Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL
Pacific Journal of Optimization
Vol. 18, No. 3, pp. 635–660, 2022

A STOCHASTIC VARIATIONAL INEQUALITY APPROACH TO THE NASH EQUILIBRIUM MODEL OF A MANUFACTURER-SUPPLIER GAME UNDER UNCERTAINTY*

Yuan Lu, Jie Sun, Min Zhang† and Ying Zhang

Abstract: We study a two-stage game model of multi-manufacturers and multi-suppliers in supply-chain optimization, where N suppliers compete to provide M products for manufacturers in an environment of uncertainty. At the first stage (the contracting stage) the suppliers compete on the basis of delivery frequency to the manufacturers, while at the second stage (the production stage) the suppliers minimize their respective production costs, which could be influenced by other suppliers' decisions and the uncertainty in the market and their production processes. We formulate this problem as a two-stage quadratic stochastic Nash game. The Nash game is converted to a possibly nonmonotone stochastic variational inequality problem. A recently developed progressive hedging algorithm is proposed for finding a Nash equilibrium of the game. This algorithm is particularly suitable for stochastic variational inequality problems due to its decomposable structure in terms of the scenarios. Numerical results are presented to show the effectiveness of the proposed algorithm and to provide suggestions on the choice of the algorithmic parameters.

Key words: *stochastic variational inequalities; Nash game; progressive hedging algorithms*

Mathematics Subject Classification: *91A10, 90B06, 90C15, 90C33*

---

1 Introduction

Consider a supply chain network consisting of N suppliers and M manufacturers, in which manufacturer i has demand Δ_i (could be random) for product i within a given period. If a manufacturer needs two different products, then she could be treated as two different manufacturers in this model. Denote $[N] = \{1, ..., N\}$ and $[M] = \{1, ..., M\}$. Each supplier, on the other hand, can provide certain amount of the M products subject to her production constraints. The competition of the suppliers is described by a two-stage noncooperative game under uncertainty. At the first stage (the contracting stage), the jth supplier ($j \in [N]$) proposes delivery frequencies x_{ij} to get contracts from the ith manufacturer ($i \in [M]$). Based on the proposed values of x_{ij}, the ith manufacturer then uses an EOQ logic to determine

*The research is partially supported by Australian Research Council under grant DP160102819; National Natural Science Foundation of China under grant 11301347, 11671184 and 12101598; and the Chinese Academy of Sciences under grant 2021000088.

†Corresponding author.

In the economic order quantity (EOQ) model, the manufacturer has a fixed cost α per order, a holding cost h per unit of inventory held and a total demand Δ for one period. The well-known EOQ logic finds the ideal order quantity by minimizing the total purchase-inventory cost in the period.

a portion λ_{ij} of Δ_i, where Δ_i is the total demand for product i, to allocate to supplier j. Naturally, it holds that $\lambda_{ij} \geq 0$ and $\sum_j \lambda_{ij} = 1$. After determined the order λ_{ij}, a random vector ξ is realized and observed by all suppliers, which may provide new information that will influence the production of the suppliers such as the market price and availability of raw materials and various random technical parameters in their production processes. Then at the second stage (the production stage) every supplier j has to solve an optimization problem parameterized by ξ to determine her production amount $y_{ij}(\xi)$, while fulfilling her promised delivery frequency x_{ij} for all i and satisfying other production constraints. In a deterministic environment, as it mostly is in the literature, this problem could be formulated as a leader-follower Nash game, in which the suppliers (the leaders) solve a quasi-variational inequality problem and the manufacturers (the followers) solve an EOQ problem.

Several non-stochastic versions of this problem has been studied from different angles in the literature (see for instances [4–6, 8, 12, 16, 18, 21–23, 36]). We note that in particular, [1] formulated this problem as a deterministic Nash game among the suppliers, but they did not provide a general scheme for solving that Nash game. Besides, there seems to be no paper that considered the multi-follower case.

