. In designing an SC structure of emergency responses, the pre-positioning of the emergency supplies is a key factor of increasing the pre-preparedness over an extreme event like natural disasters (Rawls and Turnquist, 2010). A significant portion of the emergency supplies includes a range of different facilities embodying vital commodities such as food (e.g. meals-ready-to-eat), water, and medical kits. These commodities, in fact, require delicate inventory management. On the other hand, following natural disasters, the incidence rate of intentional disruptions is considerably increased (Berrebi and Ostwald, 2013; Paul and Bagchi, 2018). In this circumstance, one of the possible targets for an interdictor is to concentrate on hitting the emergency SC structures. Despite such susceptibility, most of the previous studies are not apt cases of protecting the emergency SC facilities under intentional disruption owing to the ignorance of the inventory terms. In this spirit, compared to previous studies, the proposed SC protection framework considering the inventory costs is more consistent with the disaster preparedness problem under intentional disruptions.

3. Model Formulation

In this section, we first focus on deriving the associated costs including, transportation, penalty, restoration and the approximated inventory costs. Then, the game-theoretic mathematical formulation is proposed in terms of risk-averse tri-level programming. Henceforward, in our notations, the vectors and solutions of the decision variables are represented by the boldface and the bar sign symbols, respectively.

3.1. Modeling the associated costs

The basic framework involved in the proposed game-theoretic SC inventory-protection formulation under intuitional facility disruptions is formed in the context of the fixed-charge location problem. There are pre-defined discrete sets of facilities J (indexed by j) with infinite capacities and customers I (indexed by i) with finite demands (h_i). The un-capacitated facilities can be opened with the fixed cost of f_j while they can serve customers with the unit transportation cost of ρ_{ij}. There are also a number of finite scenarios to demonstrate whether each facility is operative or failed under intentional disruptions. We denote S as the set of scenarios (indexed by s) in which the failed and operative facilities within the scenario s are defined by the sets F_s and $\acute{F}_S$, respectively. For instance, the scenario $S = (1,1,0)$ demonstrates that facilities j_1 and j_2 are operative while facility j_3 is failed (e.g. $\acute{F}_s = [j_1, j_2], F_s = [j_3]$). Furthermore, ℓ_j^s is an indicator showing operative (if equal to one) and failed (if equal to zero) facilities while π is the penalty cost of backordering a demand. The restoration cost of a failed facility is also set at R_j.

Similar to Bricha and Nourelfath (2013), an existing SC structure is considered in which the location of the facilities has been previously determined by the simple classical fixed-charge location problem. Then, the allocation (Y_{ij}^s) and the penalty decisions (O_i^s) are considered as the scenario-dependent variables in which they are distinctively optimized in terms of a scenario. To incorporate the inventory costs, the popular economic order quantity approach is followed. The inventory costs can be formulated by specifying the ordering (A_j), holding (l_j) and purchasing costs (C_j). Assuming a negligible lead time to procure goods, the inventory cost of each facility in a scenario can be derived by $\sqrt{\sum_{i\in I} 2A_j l_j h_i Y_{ij}^s} + \sum_{i\in I} C_j h_i Y_{ij}^s$ (Chen et al., 2011). In our study, this non-linear inventory term is approximated by the linear SOS2 approach proposed in Zhang et al. (2016). In what follows, the associated scenario-based costs are derived and aggregated by the term Λ^s. In the following linearized formulation, $\bar{X} = [\bar{X}_J, \forall j \in J]$ is the solution vector of the opened facilities based on the simple classical fixed-charge location problem.

$$\underset{\forall s\in S}{Min}\ \Lambda^s = \sum_{j\in J}\left(\sqrt{2A_j l_j}\left[\sum_{\phi\in\Phi}\sqrt{T_{\phi j}^s}\kappa_{\phi j}^s\right] + \sum_{i\in I} C_j h_i Y_{ij}^s\right) + \sum_{j\in J}\sum_{i\in I} h_i \rho_{ij} Y_{ij}^s + \sum_{i\in I}\pi O_i^s + \sum_{j\in J}(1-\ell_j^s)\overline{X_j}R_j \quad (1)$$

$$\text{s.t: } \sum_{i\in I} h_i Y_{ij}^s = T_{1j}^s\kappa_{1j}^s + \cdots + T_{|\Phi|j}^s\kappa_{|\Phi|j}^s = \sum_{\phi\in\Phi} T_{\phi j}^s \kappa_{\phi j}^s \quad \forall s\in S,\ j\in J \quad (2)$$

$$\sum_{\phi\in\Phi}\kappa_{\phi j}^s = 1 \quad \forall s\in S,\ j\in J \quad (3)$$

$$Y_{ij}^s \le \ell_j^s \overline{X_j} \quad \forall i\in I, j\in J, s\in S \quad (4)$$

$$\sum_{j\in J} Y_{ij}^s + O_i^s \ge 1 \quad \forall i\in I, s\in S \quad (5)$$

$$\kappa_{\phi j}^s \in \{\text{SOS2}|\kappa_{\phi j}^s \ge 0\},\ \forall s\in S,\ j\in J, \phi\in\Phi \quad (6)$$

$$Y_{ij}^s \ge 0\ \forall i\in I, j\in J,\ s\in S;\quad O_s^i \ge 0\ \forall i\in I,\ s\in S \quad (7)$$

Relation (1) is the summation of the inventory, transportation, penalty and restoration costs, respectively. In the approximated inventory costs, the term $\sum_{\phi\in\Phi}\sqrt{T^s_{\phi j}}\kappa^s_{\phi j}$ is substituted for the non-linear term $\sqrt{\sum_{i\in I}h_iY^s_{ij}}$ using the SOS2 approach. The SOS2 is a set of consecutive variables in which not more than two adjacent elements are non-zero. In terms of any scenario and facility, assume that $\sum_{i\in I}h_iY^s_{ij}\in[T^s_{1j},T^s_{|\Phi|j}]$. Then, $\sum_{i\in I}\square_iY^s_{ij}$ can be obtained by a convex combination of the corresponding two breakpoints (T^s_{1j} and $T^s_{|\Phi|j}$) within the interval. This is exactly enforced by Relations (2), (3) where ϕ is the index of the breakpoints and $\kappa^s_{\phi j}$ is the relative ϕth SOS2 variable. Additionally, Relations (4) and (5) revisit the assignment pattern based on the realization of a specific scenario. Finally, Relations (6) and (7) restrict the decision variables.

3.2. The risk-averse game-theoretic tri-level programming

Formulation (8) – (17) includes the Stackelberg game between the designer and interdictor being translated into tri-level programming of a designer-interdictor-designer's model. We introduce the notations in advance to devise the game. It is assumed that the strategy of the designer (interdictor) is a combination of attributing different defensive (interdicting types) to any opened facility. Let D and A denote the sets of defensive and interdicting types indexed by d and a, respectively. In terms of each defensive (interdicting) type associated with a specific facility, there is an effort unit b_{jd} (q_{ja}) with cost of b'_{jd} (q'_{ja}). In addition, the set of protection strategies is represented by P (indexed by p) and the associated cost is given by B_p. The set Pcontains all permutations of using defensive types of fortifying facilities. For instance, a specific protection strategy $p_0=(d_1,d_3,d_1)$ shows that the three opened facilities are fortified by the defensive types d_1, d_3 and d_1, respectively. This special strategy uses the efforts b_{1d_1}, b_{2d_3} and $b_{3d_1}$to fortify the three opened facilities requiring the total protection cost of $B_{p_0}=b'_{1d_1}+b'_{2d_3}+b'_{3d_1}$. In the same vein, the set of malicious strategies of the interdictor is defined by M (indexed by m) with the associated cost of Q_m.

Additionally, the interdictor has a limited budget AC for carrying out a malicious strategy. The decision variable associated with the designer (interdictor) is λ_p (μ_m) expressing whether to choose p (m) or not (1 choose; 0 otherwise). In this regard, the optimal protection (malicious) strategy of the final solution is denoted by $p^{opt}(m^{opt})$. Furthermore, $Pr^{s_{p,m}}$ is the failure probability of facilities being calculated by $\prod_{j\in F_s}\left(\frac{q_{ja}}{q_{ja}+b_{jd}}\right)\prod_{j\in F'_s}\left(1-\frac{q_{ja}}{q_{ja}+b_{jd}}\right)$. It is primarily proportional to the efforts of interdicting and protecting a specific facility based on a particular scenario realization. Notice that, this type of calculating the failure probability is called as the contest success function being exactly borrowed from the rent seeking theory (Heijnen and Schoonbeek, 2019). To incorporate the risk measure, let us also introduce θ as the weight and δ as the confidence level of the risk measure. The decision variables associated with the risk measure are value-at-risk under any p (η_p) and excess value beyond the value-at-risk with respect to any p and s (v^s_p). In what follows, we discuss the interactions between the designer and interdictor via the first-level (8) – (10), second-level (11) – (14) and third-level (15) – (17) problems.

$$\underset{\boldsymbol{\lambda}}{Min}\ \ \tau=\sum_{p\in P}B_p\lambda_p+\tau' \tag{8}$$

$$\text{s.t: }\sum_{p\in P}\lambda_p=1 \tag{9}$$

$$\lambda_p\in\{0,1\}\ \forall p\in P \tag{10}$$

$$\underset{\boldsymbol{\mu}}{Max}\ \ \tau'=-\sum_{m\in M}Q_m\mu_m+\tau'' \tag{11}$$

$$\text{s.t: }\sum_{m\in M}\mu_m=1 \tag{12}$$

$$\sum_{m\in M}Q_m\mu_m\le AC \tag{13}$$

$$\mu_m\in\{0,1\}\ \ \forall m\in M \tag{14}$$

$$\underset{\boldsymbol{\eta,v}}{Min}\ \tau''=\sum_{m\in M}\sum_{p\in P}\sum_{s\in S}\mu_m\lambda_p\,Pr^{s_{p,m}}\,\overline{\Lambda^s}+\ \theta\sum_{m\in M}\sum_{p\in P}\mu_m[\eta_p+\frac{1}{1-\delta}\sum_{s\in S}Pr^{s_{p,m}}\,v^s_p] \tag{15}$$

$$\text{s.t: }\ v^s_p\ge\lambda_p\overline{\Lambda^s}-\eta_p\ \ \ \forall s\in S,p\in P \tag{16}$$

$$\eta_p\in R\ \ \forall p\in P;\ \ v^s_p\ge 0\ \ \forall s\in S,p\in P \tag{17}$$

Concerning Relation (8), the first-level problem minimizes the investment cost on protecting facilities as well as the maximum loss of the post intentional disruption. Doing so, the optimal protection decision is adopted by respecting to use exactly one strategy in Relation (9) and binary constraint in Relation (10). On the other hand, the interdictor focuses on maximizing the post disruption cost by adopting the most economic malicious strategy based on Relation (11). The interdictor is also forced to adopt a single malicious strategy and to respect the budget and binary constraints in Relations (12) – (14), respectively. The third-level problem measures the post disruption costs and determines the risk-based variables under specific protection and malicious strategies. Relation (15) minimizes the mean-risk cost (denoted as the post disruption cost). It includes the expected scenario-based costs ($\overline{\Lambda^s}$) plus the CVaR cost under corresponding protection and malicious strategies. Relation (16) determines the value of the excess variables that are beyond the value-at-risk. Finally, Relation (17) marks the types of risk-based variables.

4. Solution Procedure

Prior to solving the model, it is reformulated by the procedure explained in Section 4.1. In Section 4.2, the CCG and BD algorithms are customized based on the reformulated model. To further enhance the solution method for tackling larger-sized instances, a hybrid algorithm is developed in Section 4.3 by embedding the decomposition algorithms with BBO.

4.1. Reformulation

The proposed mixed-integer non-linear tri-level programming is not solvable, in its current formulation, by the popular decomposition algorithms. To remedy this, the formulation is linearized and reduced to a mixed-integer bi-level programming using the dual counterpart of the third-level problem. Let denote the second and the third-level programming as the sub-problem. For a given solution $\bar{\boldsymbol{\lambda}}$ to the first-level problem, the sub-problem is a max-min formulation. If the minimization part of the sub-problem (associated with the third-level programming) is reformulated by its dual form, there is a max-max formulation which can be regarded as a coherent maximization model. To do so, let represent the vector of dual multipliers in Relation (16) by $\boldsymbol{\beta}$. The resultant dual-sub problem (e.g. dual third-level problem being merged into the second-level problem) is given next. Note that, Relations (15) – (17) are repeated for the sake of tractability. This leads us to

$$\underset{\boldsymbol{\mu},\boldsymbol{\beta}}{Max} \sum_{m\in M}\sum_{p\in P}\sum_{s\in S}\mu_m\overline{\lambda_p}\,Pr^{s_{p,m}}\,\overline{\Lambda^s}+\sum_{s\in S}\sum_{p\in P}\beta_p^s\,\overline{\lambda_p\Lambda^s}-\sum_{m\in M}Q_m\mu_m \tag{18}$$

$$\text{s.t: } \beta_p^s\le\frac{\theta}{1-\delta}\sum_{m\in M}\mu_m\,Pr^{s_{p,m}}\;\forall s\in S,p\in P \tag{19}$$

$$\sum_{s\in S}\beta_p^s=\theta\forall p\in P \tag{20}$$

$$\beta_p^s\ge 0\quad\forall s\in S,p\in P \tag{21}$$

$$\sum_{m\in M}\mu_m=1;\quad\sum_{m\in M}Q_m\mu_m\le AC;\quad\mu_m\in\{0,1\}\quad\forall m\in M \tag{15 – 17}$$

Relation (18) is the sum of the dual terms of the third-level programming minus the malicious cost of the second-level programming. Relations (19) and (20) are constructed based on the risk-based variables ($\boldsymbol{\nu},\boldsymbol{\eta}$) whereas Relation (21) enforces positive values of the dual multipliers. Relations (15) – (17) are also directly borrowed from the second-level programming. Now, there is a mixed-integer bi-level programming being composed of the master problem (Karamyar et al., 2018) and the resultant dual-sub problem which can be solved by the decomposition algorithms discussed in the next subsection.

4.2. Decomposition algorithms

The CCG is customized and implemented according to the steps of Fig. 1.

Step 0) Initialization:

- Set $LB=-\infty,\ UB=+\infty,\ T=0$.
- Set an initial feasible $\bar{\boldsymbol{\lambda}}$.

Step 1) Sub-problem:

- Solve the dual sub-problem.
- Update $T\leftarrow T+1,\bar{\boldsymbol{\mu}}^T\leftarrow\bar{\boldsymbol{\mu}},\bar{\boldsymbol{\beta}}^T\leftarrow\bar{\boldsymbol{\beta}},UB\leftarrow\bar{\tau}'+\sum_{p\in P}B_p\bar{\lambda}_p$.
- Create new risk-based variables$\boldsymbol{\nu}^T,\boldsymbol{\eta}^T$.

Step 2) Master problem

- Solve the following master problem.

$$Min\quad\tau=\sum_{p\in P}B_p\lambda_p+\tau' \tag{8}$$

$$\text{s.t: }\tau'\ge-\sum_{m\in M}Q_m\bar{\mu}^t{}_m+\sum_{m\in M}\sum_{p\in P}\sum_{s\in S}\bar{\mu}^t{}_m\lambda_p\,Pr^{s_{p,m}}\,\overline{\Lambda^s}$$

$$+\,\theta\sum_{m\in M}\sum_{p\in P}\bar{\mu}^t{}_m\left\{\left(\eta_p^t+\frac{1}{1-\delta}\sum_{s\in S}Pr^{s_{p,m}}\,\nu_p^{st}\right)\right\}\quad\forall t\in\{1,\dots,T\} \tag{22}$$

$$\nu_p^{st}\ge\lambda_p\overline{\Lambda^s}-\eta_p^t\quad\forall s\in S,p\in P,t\in\{1,\dots,T\} \tag{23}$$

$$\tau'\in\mathbb{R},\eta_p^t\in\mathbb{R}\quad\forall p\in P;\quad\nu_p^{st}\ge 0\quad\forall s\in S,p\in P,t\in\{1,\dots,T\} \tag{24}$$

$$\sum_{p\in P}\lambda_p=1;\;\lambda_p\in\{0,1\}\;\forall p\in P \tag{11 – 12}$$

- Update vector $\bar{\boldsymbol{\lambda}}$ and $LB\leftarrow\tau$.

Step 3) Stopping criterion:

- If $\frac{UB-LB}{LB}<\varepsilon$, then stop. Otherwise, repeat **Steps** (1) and (2).