While game models and uncertainty have been long recognized as inevitable factors in manufacture and supply chain management (see for examples [7, 35]), there have been only very few papers discussing game models in combination with stochasticity. In particular, there has been no paper on multi-stage stochastic Nash games until very recently [20, 27, 40]. This is probably due to the fact that there has been no clear idea on the notion of "stochastic variational inequality" in the optimization community until the seminal paper of [34], in which this notion is clearly defined as a variational inequality in a specific Hilbert space of response functions to the stochastic vector ξ. Such problems have not previously been spotlighted, or for that matter even given a name. The motivation of this paper is to introduce this new development to the community of supply chain management, therefore to provide a new tool to researchers and practitioners in these fields. Our focus in this paper is to address the " optimization " side of this problem; namely, we are concerned with the general mathematical structure of the problem, the conditions for existence of solutions and possible numerical methods, rather than the "management side" of this problem, which may include managerial insights of the solutions and sensitivity analysis for real-world applications. Although the current paper does not include application examples, it is our belief that the proposed model and computational scheme would have an impact on solving real-world problems. After the first draft of this paper is finished, we noticed that [13] studied a one-leader-multiple-follower game in which every player solved a two-stage stochastic optimization problem, and applied it in the flight booking problem under uncertain flight delays and multi-stakeholder shale gas supply chain. Thereafter, such leader-follower game models under uncertainty have been adopted and used in real-world applications such as shale gas-water supply chains [10], global crude oil purchase and sale planning [25], customer allocation in oligopolies [9], and problems on the energy, water, and food nexus [24]. All these impressive developments have provided further motivational support to the study of a general Nash game model under uncertainty from the angle of stochastic variational inequality.

The contributions of this paper are as follows.

1. We introduce a two-stage multi-manufacturer-multi-supplier game model under uncertainty that attempts to suitably combine the competition factor and the randomness factor in supply-chain management. We show that this game is essentially a two-stage quadratic game under uncertainty among the suppliers. In particular, each supplier has to solve a two-stage stochastic quadratic minimization problem parameterized by

other suppliers' decisions, which appears to be new in the literature of supply chain management.

2. We introduce certain recent developments in stochastic optimization such as the response function space and stochastic variational inequality (SVI) theory to describe the mathematical structure of the proposed game model. Moreover, we demonstrate how the suppliers' game can be identified as a two-stage stochastic linear complementarity (SLC) problem.

3. The progressive hedging algorithm, initiated from multistage stochastic optimization and recently transplanted to the area of SVI, is then introduced as a scenario-based decomposition algorithm for the Nash equilibrium of the suppliers' game. It is argued that, if the resulted SLC problem is monotone or elicitable monotone (see its meaning later), then it can be numerically solved by the progressive hedging algorithm.

4. Preliminary numerical results are presented to show the effectiveness of the proposed algorithm. Randomly generated problems of various sizes and different number of scenarios are tested. Certain critical observations are made and heuristics in choosing the key parameters are provided.

The rest of this paper is organized as follows. In Section 2 the manufacturer-supplier game is formulated as a two-stage game of the suppliers under uncertainty. Section 3 establishes the connection between the proposed game model and an SLC problem, which is a special type of the SVI problem. Section 4 describes specially designed progressive hedging algorithms for the resulting SLC problems for monotone and elicitable monotone problems, respectively, and presents convergence results on the algorithms. Section 5 reports results of numerical experiments on randomly generated games with $[M, N]$ increases from [2,2] to [10,10] and number of scenarios grows from 10 to 200. Section 6 concludes this paper.

$\boxed{2}$ The Two-Stage Multi-Leader Multi-Follower Nash Game Model Under Uncertainty

We proceed to formulate a leader-follower Nash game model for competition among the manufacturers and the suppliers. Generally, we adopt the notation system commonly used in optimization. For example, lower case letters represent vectors, such as x and upper case letters represent matrices or sets, such as A or $\mathcal{A}$. In addition, unless otherwise specified, we denote scalars and random vectors by Greek letters. Conventionally, $\mathbb{R}^n$ stands for the usual n-dimensional Euclidean space and by "Hilbert space" we mean a vector space equipped with an inner product. Therefore, $\mathbb{R}^n$ is a Hilbert space, but not vice versa since the inner product could be defined differently.

The parameters and variables related to the suppliers ($j \in [N]$) and manufacturers ($i \in [M]$) involved in the Nash game are as follows.

Variables (where ξ is a random vector of finite discrete distribution with support Ξ):

x_{ij}: the delivery frequency of supplier j to manufacturer i at the first stage;

x: the vector combining all x_{ij} in a natural order; namely

$$x = (x_{11}, \ldots, x_{1N}, \ldots, x_{M1}, \ldots, x_{MN})^T,$$

where "T" stands for the transpose;

(all vectors in this paper are column vectors)

$y_{ij}(\xi)$: the amount of product i produced by supplier j at the second stage;

$y(\xi)$: the vector combining all $y_{ij}(\xi)$; namely

$$y(\xi) = (y_{11}(\xi), \ldots, y_{1N}(\xi), \ldots, y_{M1}(\xi), \ldots, y_{MN}(\xi))^T;$$

$z(\xi)$: combination of x and $y(\xi)$, namely $z(\xi) = (x^T, y(\xi)^T)^T$;

$z(\cdot)$: the mapping from ξ to $z(\xi)$, which is also called "the response function";

λ_{ij} : demand allocations to each supplier j determined by the manufacturer i at the first stage;

λ: the vector combining all λ_{ij}, $\lambda = (\lambda_{11}, \ldots, \lambda_{1N}, \ldots, \lambda_{M1}, \ldots, \lambda_{MN})^T$;

λ_i : the demand allocation vector of manufacturer i, i.e., $\lambda_i = (\lambda_{i1}, \ldots, \lambda_{iN})^T$;

x_j : the subvector of x consisting of all components involving supplier j, i.e., $x_j = (x_{1j}, \ldots, x_{Mj})^T$;

x_{-j} : the vector obtained by deleting $x_{1j}, ..., x_{Mj}$ from x;

$y_j(\xi), y_{-j}(\xi), z_j(\xi), z_{-j}(\xi), z_j(\cdot)$ and $z_{-j}(\cdot)$, are similarly defined.

Objective Functions:

$\theta_j(x_j; x_{-j})$: the first-stage cost function of supplier j, where x_j is the decision vector and x_{-j} is viewed as a parameter;

$\phi_j(y_j(\xi); x, y_{-j}(\xi), \xi)$: the second-stage cost function of supplier j, where $y_j(\xi)$ is the decision vector and $x, y_{-j}(\xi), \xi$ are regarded as parameters (*Note: we use a semicolon ";" to separate decision variables and parameters, and we will do the same below*).

Parameters of the first stage:

Δ_i: manufacturer i's demand;

h_i: unit inventory holding cost of manufacturer i;

p_{ij}: unit selling price of the product i from supplier j with $p_j = (p_{1j}, \ldots, p_{Mj})^T$;

γ_{ij}: unit flexible cost to produce product i of supplier j;

β_{ij}: unit flexible cost to deliver product i of supplier j;

Γ_{ij}: fixed cost to pack and deliver a batch of products to manufacturer i from supplier j.

Parameters of the second stage will be introduced when the concrete form of θ_j and ϕ_j are introduced. They might be dependent on ξ.

In the following analysis, we assume that the parameters $\Delta_i > 0$, $h_i > 0$, $p_{ij} > 0$, $\gamma_{ij} > 0$, $\beta_{ij} > 0$, and $\Gamma_{ij} > 0$ are fixed scalars for $i \in [M], j \in [N]$. We assume that suppliers may set different prices p_{ij} of the product and those prices are fixed constants. However, as it often happens in practice, this price is often specified by buyer (the manufacturers), in that case it simply becomes $p_{ij} = p_{ik}$ for all $j \neq k$. Note that it is reasonable to assume that the selling price p_{ij} is greater than the sum of unit production cost γ_{ij} and unit transportation cost β_{ij} of the product, i.e., $p_{ij} > \gamma_{ij} + \beta_{ij}$, for all $i \in [M]$ and $j \in [N]$.

We now show that the solutions to the followers' (i.e. the manufacturers') EOQ problems have explicit forms. Thus, the value of λ can be eliminated from the objectives of the leaders' (i.e. the suppliers') game. As a result, the leaders' game will become a two-stage stochastic quadratic game on $z(\cdot)$ only. Consider supplier j with $x_{ij} > 0$. Suppose that the product i is delivered in batch of size s_i from the suppliers. Then by using the EOQ logic with the similar arguments as [15], the total inventory cost of manufacturer i due to supplier j is $(h_i x_{ij}) \left(\frac{1}{2}\right) s_{ij} \left(\frac{s_{ij}}{\Delta_i}\right) = \frac{h_i \lambda_{ij}^2 \Delta_i}{2x_{ij}}$, where $s_{ij} = \frac{\lambda_{ij}\Delta_i}{x_{ij}}$. Thus, manufacturer i can obtain its optimal allocation decision by minimizing the total cost of purchase and inventory, i.e.,

$$\begin{aligned} \min_{\lambda_i} \quad & \sum_{j=1}^{N} p_{ij}\lambda_{ij}\Delta_i + \sum_{j=1}^{N} \frac{h_i \lambda_{ij}^2 \Delta_i}{2x_{ij}} \\ \text{s.t.} \quad & \sum_{j=1}^{N} \lambda_{ij} = 1, \ \lambda_{ij} \geq 0 \ \forall j \in [N]. \end{aligned} \tag{2.1}$$