Fig.1. Column-and-constraint generation algorithm

The CCG is initialized by setting the lower-bound (*LB*), upper-bound (*UB*) and iteration counter (*T*). It is also required to set an initial feasible solution to the first-level programming, namely $\bar{\boldsymbol{\lambda}}$. Next, the dual sub-problem is solved with respect to the incumbent protection strategy. Doing so, the iteration counter, optimal malicious strategy, and LB are updated. The subsequent step deals with the master problem. In each iteration, the first-level problem is solved with respect to the new constraints associated with Relations (22) – (23) and new risk-based decisions in Relation (24) stemming from the dual sub-problem. Using the incumbent malicious strategy, the master problem iteratively updates the optimal protection strategy, risk-based decisions, and *UB*. When the relative gap between *LB* and *UB* reaches to a negligible amount ε, the algorithm is stopped and the incumbent solutions are set as the optimal ones.

Note that, in contrast to CCG, the master problem of the BD algorithm only iteratively constructs Relation (25) while ignoring Relations (22) – (24).

$$\tau' \geq -\sum_{m\in M} Q_m \bar{\mu}^t{}_m + \sum_{m\in M}\sum_{p\in P}\sum_{s\in S}\bar{\mu}^t{}_m \lambda_p Pr^{s_{p,m}}\overline{\Lambda^s} + \sum_{s\in S}\sum_{p\in P}\bar{\beta}_p^{st}\lambda_p\overline{\Lambda^s} \quad \forall\ t\in\{1,\dots,T\} \qquad (25)$$

4.3. A hybrid solution method

The main reason for embedding a heuristic algorithm in the proposed decomposition approaches is to convert the extremely large sets of the protection and malicious strategies into workable sizes. Doing so, a population-based algorithm inspiring from the distribution of species over space and time called the biogeography-based optimization (BBO) is employed. BBO mainly evolves the population of chromosomes by the migration and mutation operators. The migration operator is akin to the crossover operation taking the immigration and emigration rates of habitats (i.e. chromosomes) into account. The immigration and emigration rates of a chromosome are inversely ($\phi_i = 1 - \frac{fit_i}{fit_{max}}$) and directly ($\varphi_i = \frac{fit_i}{fit_{max}}$) proportional to their fitness function (*fit*), respectively. The fitness function of a chromosome is, in fact, analogous to the suitability index of a habitat. The mutation operator, on the other hand, is to simulate cataclysmic changes of a habitat feature (i.e. genes). The mutation rate is also adjustable with respect to $\Phi_i = 1 - \frac{p_i}{p_{max}}$ where p_i shows the likelihood of a chromosome to be a solution to the problem (for detail discussion on computing the probability see Salehi and Masoumi (2019)). Notably, in order to normalize the fitness function and probability of chromosomes, they are divided by their maximum values namely, fit_{max} and p_{max}, respectively.

Now, there are two populations in the hybridized algorithm either of which with the size of Δ chromosomes. In the same vein of Konak et al. (2015) besides Jalali et al. (2018), one population is attributed to the designer's decisions (PD) while the other one is to consider the interdictor's decisions (PA). Each chromosome PD ($\omega_p \in PD$) /PA ($\varpi_m \in PA$) is an array of defensive/interdicting type as size as the number of facilities. For instance, consider the representation of $< \omega_1 = [3,1]\ |\omega_1 \in PD >$ demonstrating the first chromosome of the population PD (ω_1) related to a protection strategy. This representation indicates that the first and second available facilities are protected by the defensive types d_3 and d_1, respectively. In this way, the mathematical model is reduced by considering the new protection and malicious sets (namely, the population of PD/PA instead of the set of P/M). Then, the reduced model is solved via CCG and/or BD. The selection of the solution algorithm is affected by Θ_{update} (entitled as the switch coefficient) which is iteratively updated in decreasing order. The initial value of the switch coefficient is also denoted by Θ_0. As the number of iterations increases, it is more probable that the solution method switches from BD to CCG. Notably, in addition to the final solution, the inner upper-level solutions of the corresponding algorithm are stored. To do so, the matrix $\Upsilon(\omega_p|\varpi_m)$ shows the stored upper-level objective value in terms of the protection strategy ω_p and malicious strategy ϖ_m. Taking the resultant matrix Υ into account, the fitness value of ω_p/ϖ_m is scored based on its ability to dominate other protection/malicious strategies under any malicious/protection chromosome.

The dominance rule is as follows. Let ω_p dominates $\omega_{p'}$ (under given ϖ_m) if and only if the stored upper-level objective value ω_p is lower than the corresponding value $\omega_{p'}$ (i.e. $\omega_p \succ \omega_{p'} \Leftrightarrow \Upsilon(\omega_p|\varpi_m) < \Upsilon(\omega_{p'}|\varpi_m)$). Further, the dominant malicious chromosome is checked according to $\varpi_m \succ \varpi_{m'} \Leftrightarrow \Upsilon(\omega_p|\varpi_m) > \Upsilon(\omega_p|\varpi_{m'})$. The evolutions of the populations are accomplished via the migration and mutation operators. The roulette wheel selection and the multi-point preservative crossover (MPX) are also utilized within the migration operator. The algorithm is stopped when the required number of iterations (IT) is met. The pseudo code of Fig.2 clarifies the main skeleton of the proposed hybridized method and specifies the procedure of the related functions.

Initialization:
Randomly initialize PA and PD populations.
Set $\Theta_{update} = \Theta_0$, $it = 0$.
Main body:
While $it \leq IT$
If rand < Θ_{update} then run BD, else run CCG.
Store inner upper-level solutions and final solution in the matrix Υ.
For any $\omega_p/\varpi_m \in PD/PA$ calculate $fit(\omega_p)/fit(\varpi_m)$.
Remove the worst Δ chromosomes from PD/PA.

```
    If $it \le IT$ then
        For any $\omega_p \in PD$ call the BBO procedure.
        (Repeat the same procedure for any $\varpi_m \in PA$).
        Calculate BBO components: $\phi_{\omega_p}, \varphi_{\omega_p}, \Phi_{\omega_p}$.
        [if rand < $\phi_{\omega_p}$ → select $\omega_p$
        [apply Roulette wheel on $\boldsymbol{\varphi}$ → select $\omega_{p'}$   ⇒ MPX($\omega_p, \omega_{p'}$)
        If rand < $\Phi_{\omega_p}$ → select $\omega_p$ ⇒ Mutate($\omega_p$)
        $it = it + 1$
        $\Theta_{update} = (\Theta_0)^{it}$
    Else Stop While loop.
End While.
Related Function:
Function $fit(\omega_p)$                            Function $fit(\varpi_m)$
    Set $u \leftarrow \omega_p$                             Set $u \leftarrow \varpi_m$
    Set $\iota = 0$                                  Set $\iota = 0$
    For any $\omega_p \in PD$                           For any $\varpi_m \in PA$
        For any $\varpi_m \in PA$                           For any $\omega_p \in PD$
        If $\Upsilon(u|\varpi_m) \le \Upsilon(\omega_p|\varpi_m) \Rightarrow \iota = \iota + 1$     If $\Upsilon(\omega_p|u) \ge \Upsilon(\omega_p|\varpi_m) \Rightarrow \iota = \iota + 1$
        End For                                   End For
    Set $fit(\omega_p) \leftarrow \iota$                      Set $fit(\varpi_m) \leftarrow \iota$
    End for                                   End for
End Function                                  End Function
Function Roulette wheel ($\boldsymbol{\varphi}$)
    For each chromosome ($k$)
            calculate the interval $int_k = \left(\frac{\sum_{i=1}^{k-1}\varphi_i}{\sum_{i=1}^{end}\varphi_i}, \frac{\sum_{i=1}^{k}\varphi_i}{\sum_{i=1}^{end}\varphi_i}\right)$
    End For
    If $rand \in int_k$ ⇒Select $kth$ chromosome
End Function
Function MPX($\Theta_u, \Theta_v$)
    Generate a binary array ($\boldsymbol{\xi}$) as size as the number of facilities
    For any $j \in J$
        If $\xi_j = 0$ then $offspring(j) \leftarrow \Theta_u(j)$ else $offspring(j) \leftarrow \Theta_v(j)$
End Function
Function Mutate ($\Theta$)
    $offspring \leftarrow \Theta$
    Select a random gene of $\Theta$ namely, $k$th gene
    Select a random feasible value for the $k$th gene denoted as val
    $offspring(k) \leftarrow$ val
End Function
```

Fig. 2. Pseudo code of the hybrid method

5. Computational Results

The computational results begin with the introduction of the numerical instances in Section 5.1. Then, the impact of incorporating the inventory costs on the optimal protection and malicious configurations are assessed in Section 5.2. In Section 5.3, the next purpose is to justify the accuracy of approximating the inventory costs through SOS2. Section 5.4 measures the effectiveness of utilizing the mean-risk formulation while Section 5.5 compares the performance of the proposed decomposition algorithms over small-sized instances. In terms of medium-sized instances, the performance of the hybrid algorithm is evaluated in Section 5.6. The proposed algorithms are implemented using the ILOG Concert Technology for Java.