Note that the optimal solution λ_i^* is independent of Δ_i due to $\Delta_i > 0$. Since the objective is strictly convex and quadratic, the optimal solution exists and is unique. Let the corresponding Lagrange multiplier (which can be shown to be unique) to the equation constraint be v_i. Then, based on the equilibrium theorem of monotropic programming (see Chapter 11 of [29]), the optimal solution is

$$\lambda_{ij}^* = \max\left[0, \frac{x_{ij}}{h_i}(v_i - p_{ij})\right] \text{ for } j \in [N]. \tag{2.2}$$

Direct verification shows that this formula also applies to $x_{ij} = 0$. Thus, (2.2) applies to all $x \geq 0$. However, it is reasonable to assume $v_i > p_{ij}$ for otherwise the manufacturer should instead consider $k(< N)$ suppliers rather than the N suppliers by ignoring the suppliers with zero allocation.

From $\sum_{j=1}^{N} \lambda_{ij}^* = 1$, one has

$$v_i = \left(\sum_{k=1}^{N} x_{ik}\right)^{-1} \left(\sum_{k=1}^{N} x_{ik}p_{ik} + h_i\right) \text{ for all } i \in [M]. \tag{2.3}$$

Note that constraint (2.1) also guarantees $\sum_{k=1}^{N} x_{ik} > 0$. Substituting (2.3) into (2.2), we have

$$\lambda_{ij}^* = \frac{x_{ij}}{\sum_{k=1}^{N} x_{ik}} \left[1 + \frac{1}{h_i} \sum_{k=1}^{N} x_{ik}(p_{ik} - p_{ij})\right], \text{ for all } i \in [M], \ j \in [N]. \tag{2.4}$$

Therefore, $\sum_{j=1}^{N} \lambda_{ij}^* = 1$ for all i and $\lambda_{ij}^* > 0$ are equivalent to (2.3) and $v_i > p_{ij}$ for all i, which are further equivalent to that there exists a positive number ε such that

$$\sum_{k=1}^{N} x_{ik}p_{ij} - \sum_{k=1}^{N} x_{ik}p_{ik} \leq h_i - \varepsilon, \text{ for all } i \in [M], \ j \in [N], \tag{2.5}$$

where ε is a sufficiently small positive number that is introduced to avoid the possibility that λ_{ij}^* becomes zero for some i and j.

Assume that for manufacturer i, it requires a total number of r_i deliveries of the ith product at the first stage (the contracting stage), therefore we can further simplify (2.4) by using the following relationship among the x_{ij}s:

$$x_{i1} + \cdots + x_{iN} = r_i, \text{ with } x_{i1}, ..., x_{iN} \geq 0 \quad \forall i \in [M]. \tag{2.6}$$

Note that the total cost at the first stage of supplier j is

$$\sum_{i=1}^{M} \left[(\gamma_{ij} + \beta_{ij} - p_{ij})\Delta_i \lambda_{ij}^* + \Gamma_{ij} x_{ij} \right]. \tag{2.7}$$

Substituting the explicit formula of λ_{ij}^* into (2.7) and using (2.6), the first-stage cost function of supplier j becomes

$$\begin{aligned}
\theta_j(x_j; x_{-j}) &= \sum_{i=1}^{M} \left\{ \frac{(\gamma_{ij} + \beta_{ij} - p_{ij})\Delta_i}{r_i} x_{ij} \left[1 + \frac{1}{h_i} \sum_{k=1}^{N} x_{ik}(p_{ik} - p_{ij}) \right] + \Gamma_{ij} x_{ij} \right\} \\
&= \sum_{i=1}^{M} c_{ij} x_{ij} + \sum_{k \neq j} \sum_{i=1}^{M} \left[\frac{(p_{ij} - \gamma_{ij} - \beta_{ij})\Delta_i}{r_i h_i} x_{ij}(p_{ij} - p_{ik}) x_{ik} \right] \\
&= c_j^T x_j + x_j^T R_{-j} x_{-j},
\end{aligned} \tag{2.8}$$

where $c_j = (c_{1j}, \ldots, c_{Mj})^T$ with $c_{ij} = \Gamma_{ij} - \frac{(p_{ij} - \gamma_{ij} - \beta_{ij})\Delta_i}{r_i}$ and $R_{-j} = (R_{jk})_{k \in [N], k \neq j}$ with

$$R_{jk} = \begin{pmatrix} \frac{(p_{1j} - \gamma_{1j} - \beta_{1j})(p_{1j} - p_{1k})\Delta_1}{r_1 h_1} & & \\ & \ddots & \\ & & \frac{(p_{Mj} - \gamma_{Mj} - \beta_{Mj})(p_{Mj} - p_{Mk})\Delta_M}{r_M h_M} \end{pmatrix} \in \mathbb{R}^{M \times M}.$$

To keep flexibility of this model, instead of specifying various concrete constraints for the second stage problem, we assume that the production of supplier j at the second stage is the minimizer of a quadratic production cost function parameterized by x and $y_{-j}(\xi)$ for every ξ as follows (note that $y_j(\xi)$ is the decision variable, the others are regarded as parameters).

$$\begin{aligned}
& \phi_j(y_j(\xi); x, y_{-j}(\xi), \xi) \\
= \; & \frac{1}{2} y_j(\xi)^T O_{jj}(\xi) y_j(\xi) + \sum_{k \neq j} y_j(\xi)^T O_{jk}(\xi) y_k(\xi) + \sum_{k=1}^{N} y_j(\xi)^T P_{jk}(\xi) x_k + d_j(\xi)^T y_j(\xi) \\
:= \; & \frac{1}{2} y_j(\xi)^T O_{jj}(\xi) y_j(\xi) + y_j(\xi)^T O_{j,-j}(\xi) y_{-j}(\xi) \\
& + y_j(\xi)^T P_{jj}(\xi) x_j + y_j(\xi)^T P_{j,-j}(\xi) x_{-j} + d_j(\xi)^T y_j(\xi),
\end{aligned} \tag{2.9}$$

where $P_{jk}(\xi), O_{jk}(\xi)$, $P_{j,-j}(\xi)$ and $O_{j,-j}(\xi)$ are ξ-dependent matrices of appropriate dimensions with

$$P_{j,-j}(\xi) x_{-j} = \sum_{k \neq j} P_{jk}(\xi) x_k, \quad O_{j,-j}(\xi) y_{-j}(\xi) = \sum_{k \neq j} O_{jk}(\xi) y_k(\xi),$$

and $d_j(\xi)$ is a random vector of appropriate dimension for $j \in [N]$, subject to a set of linear constraints of the form

$$
\begin{aligned}
& Y_j(x, y_{-j}(\xi), \xi) \\
= \quad & \left\{ y_j(\xi) \in \mathbb{R}_+^M : D_j(\xi)x_j + \sum_{k \neq j} D_{jk}(\xi)x_k + B_j(\xi)y_j(\xi) + \sum_{k \neq j} B_{jk}(\xi)y_k(\xi) \geq b_j(\xi) \right\}, \\
=: \quad & \left\{ y_j(\xi) \in \mathbb{R}_+^M : D_j(\xi)x_j + D_{-j}(\xi)x_{-j} + B_j(\xi)y_j(\xi) + B_{-j}(\xi)y_{-j}(\xi) \geq b_j(\xi) \right\},
\end{aligned}
\tag{2.10}
$$

where $\mathbb{R}_+^M$ represents the nonnegative orthant of the space $\mathbb{R}^M$ and D_j, B_j, $D_{jk}(\xi)$ and $B_{jk}(\xi)$ are random matrices of appropriate dimensions with

$$
D_{-j}(\xi)x_{-j} = \sum_{k \neq j} D_{jk}(\xi)x_k, \ B_{-j}(\xi)y_{-j}(\xi) = \sum_{k \neq j} B_{jk}(\xi)y_k(\xi),
$$

and $b_j(\xi)$ is a random vector of appropriate dimension for $j \in [N]$.

Overall, the objective function of supplier j is the expectation of the total costs in two stages

$$
\mathbb{E}_\xi \left[\theta_j(x_j; x_{-j}) + \phi_j(y_j(\xi); x, y_{-j}(\xi), \xi) \right], \tag{2.11}
$$

where $\mathbb{E}_\xi$ stands for the expectation over ξ.