5.1. Instances

The computational results are based on two categories of instances. In this regard, two small-sized instances are adopted from Bricha and Nourelfath (2013) and Bricha and Nourelfath (2014) being denoted as S1 and S2, respectively. There is also a medium-sized instance (denoted by M1) being borrowed from Jalali et al. (2018). The key point in the size of the instances is their relation to the number of potential facilities. In fact, an increase in the potential facilities is analogous to an exponential growth of the number of scenarios. In this way, S1, S2, and M1 include 3, 5 and 10 potential facilities, respectively. The relevant data of these instances are summarized in Table A1 of Appendix A.

5.2. The managerial insight on the impact of considering the inventory costs

In this section, we demonstrate how the configuration of an SC protection problem may be influenced by the inventory costs. Without loss of generality, we focus on the S1 instance by setting $\theta = 10, \delta = 0.9, AC = 14100$. Substituting the optimal solution of a free-inventory case into the proposed inventory-based formulation, the corresponding output is consequently compared with a local optimum solution to the inventory-based S1 instance. Notice that, for the sake of comparison, instead of finding the optimal solution of the approximated version, a local optimum point is obtained via a trial and error method.

The results of Table 2 show that the inventory-based setting has a lower total cost compared to its free-inventory counterpart. The significant portion of such a priority is due to the inventory cost. The inventory cost of the inventory-based setting is 54,540 units lower than the one in the free-inventory case. However, the sum of expected transportation, penalty and restoration costs of the inventory-based setting exceeds the ones of the free-inventory one. Although the fixed protection and interdiction costs are the same for these two cases, the configurations of the malicious strategies are not similar. According to the content given in Appendix A, the second (a_2) and the third (a_3) levels of the interdicting types are associated with 250 and 150 effort units, respectively. In the inventory-based setting, the higher the interdicting effort (i.e. a_2) is imposed on the first facility while in the free-inventory case, a_2 is exerted on the second facility. This primarily stems from the changes in the transportation pattern of the facilities as a result of involving the inventory costs leading to the variations on the percent of satisfied demands by facilities.

Table 2
Comparing the free-inventory and inventory-based responses of S1 instance

Solution type	Resulted strategies		Fixed Investment costs		Expected costs				CVaR cost	Total cost
	p^{opt}	m^{opt}	Protection	Interdiction	Transportation	Penalty	Restoration	Inventory		
Free-inventory	$d_2d_2d_2$	$a_3a_2a_3$	466200	14100	3,993,380	5,326,677	10,854	82,806,470	1,301,601,000	1,394,205,000
Inventory-based	$d_2d_2d_2$	$a_2a_3a_3$	466200	14100	4,045,294	5,326,677	10,712	82,751,930	1,301,539,000	1,394,140,000

Concerning the free-inventory and the inventory-based cases, Fig. 3 compares the percent of satisfied demands over all possible failure scenarios. In this regard, the most considerable variation occurs within the first scenario (s_1). Under the inventory-based setting s_1 , Facilities 1 and 2 serve 26% higher and lower demands than the ones in the free-inventory setting, respectively. On the other hand, s_1 is the most likely scenario with the occurrence probability equal to 0.44 using the related formula described in Section 3. Due to this significant occurrence probability, the fully optimizing interdictor focuses on strengthening the malicious effort level of Facility 1 to disrupt the functionality of the SC structure at most. In order to afford the interdicting budget, the interdictor has to reduce the effort level of Facility 2 and exert the highest available effort unit on Facility 1 (i.e. a_2). Accordingly, m^{opt} changes from $a_3a_2a_3$ (associated with the free-inventory setting) to $a_2a_3a_3$ (in relation to the inventory-based case).

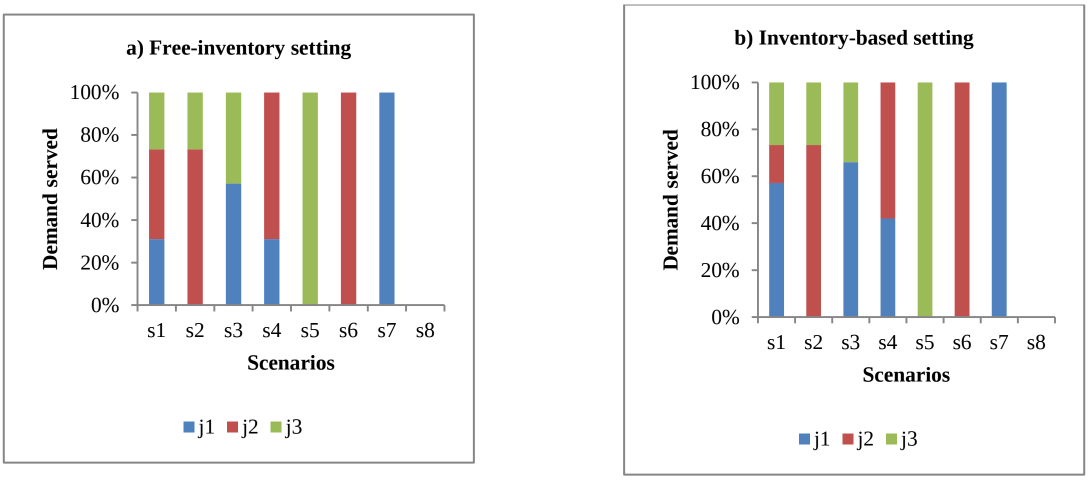

Fig. 3. Fraction of demand served by each facility concerning free-inventory (a) and inventory-based (b) cases

5.3. Accuracy of SOS2 approximation

To justify how well SOS2 estimates the expected inventory costs, the approximate costs are compared versus the actual in terms of all 1024 scenario solutions in the M1 instance. Based on the resultant allocation solution, the actual costs are, in fact, the direct square roots of the inventory costs. In Fig 4, the actual values are represented over the ideal continuous line while the approximate values are the scatter points. By computing the relative difference between the actual and approximate values, the mean (6.82%) and the maximum (12.976%) results advocate the accuracy of the SOS2 approximation.

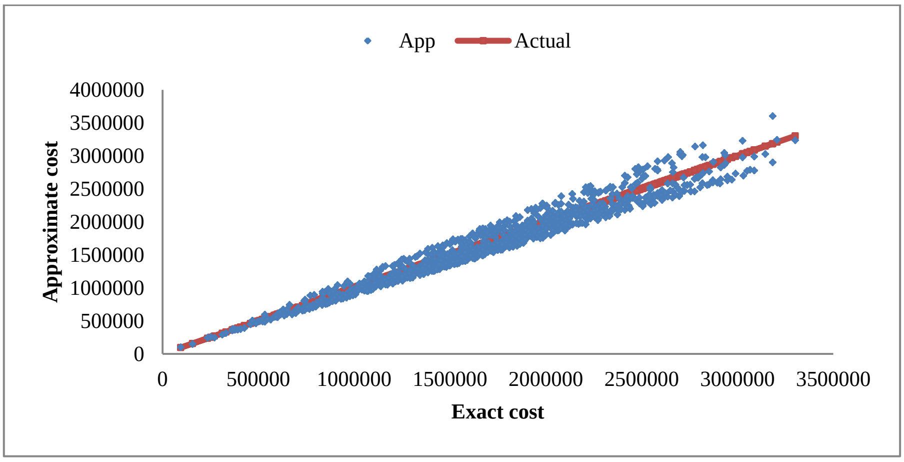

Fig.4. Approximate versus exact inventory costs

5.4. Stochastic measures

The proposed model formulation captures the vulnerability of facilities by means of one of the most well-known stochastic approaches, namely the scenario-based mean-risk programming. Nevertheless, this approach is not the unique solution for capturing the stochastic parameters and can be substituted by other alternatives such as the worst-case and the robust programming. The value of perfect information and the value of stochastic solution are well-known measures to judge whether it is worthy to utilize stochastic scenario-based programming versus other alternatives (Jalali et al., 2016). Noyan (2012) defined these measures based on a single-level mean-risk formulation. The mean-risk value of the perfect information (MRVPI) specifies the difference between the mean-risk solution to the scenario-based approach (MRS) and the mean-risk solution to the wait-and-see approach (MRWS). In this paper, the corresponding stochastic measures are adapted based on the proposed tri-level programming model with respect to the pseudo code of Fig 5.