We comment on the generality of this model. First, if $D_j(\xi), D_{-j}(\xi), B_j(\xi),\ B_{-j}(\xi)$, $b_j(\xi)$ and $\phi_j(y_j(\xi), x(\xi), y_{-j}(\xi), \xi)$ are independent of ξ, it is a deterministic game problem for (x_j, y_j), parameterized by the decisions of the other suppliers in both stages; second, as long as one element of $D_j(\xi), D_{-j}(\xi), B_j(\xi),\ B_{-j}(\xi)\ b_j(\xi)$ and $\phi_j(y_j(\xi), x, y_{-j}(\xi), \xi)$ is dependent on ξ, the solution of the game will be generally dependent on ξ and it is therefore sensible to write its solution as $(x_j, y_j(\xi))$; third, since the Nash equilibrium (if any) of the model is dependent on ξ, it is no longer a single vector. Instead, the solution is a response **function (or mapping)** of ξ

$$
z(\cdot) := (x(\cdot), y(\cdot))^T : \Xi \to \mathbb{R}^n,
$$

where Ξ is the support of the random vector ξ, which is the finite sample space composed of all possible realizations of the random vector ξ, and $\mathbb{R}^n := \mathbb{R}^{MN} \times \mathbb{R}^{MN}$. Intuitively, we could identify $z(\cdot)$ with a vector of length $n|\Xi|$, where $|\Xi|$ is the cardinality of Ξ. Since $|\Xi|$ is exponential in term of the dimension of ξ, any direct solution method for $z(\cdot)$ seems unrealistic.

Let $\mathscr{L}$ be the Hilbert space consisting of all response functions from Ξ to $\mathbb{R}^n$, equipped with the inner product

$$
\langle z(\cdot), w(\cdot) \rangle := \mathbb{E}_\xi[z(\xi)^T w(\xi)] := \sum_{\xi \in \Xi} \pi(\xi) z(\xi)^T w(\xi),
$$

where $\pi(\xi) > 0$ is the probability of sample ξ and all such probabilities add up to one. A notational difference ought to be emphasized. By $z(\cdot)$ we mean a function from Ξ to $\mathbb{R}^n$, but by $z(\xi)$ we mean the image of ξ under the mapping $z(\cdot)$, where ξ is a certain scenario in Ξ.

Since the first-stage decision x has to be made before ξ is realized, the solution of any two-stage stochastic optimization problem must be "nonanticipative". The nonanticipativity constraint for $z(\cdot)$ is defined as $z(\cdot) \in \mathcal{N}$, where

$$
\mathcal{N} := \{ z(\cdot) \in \mathscr{L} \ : \ \text{the } x\text{-part of } z(\xi) \text{ is independent of } \xi \}.
$$

Obviously, $\mathcal{N}$ is a linear subspace of $\mathscr{L}$, and its complementary subspace is denoted by $\mathcal{M}$, which is important for algorithmic development. Correspondingly, for supplier j, the solution has to satisfy $z_j(\cdot) \in \mathcal{N}_j$, where

$$\mathcal{N}_j := \{z_j(\cdot) \ : \ \text{the } x_j\text{-part of } z_j(\xi) \text{ is independent of } \xi\}. \tag{2.12}$$

Under the nonanticipativity constraint $z(\cdot) \in \mathcal{N}$ it is no confusion to superfluously write $x = x(\xi)$, which will provide a more elegant and generalized view on the progressive hedging algorithm.

Let us write constraints for supplier j as

$$z_j(\cdot) \in \mathcal{N}_j \cap \mathcal{C}_j(z_{-j}(\cdot)),$$

where $\mathcal{N}_j$ imposes the nonantipativity and $\mathcal{C}_j(z_{-j}(\cdot))$ describes other constraints for $z_j(\xi)$, which may depend on ξ and $z_{-j}(\xi)$, and is interpreted as follows.

$$z_j(\cdot) \in \mathcal{C}_j(z_{-j}(\cdot)) \quad \Longleftrightarrow \quad z_j(\xi) \in C_j(z_{-j}(\xi), \xi) \ \forall \ \xi$$

with

$$C_j(z_{-j}(\xi), \xi) := \left\{ z_j(\xi) = \begin{pmatrix} x_j(\xi) \\ y_j(\xi) \end{pmatrix} \ : \ \begin{array}{l} x_j(\xi) \text{ satisfies (2.5) and (2.6)}, \\ y_j(\xi) \in Y_j(x(\xi), y_{-j}(\xi), \xi) \end{array} \right\}. \tag{2.13}$$