In detail, MRS is related to the main formulation developed in this study, including Relations (8) – (17). On the other hand, the wait-and-see approach assumes that an exact forecasting system could determine which scenario will be realized in the future. Doing so, it distinctively solves the scenario-based model in terms of every single scenario. Then, the set of resultant solutions is aggregated through the mean-risk function. The mean-risk value of the stochastic solution (MRVSS) also compares the resultant solution to MRS with the mean-risk solution to the alternative worst-case programming (MRWC). The worst-case (or robust programming) considers a set of scenarios of interest (*SI*) instead of necessarily enumerating all scenarios (Peng et al., 2011). *SI*, in fact, includes the scenarios of the maximum k simultaneous failed facilities, namely $SI = [s \in S | \sum_{j\in J}(1 - \ell_j^s) \leq k]$. After solving the worst-case counterpart of the formulation, the resulted upper-level solution ($\bar{\boldsymbol{\lambda}}_{WC}$) is stored. Considering $\bar{\boldsymbol{\lambda}}_{WC}$, the lower-level problem is solved and aggregated through the mean-risk function. In what follows, the necessary steps of computing the stochastic measures are summarized.

MRVPI calculation:

For $s = 1: |S|$

Solve $[(8) - (15) | \theta = 0] \xrightarrow{\text{Store}} \begin{bmatrix} \Re(s) = \sum_{p\in P} B_p \overline{\lambda_p} - \sum_{m\in M} Q_m \overline{\mu_m} + \sum_{m\in M} \sum_{p\in P} \overline{\mu_m} \overline{\lambda_p} \overline{\Lambda^s} \\ p^{opt}(s) = p^{opt};\ m^{opt}(s) = m^{opt} \end{bmatrix}$

End For

Normalize probabilities $\Rightarrow n\,Pr^s = \dfrac{{Pr^s}_{p^{opt}(s), m^{opt}(s)}}{\sum_{s\in S} {Pr^s}_{p^{opt}(s), m^{opt}(s)}} \quad \forall s \in S$

$$MRWS = \sum_{s\in S} n\,Pr^s \;\; \Re(s) + \theta \underset{\substack{\eta\in\mathbb{R} \\ v^s\in\mathbb{R}^+\cup\{0\}}}{Min} \left[\eta + \frac{1}{1-\delta} \sum_{s\in S} n\,Pr^s \;\; v^s \,|\, v^s \geq \Re(s) - \eta \;\; \forall s \in S\right]$$

$MRVPI = MRS - MRWS$

***MRVSS* calculation:**

Consider new third-level programming based on the robust optimization

$$\underset{\omega\in\mathbb{R}}{Min}\ \tau'' = Min[\omega | \omega \geq \sum_{m\in M} \sum_{p\in P} \mu_m \lambda_p\, Pr^{s_{p,m}}\, \overline{\Lambda^s} \quad \forall s \in SI] \tag{26}$$

Solve $[(8) - (14)\ \&\ (26)] \xrightarrow{\text{Restore}} \bar{\boldsymbol{\lambda}}_{WC} = \bar{\boldsymbol{\lambda}}$

Solve $[(8) - (17) | \boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}}_{WC}] \xrightarrow{\text{Restore}} MRWC = \bar{\tau}$

$MRVSS = MRWC - MRS$

Fig. 5. Pseudo code of adapted stochastic measures

In Table 3, the stochastic measures are evaluated in terms of different values of the risk parameters. The results show that an increase in the value of δ and θ reduces the difference between *MRS* and *MRWS*. For instance, consider δ = 0.9. When comparing θ = 1 and θ = 50, it can be observed that the fraction of *MRVPI/MRS* is reduced from 31.65% to 11.60%. Furthermore, the positive values of *MRVSS* indicate that the adopted scenario-based mean-risk approach leads to the more desirable solutions in comparison with the less cumbersome worst-case method.

Table 3
Stochastic measures

δ	Θ	*MRS* ($\times 10^7$)	*MRWS* ($\times 10^7$)	*MRVPI* ($\times 10^7$)	*MRVPI/MRS*	*MRWC* ($\times 10^7$)	*MRVSS* ($\times 10^7$)	*MRVSS/MRS*
0.8	1	627.05	417.16	209.90	33.47%	628.21	1.16	0.18%
	10	4498.93	3794.66	704.27	15.65%	4505.45	6.52	0.14%
	50	21707.30	18805.90	2901.40	13.37%	21737.60	30.30	0.14%
0.9	1	682.33	466.36	215.97	31.65%	703.34	21.00	3.08%
	10	5080.00	4343.30	736.70	14.50%	5200.00	120.00	2.37%
	50	24800.00	21923.70	2876.30	11.60%	24900.00	29.70	0.12%

5.5. Convergence performance of the decomposition algorithms

Here, the computational performances of the three algorithms are compared together under different combinations of the risk parameters in S1 and S2 instances. *Algorithm1* belongs to the linearized single-level version of the main problem solved by the standard CPLEX approach. *Algorithm2* uses BD while *Algorithm3* solves the problem by CCG. The computational results in Table 4 prove the superior performance of the CCG over the other algorithms. It can be observed that the risk-neutral case is significantly a less challenging problem than the risk-averse cases when the problem size is increased. For instance, consider *Algorithm2*. Under S1, the CPU times of all the risk-neutral and risk-averse cases are close and below 10s. Nevertheless, under S2, the minimum CPU time of the risk-averse cases (573s) is significantly greater than the risk-neutral counterpart (80.557s). It is obvious that both *Algorithms* 2 and 3 are better than *Algorithm1* implying the significant importance of utilizing the decomposition algorithms to solve the multi-level stochastic programming. For the sake of comparison, the convergence plots of *Algorithms 2* and 3 within the given CPU times are provided in Fig 6. Both algorithms are converged in 10 seconds when they solve the S1 instance. Conversely, BD is unable to converge in 300 seconds over S2 while CCG reaches convergence after 275.48 seconds. This advocates the fact that the generated constraints of CCG are much stronger than the Benders-cut.

Table 4
Execution time of *Algorithm1*, *Algorithm2* and *Algorithm3* over small-sized instances

Instance	Risk parameters		CPU (s)		
	Θ	δ	*Algorithm1*	*Algorithm2*	*Algorithm3*
S1	0	-	114.00	8.84	8.94
	1	0.7	121.31	9.36	9.46
		0.8	127.50	9.69	9.18
		0.9	124.02	8.95	9.19
	5	0.7	149.61	9.17	9.21
		0.8	137.23	9.07	9.20
		0.9	118.10	9.16	8.99
	10	0.7	169.72	8.86	9.21
		0.8	142.59	9.19	8.97
		0.9	158.90	10.21	9.20
S2	0	-	960.18	80.56	86.97
	1	0.7	2460.16	656.80	201.00
		0.8	2100.53	573.00	164.64
		0.9	1620.80	580.14	153.67
	5	0.7	2160.89	672.10	279.70
		0.8	1920.23	619.40	161.50
		0.9	1740.02	632.52	139.42
	10	0.7	2460.07	691.87	187.80
		0.8	1440.97	645.07	150.57
		0.9	1380.73	592.05	275.48

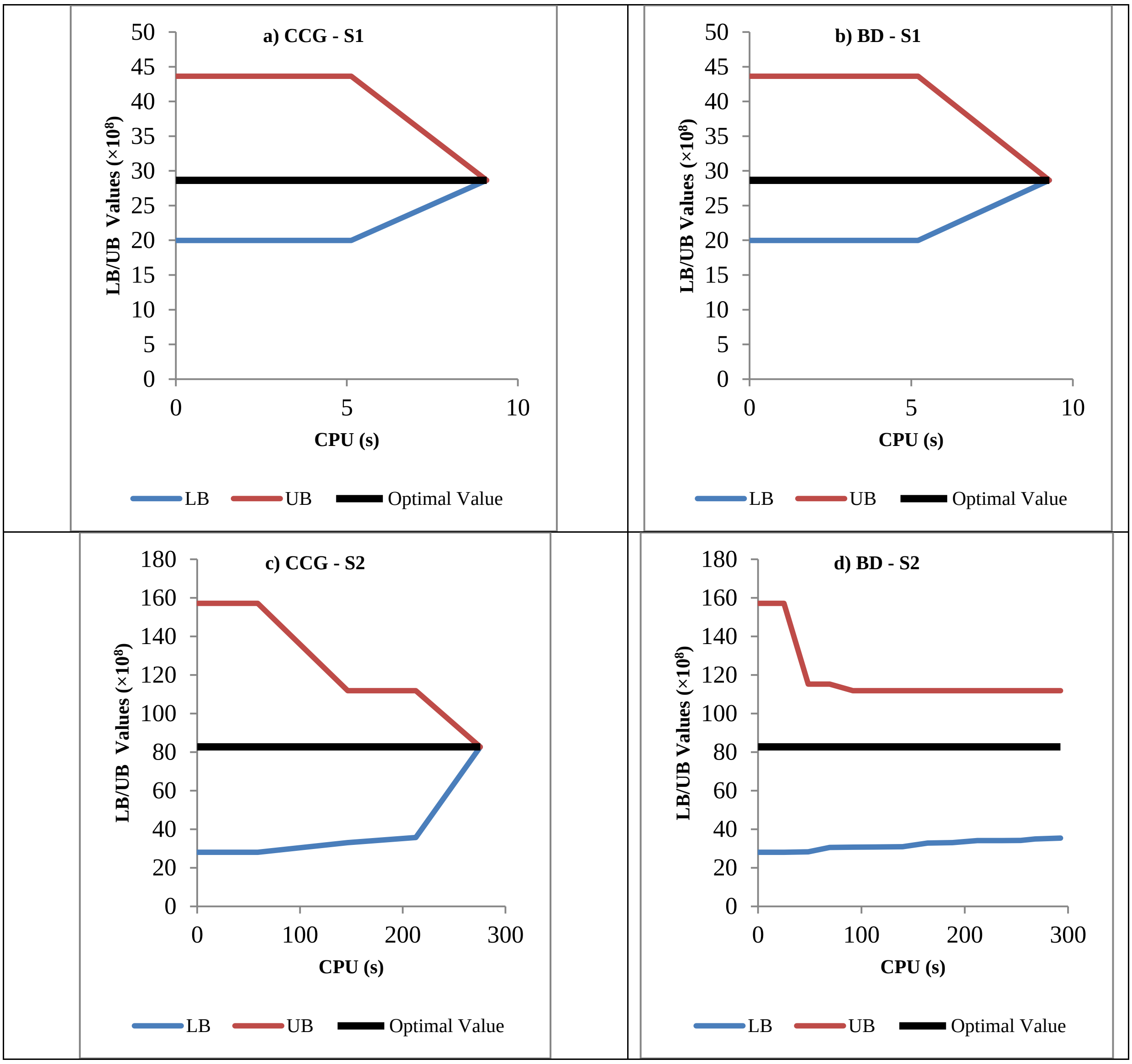

Fig. 6. LB/UB convergence over a limited time (a, b) 10s and (c, d) 300s under δ=0.9, θ=10.

5.6. Hybrid optimization

The hybrid optimization is implemented on the M1 instance with respect to the different configurations of the population size, initial value of the switch coefficient (Θ_0) and the risk parameters. The associated parameters are set as $= [40,100]$, $\Theta_0 = [0.2,0.6]$ and $\varepsilon = 0.1\%$.The results in Table 5 show that the hybrid algorithm provides higher quality solutions in terms of the greater values of Θ_0. At the same time, the algorithm demands less computational time based on the lower value of Θ_0. In fact, as the value of Θ_0 increases, the chance of selecting BD rises. On the other hand, the lower value of Θ_0 is analogous to the higher chance of CCG to be selected. Adopting from the results of the small-sized instances, CCG performs faster than BD with a lower number of iterations

Table 5
Numerical results for M1 instance using the hybrid method under $AC = 189000$

Δ	Θ_0	θ	δ	Cost ($\times 10^9$)	CPU time (s)
40	0.2	0	-	1.9740	1864
		1	0.75	6.0694	1948
			0.9	6.9078	1925
			0.95	7.4783	1936
		5	0.75	22.4510	1914
			0.9	26.6430	1879
			0.95	29.4957	1902

		10	0.75	42.9280	1909
			0.9	51.3120	1895
			0.95	57.0174	1909
	0.6	0	-	1.8478	2480
		1	0.75	5.8126	2604
			0.9	6.6317	2604
			0.95	7.2127	2487
		5	0.75	21.6716	2549
			0.9	25.7672	2483
			0.95	28.6724	2610
		10	0.75	41.4954	2538
			0.9	49.6867	2616
			0.95	55.4970	2619
100	0.2	0	-	1.7594	4878
		1	0.75	5.6311	4932
			0.9	6.4389	4937
			0.95	7.0289	4920
		5	0.75	21.1177	4881
			0.9	25.1567	4916
			0.95	28.1068	4938
		10	0.75	40.4760	4938
			0.9	48.5540	4884
			0.95	54.4541	4964
	0.6	0	-	1.7121	5596
		1	0.75	5.5323	5673
			0.9	6.3188	5708
			0.95	6.9310	5641
		5	0.75	20.8132	5622
			0.9	24.7455	5757
			0.95	27.8065	5640
		10	0.75	39.9143	5656
			0.9	47.7789	5734
			0.95	53.9008	5750

BD searches more inner solutions compared to CCG since it requires a greater number of iterations. Hence, the greater values of Θ_0 results in the higher quality solutions whereas the lower values of Θ_0 provides solutions with less computational time. Without loss of generality, we compare the results when $\Theta_0 = 0.2$ and $\Theta_0 = 0.6$ under $\Delta = 40$, $\theta = 5$, and $\delta = 0.9$. Although the resulted cost associated with $\Theta_0 = 0.6$ (i.e. $25.7672 * 10^9$) is less than its counterpart (i.e. $26.643 * 10^9$), the corresponding processing time demands 725 extra seconds to be completed. Notice that, we have ignored the results associated with a higher amount of Θ_0 such as ($\Theta_0 = 0.8$), since the required CPU time exceeded our pre-defined threshold, namely 6,000 seconds.

Another noticeable observation is in relation to the impact of the population size. The higher amount of Δ requires more computational time resulting in a better objective value. In fact, any observation related to $\Delta = 40$ requires lower CPU time but more prohibitive cost compared to the corresponding result associated with$\Delta = 100$. In the context of the computational times with respect to the different configurations of the risk parameters, no specific conclusion can be made except for the risk-neutral and risk-averse cases. In other words, concerning any value of $\Delta = 100$ and Θ_0, the risk-neutral cases (where $\theta = 0$) perform faster than the risk-averse cases (where $\theta > 0$). This is due to the extra complexity being required to derive the risk-dependent variables in terms of the risk-averse cases. However, one cannot claim that a specific risk-averse case (e.g. $\theta = 10$) demands more/less CPU time than another one (e.g. $\theta = 5$). Similarly, the fluctuations of the confidence level do not significantly influence the processing time. Let us compare the CPU times associate with $\delta = 0.9$ and $\delta = 0.95$ under $\Delta = 100$ and $\Theta_0 = 0.6$. When $\theta = 5$, the resulted CPU time of $\delta = 0.9$ (i.e. 5757**s**) is greater than the required CPU time of $\delta = 0.95$ (i.e. 5622**s**). Meanwhile under the same configuration with regard to θ = 10, the case related to $\delta = 0.9$ is completed 16**s** faster than the case of $\delta = 0.95$.