Again, we emphasize that $\mathcal{C}_j(z_{-j}(\cdot))$ is a set in $\mathscr{L}$ and $C_j(z_{-j}(\xi), \xi)$ is a set in $\mathbb{R}^n$. For convenience of analysis, let us re-write the first-stage constraint as

$$x_j(\xi) \text{ satisfies (2.5) and (2.6)} \Leftrightarrow x_j(\xi) \geq 0, \ A_j x_j(\xi) + A_{-j} x_{-j}(\xi) \geq a_j, \tag{2.14}$$

where A_j, A_{-j} and a_j are the matrices and vectors defined by (2.5) and (2.6), respectively. In fact, conditions (2.5) and (2.6) are equivalent to that $x_j(\xi) \geq 0$, and

$$\begin{pmatrix} I_{M\times M} \\ -I_{M\times M} \\ \mathrm{Diag}(p_1) \end{pmatrix} x_1(\xi) + \begin{pmatrix} I_{M\times M} \\ -I_{M\times M} \\ \mathrm{Diag}(p_2) \end{pmatrix} x_2(\xi) + \cdots + \begin{pmatrix} I_{M\times M} \\ -I_{M\times M} \\ \mathrm{Diag}(p_N) \end{pmatrix} x_N(\xi) \geq \begin{pmatrix} r \\ -r \\ r \circ \bar{p} - h + \epsilon \end{pmatrix},$$

where $I_{M\times M}$ is an identity matrix in size $M \times M$, $\mathrm{Diag}(p_j)$ is a diagonal matrix with diagonal elements being vector $p_j = (p_{1j}, \ldots, p_{Mj})^T$, $r = (r_1, \ldots, r_M)^T$, $\bar{p} = (\max_j\{p_{1j}\}, \ldots, \max_j\{p_{Mj}\})^T$, $h = (h_1, \ldots, h_M)^T$, and $\circ$ means the Hadamard product, i.e., $r \circ \bar{p} = (r_1 \bar{p}_1, \ldots, r_M \bar{p}_M)^T$; thus, A_j, A_{-j}, a_j in (2.14) has the following form:

$$A_j = \begin{pmatrix} I_{M\times M} \\ -I_{M\times M} \\ \mathrm{Diag}(p_j) \end{pmatrix}, \ A_{-j} = (A_k)_{k\in[N], k\neq j}, \text{ and } a_j = \begin{pmatrix} r \\ -r \\ r \circ \bar{p} - h + \epsilon \end{pmatrix} \ \forall \ j \in [N].$$

Thus, $C_j(z_{-j}(\xi), \xi))$ defined in (2.13) has the following specific "analytical form"

$$C_j(z_{-j}(\xi), \xi) = \left\{ z_j(\xi) : \bar{A}_j(\xi) z_j(\xi) \geq \bar{b}_j(\xi) - \bar{A}_{-j}(\xi) z_{-j}(\xi) \text{ and } z_j(\xi) \geq 0 \right\}, \tag{2.15}$$

where

$$\bar{A}_j(\xi) = \begin{pmatrix} A_j & 0 \\ D_j(\xi) & B_j(\xi) \end{pmatrix}, \bar{A}_{-j}(\xi) = \begin{pmatrix} A_{-j} & 0 \\ D_{-j}(\xi) & B_{-j}(\xi) \end{pmatrix}, \text{ and } \bar{b}_j(\xi) = \begin{pmatrix} a_j \\ b_j(\xi) \end{pmatrix}.$$

Now, we can rewrite the objective function (2.11) of supplier j as the mean value of a quadratic function of $z_j(\cdot)$, namely

$$\begin{aligned}\mathcal{G}_j(z_j(\cdot); z_{-j}(\cdot)) &:= \mathbb{E}_\xi\left[\theta_j(x_j; x_{-j}) + \phi_j(y_j(\xi); x, y_{-j}(\xi), \xi)\right] \\ &= \mathbb{E}_\xi\left[\frac{1}{2} z_j(\xi)^T \bar{Q}_j(\xi) z_j(\xi) + \left(\bar{c}_j(\xi) + \bar{R}_{-j}(\xi) z_{-j}(\xi)\right)^T z_j(\xi)\right],\end{aligned}$$

where

$$\bar{Q}_j(\xi) = \begin{pmatrix} 0 & P_{jj}(\xi)^T \\ P_{jj}(\xi) & O_{jj}(\xi) \end{pmatrix}, \quad \bar{c}_j(\xi) = \begin{pmatrix} c_j \\ d_j(\xi) \end{pmatrix} \text{ and } \bar{R}_{-j}(\xi) = \begin{pmatrix} R_{-j} & 0 \\ P_{j,-j}(\xi) & O_{j,-j}(\xi) \end{pmatrix}.$$