Fig 7 shows the performance of the hybrid method over 27 iterations of a specific case related to the M1 instance in details. The turbulent behavior stems from the gradual exploration of the new protection and malicious strategies over the search process of the hybrid method. In fact, the curve descends/ascends by exploring a new non-dominated protection/malicious strategy with respect to PD/PA, respectively. Precisely, at the beginning, the interaction between the protection and malicious strategies leads to a cost equal to 4.86×10^9. In the next iteration, the hybrid algorithm generates and adds a new malicious strategy into the PA list causing a more prohibitive cost equal to 6.49×10^9. In the seventh iteration (when the CPU time is around 466 seconds), the introduction of a prominent new protection strategy is responsible for bouncing back the cost to 4.43×10^9. This trend remains volatile until the algorithm proceeds to 6.07×10^9 after 1,948 seconds. Note that we use trial and error approach to tune parameters of algorithms in the same vein of SnyderKonak et al. (2015) and Jalali et al. (2018). Future research can improve algorithms by using statistical methods such as Taguchi method (Gan and Safaei, 2016; Rizvi and Wajahat, 2019; Sadeghi et al., 2018).

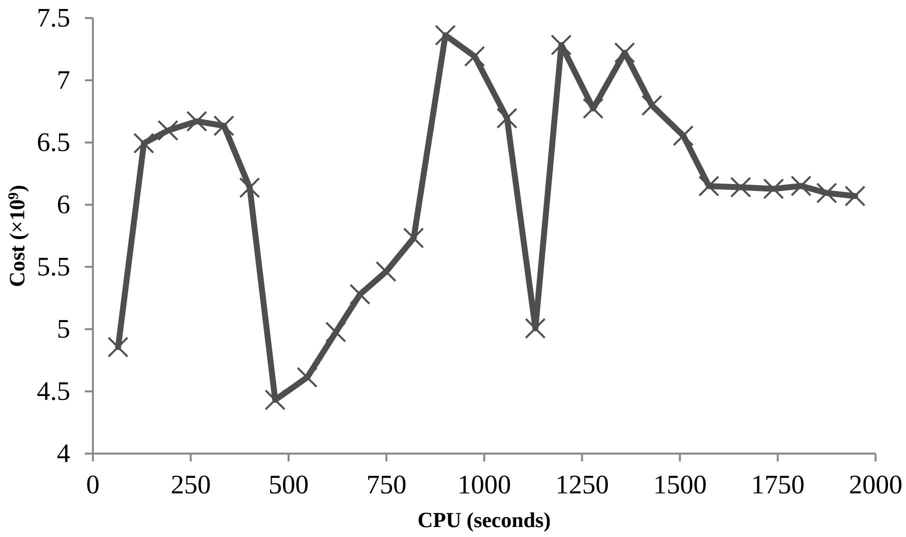

Fig.7. Detail performance of the hybrid method over M1 instance when $\Delta = 40, \Theta_0 = 0.2, \theta = 1, \delta = 0.75$

6. Conclusion

This paper developed a risk-averse supply chain (SC) protection problem to include inventory decisions. To solve the proposed problem, two algorithms, the column-and-constraint generation (CCG) and the Benders decomposition (BD) algorithms, were used. To improve the efficiency of algorithms, we hybridized them with a bio-geography based algorithm. The drawn managerial insight showed that inventory costs could revisit protection or malicious strategies by changing the fraction of demand served by the facilities. The adapted stochastic measures elucidated that the mean-risk formulation was more appropriate for capturing the vulnerability of facilities compared to the less cumbersome formulations such as worst-case programming. CCG also performed faster than BD algorithm due to the generation of the stronger cuts. The results of the hybrid approach demonstrated that the mixture of CCG and BD algorithms could preserve both aspects associated with the execution time and quality of the solutions. Moreover, the volatility associated with the convergence curve of the hybrid method was caused by the close contest between the designer and the interdictor. To extend the current study further, one can relax the assumption of considering an existing SC structure of opened facilities using the classical fixed-charge location problem. Doing so, a specific framework should be developed to optimize location, allocation, penalty, protection and malicious decisions simultaneously under the consideration of the inventory costs.

Appendix A. (Input data)

Table A1. Input data of S1, S2 and M1 instances

Instance	Sets and parameters
S1	$I = \{i_1, i_2, i_3, i_4, i_5\};\ J = \{j_1, j_2, j_3\};\ A = \{a_0, a_1, a_2, a_3\}; D = \{d_0, d_1, d_2, d_3\}; S = \{s_1, s_2, \ldots, s_8\}$
	$\boldsymbol{f} = \begin{pmatrix} 2{,}100{,}000 \\ 2{,}400{,}000 \\ 1{,}800{,}000 \end{pmatrix};\ \boldsymbol{h} = \begin{pmatrix} 25{,}000 \\ 21{,}000 \\ 13{,}000 \\ 11{,}000 \\ 10{,}500 \end{pmatrix}; \boldsymbol{\rho} = \begin{pmatrix} 48 & 62 & 72 \\ 55 & 50 & 66 \\ 50 & 44 & 52 \\ 65 & 58 & 48 \\ 70 & 65 & 44 \end{pmatrix};$
	$\boldsymbol{b} = \begin{pmatrix} 0 \\ 70 \\ 555 \\ 200 \end{pmatrix};\ \boldsymbol{b}' = \begin{pmatrix} 0 \\ 7{,}000 \\ 155{,}400 \\ 26{,}000 \end{pmatrix}; \boldsymbol{Q} = \begin{pmatrix} 0 \\ 220 \\ 150 \\ 350 \end{pmatrix};\ \boldsymbol{Q}' = \begin{pmatrix} 0 \\ 6{,}600 \\ 3{,}750 \\ 18{,}900 \end{pmatrix}; \boldsymbol{R} = \begin{pmatrix} 16{,}000 \\ 18{,}000 \\ 11{,}000 \end{pmatrix}.$
	$\underset{\forall j \in J}{A_j} = 4000;\ \underset{\forall j \in J}{l_j} = 8000;\ \underset{\forall j \in J}{C_j} = 1000;$
S2	$I = \{i_1, i_2, \ldots, i_{10}\};\ J = \{j_1, j_2, j_3, j_4, j_5\};\ A = \{a_0, a_1, a_2, a_3, a_4\}; D = \{d_0, d_1, d_2, d_3, d_4\}; S = \{s_1, s_2, \ldots, s_{32}\}$

$$\boldsymbol{f}=\begin{pmatrix}42\\48\\24\\32\\17\end{pmatrix}\times 10^6;\ \boldsymbol{h}=\begin{pmatrix}150\\10\\8\\180\\7\\20\\120\\9\\280\\130\end{pmatrix}\times 10^2;\boldsymbol{\rho}=\begin{pmatrix}16&61&63&78&74\\10&56&60&72&70\\42&19&42&75&56\\50&17&44&61&49\\66&49&31&76&28\\58&48&13&66&29\\60&51&9&48&27\\70&79&38&6&31\\76&82&51&7&41\\75&71&46&69&8\end{pmatrix}\times 10^2;$$

$$\boldsymbol{b}=\begin{pmatrix}0\\70\\200\\250\\555\end{pmatrix};\ \boldsymbol{b}'=\begin{pmatrix}0\\7000\\26000\\33750\\155400\end{pmatrix};\boldsymbol{Q}=\begin{pmatrix}0\\220\\150\\270\\350\end{pmatrix};\ \boldsymbol{Q}'=\begin{pmatrix}0\\6600\\3750\\9450\\18900\end{pmatrix};\boldsymbol{R}=\begin{pmatrix}180000\\175000\\165000\\166000\\190000\end{pmatrix}.$$

$$\underset{\forall j\in J}{A_j}=10^3;\ \underset{\forall j\in J}{l_j}=48\times 10^4;\ \underset{\forall j\in J}{C_j}=10^3;$$

$$I=\{i_1,i_2,\dots,i_{10}\};\ J=\{j_1,j_2,\dots,j_{10}\};\ A=\{a_0,a_1,a_2,a_3\};D=\{d_0,d_1,d_2,d_3\};S=\{s_1,s_2,\dots,s_{1024}\}$$

M1

$$\boldsymbol{f}=\begin{pmatrix}1158\\1018\\966\\1614\\778\\771\\495\\795\\1124\\687\end{pmatrix}\times 10^2;\boldsymbol{h}=\begin{pmatrix}29760021\\17990455\\6628637\\6016425\\4866692\\3665228\\2776755\\1227928\\1109252\\453588\end{pmatrix};\boldsymbol{\rho}=\begin{pmatrix}0&2483&2343&2622&590&629&1481&2664&2588&901\\2483&0&539&143&2384&2154&1016&234&120&1588\\2343&539&0&608&2383&1896&898&753&636&1455\\2622&143&608&0&2522&2294&1159&152&68&1732\\590&2384&2383&2522&0&1091&1486&2523&2471&983\\629&2154&1896&2294&1091&0&1150&2365&2269&660\\1481&1016&898&1159&1486&1150&0&1219&1128&580\\2664&234&753&152&2523&2365&1219&0&119&1779\\2588&120&636&68&2471&2269&1128&119&0&1696\\901&1588&1455&1732&983&660&580&1779&1696&0\end{pmatrix};$$

$$\boldsymbol{b}=\begin{pmatrix}0\\100\\280\\130\end{pmatrix};\ \boldsymbol{b}'=\begin{pmatrix}0\\7{,}000\\155{,}400\\26{,}000\end{pmatrix};\boldsymbol{Q}=\begin{pmatrix}0\\30\\25\\54\end{pmatrix};\ \boldsymbol{Q}'=\begin{pmatrix}0\\6{,}600\\3{,}750\\18{,}900\end{pmatrix};\ \underset{\forall j\in J}{R_j}=100;\ \pi=100.$$

$$\underset{\forall j\in J}{A_j}=10^3;\ \underset{\forall j\in J}{l_j}=10;\ \underset{\forall j\in J}{C_j}=5;$$

References

Akbari-Jafarabadi, M., Tavakkoli-Moghaddam, R., Mahmoodjanloo, M., Rahimi, Y., 2017. A tri-level r-interdiction median model for a facility location problem under imminent attack. *Comput. Ind. Eng.* 114(C), 151-165.

An, Y., Zeng, B., Zhang, Y., Zhao, L., 2014. Reliable p-median facility location problem: two-stage robust models and algorithms. *Transportation Research Part B: Methodological* 64(0), 54-72.

Berrebi, C., Ostwald, J., 2013. Exploiting the Chaos: Terrorist Target Choice Following Natural Disasters. *Southern Economic Journal* 79(4), 793-811.

Bricha, N., Nourelfath, M., 2013. Critical supply network protection against intentional attacks: A game-theoretical model. *Reliability Engineering & System Safety* 119, 1-10.

Bricha, N., Nourelfath, M., 2014. Extra-capacity versus protection for supply networks under attack. *Reliability Engineering & System Safety* 131, 185-196.

Chen, Q., Li, X., Ouyang, Y., 2011. Joint inventory-location problem under the risk of probabilistic facility disruptions. *Transportation Research Part B: Methodological* 45(7), 991-1003.

Diabat, A., Theodorou, E., 2015. A location–inventory supply chain problem: Reformulation and piecewise linearization. *Computers & Industrial Engineering* 90, 381-389.

Du, B., Zhou, H., Leus, R., 2020. A two-stage robust model for a reliable p-center facility location problem. *Applied Mathematical Modelling* 77, 99-114.

Fan, Y., Feng, Y., Shou, Y., 2020. A risk-averse and buyer-led supply chain under option contract: CVaR minimization and channel coordination. *International Journal of Production Economics* 219, 66-81.

Gan, H.-S., Safaei, A.S., 2016. Monitoring process variability: a hybrid Taguchi loss and multiobjective genetic algorithm approach. *Journal of Optimization in Industrial Engineering* 9(20), 1-8.

Ghavamifar, A., Makui, A., Taleizadeh, A.A., 2018. Designing a resilient competitive supply chain network under disruption risks: A real-world application. *Transportation Research Part E: Logistics and Transportation Review* 115, 87-109.

Heijnen, P., Schoonbeek, L., 2019. Rent-seeking with uncertain discriminatory power. *European Journal of Political Economy* 56, 103-114.

Jalali, S., Seifbarghy, M., Niaki, S.T.A., 2018. A risk-averse location-protection problem under intentional facility disruptions: A modified hybrid decomposition algorithm. *Transportation Research Part E: Logistics and Transportation Review* 114, 196-219.

Jalali, S., Seifbarghy, M., Sadeghi, J., Ahmadi, S., 2016. Optimizing a bi-objective reliable facility location problem with adapted stochastic measures using tuned-parameter multi-objective algorithms. *Knowledge-Based Systems* 95, 45-57.

Karamyar, F., Sadeghi, J., Yazdi, M.M., 2018. A Benders decomposition for the location-allocation and scheduling model in a healthcare system regarding robust optimization. *Neural Computing and Applications* 29(10), 873-886.

Khanduzi, R., Sangaiah, A.K., 2019. A fast genetic algorithm for a critical protection problem in biomedical supply chain networks. *Applied Soft Computing* 75, 162-179.

Kim, J.R., Lee, J.U., Jo, J.-B., 2009. Hierarchical spanning tree network design with Nash genetic algorithm. *Computers & Industrial Engineering* 56(3), 1040-1052.

Konak, A., Kulturel-Konak, S., Snyder, L.V., 2015. A Game-Theoretic Genetic Algorithm for the reliable server assignment problem under attacks. *Computers & Industrial Engineering* 85, 73-85.

Liberatore, F., Scaparra, M.P., Daskin, M.S., 2012. Hedging against disruptions with ripple effects in location analysis. *Omega* 40(1), 21-30.

Lin, B., Liu, S., Lin, R., Wang, J., Sun, M., Wang, X., Liu, C., Wu, J., Xiao, J., 2019. The location-allocation model for multi-classification-yard location problem. *Transportation Research Part E: Logistics and Transportation Review* 122, 283-308.

Mahmoodjanloo, M., Parvasi, S.P., Ramezanian, R., 2016. A tri-level covering fortification model for facility protection against disturbance in r-interdiction median problem. *Computers & Industrial Engineering* 102, 219-232.

Noyan, N., 2012. Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research* 39(3), 541-559.

Paul, J.A., Bagchi, A., 2018. Does Terrorism Increase after a Natural Disaster? An Analysis based upon Property Damage. *Defence and Peace Economics* 29(4), 407-439.

Peng, P., Snyder, L.V., Lim, A., Liu, Z., 2011. Reliable logistics networks design with facility disruptions. *Transportation Research Part B: Methodological* 45(8), 1190-1211.

Rawls, C.G., Turnquist, M.A., 2010. Pre-positioning of emergency supplies for disaster response. *Transportation Research Part B: Methodological* 44(4), 521-534.

Rizvi, S.A., Wajahat, A., 2019. Integration of grey-based Taguchi technique in optimization of parameters process during the turning operation of 16MnCr5 steel. *IUST* 30(3), 245-254.

Sadeghi, J., Niaki, S.T.A., Malekian, M.R., Wang, Y., 2018. A Lagrangian Relaxation for a Fuzzy Random EPQ Problem with Shortages and Redundancy Allocation: Two Tuned Meta-heuristics. *International Journal of Fuzzy Systems* 20(2), 515-533.

Salehi, A., Masoumi, B., 2019. Participative Biogeography-Based Optimization. *Journal of Optimization in Industrial Engineering* 12(1), 79-91.

Scaparra, M.P., Church, R.L., 2008. A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research* 35(6), 1905-1923.

Setak, M., Feizizadeh, F., Tikani, H., Ardakani, E.S., 2019. A bi-level stochastic optimization model for reliable supply chain in competitive environments: Hybridizing exact method and genetic algorithm. *Applied Mathematical Modelling* 75, 310-332.

Shen, B., Li, Q., 2016. Market disruptions in supply chains: a review of operational models. *International Transactions in Operational Research* 24(4), 697-711.

Tang, C.S., 2006. Robust strategies for mitigating supply chain disruptions. *International Journal of Logistics Research and Applications* 9(1), 33-45.

Yolmeh, A., Baykal-Gürsoy, M., 2017. A robust approach to infrastructure security games. *Computers & Industrial Engineering* 110, 515-526.

Zeng, B., Zhao, L., 2013. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters* 41(5), 457-461.

Zhang, Y., Snyder, L.V., Qi, M., Miao, L., 2016. A heterogeneous reliable location model with risk pooling under supply disruptions. *Transportation Research Part B: Methodological* 83, 151-178.

Zheng, X., Yin, M., Zhang, Y., 2019. Integrated optimization of location, inventory and routing in supply chain network design. *Transportation Research Part B: Methodological* 121, 1-20.

This article can be cited: Behnamian, J. (2021). A Risk-averse Inventory-based Supply Chain Protection Problem with Adapted Stochastic Measures under Intentional Facility Disruptions: Decomposition and Hybrid Algorithms*Journal of Optimization in Industrial Engineering*. 13 (2), 211-226.

http://www.qjie.ir/article_671444.html

DOI: 10.22094/JOIE.2019.1868275.1654