Let us remark on randomness and non-randomness of the parameters in the first-stage problem, in particular on the case that the demand Δ_i in first stage is random, therefore is ξ-dependent. In that case, we only need regard it as parameters of the second stage. That is, we move the objective terms containing Δ_i into the second stage objective function and regard the constraints containing Δ_i as a second stage constraint and merge them into $\mathcal{C}_j(z_{-j}(\cdot))$. Then we write $\Delta_i = \Delta_i(\xi)$ and particularly, change R_{-j} to $R_{-j}(\xi)$ in the above formula. There would be no conceptual complications to the algorithmic development in the subsequent sections.

In summary, supplier j's problem in the game model is

$$\begin{aligned}\min_{z_j(\cdot)} \quad & \mathcal{G}_j(z_j(\cdot); z_{-j}(\cdot)) \\ \text{s.t.} \quad & z_j(\cdot) \in \mathcal{N}_j \cap \mathcal{C}_j(z_{-j}(\cdot)),\end{aligned} \tag{2.16}$$

Note that $\mathcal{G}_j(z_j(\cdot); z_{-j}(\cdot))$ is quadratic in $z_j(\cdot)$ and $\mathcal{C}_j(z_{-j}(\cdot))$ is a polyhedron and therefore (2.16) is in general a quadratic optimization problem in space $\mathscr{L}$ with a quadratic objective function and linear constraints.

It appears that Model (2.16) is the most general two-stage quadratic game model in the literature so far. It allows all interactions (cross terms) in the objective and also allows all variables (all players' decisions in first stage and other players' decisions in the second stage) to show up in the linear constraints. We notice that there are often various restrictions on the cross terms like $x_j^T R_{-j} x_{-j}$ in the deterministic and stochastic game models in the literature. Perhaps for the sake of enabling certain specific algorithms, or simply for keeping convexity of the objective function. However, this kind of restriction do not appear in the proposed model. Moreover, compared to robust optimization approaches to stochastic optimization, a number of robust optimization methods (e.g., [2, 11, 19]) assume affine dependence of the random parameters on ξ. In contrast, there is no linear dependence requirements on ξ for the random parameters in the proposed model. Therefore, the proposed model may allow a wider range of potential applications. In fact, the proposed model allows all possible cross terms to be in both first and second stage objective functions.

3 Reformulation of the Suppliers' Game into an SLC Problem

In this section, we derive an equivalent formulation of problem (2.16) that is an SLC problem. Most of the work has appeared in [40]. We include the analysis here for self-containedness. Readers who are only interested in finding a numerical solution to Problem (2.16) may go to Section 4 directly and just keep in mind that Problem (2.16) is equivalent to the SLC Problem below.

Based on the theory of variational inequalities, a necessary condition for $z_j(\cdot)$ to be an optimal solution to problem (2.16) is

$$-\partial\mathcal{G}_j(z_j(\cdot); z_{-j}(\cdot)) \in N_{\mathcal{C}_j(z_{-j}(\cdot))\cap\mathcal{N}_j}(z_j(\cdot)), \tag{3.1}$$

where $\partial\mathcal{G}_j$ is the subdifferential mapping of $\mathcal{G}_j(z_j(\cdot), z_{-j}(\cdot))$ and $N_{\mathcal{C}_j(z_{-j}(\cdot))\cap\mathcal{N}_j}(z_j(\cdot))$ stands for the normal cone of set $\mathcal{C}_j(z_{-j}(\cdot)) \cap \mathcal{N}_j$ at $z_j(\cdot)$ in the sense of convex analysis [28]. Condition (3.1) will be also sufficient if problem (2.16) is convex for $z_j(\cdot)$. Under the following constraint qualification (CQ for short) condition

$$\mathcal{C}_j(z_{-j}(\cdot)) \cap \mathcal{N}_j \neq \emptyset \quad \forall j \in [N], \tag{3.2}$$

(which simply says that every player has a feasible solution) one has

$$N_{\mathcal{C}_j(z_{-j}(\cdot))\cap\mathcal{N}_j}(z_j(\cdot)) = N_{\mathcal{C}_j(z_{-j}(\cdot))}(z_j(\cdot)) + N_{\mathcal{N}_j}(z_j(\cdot)) = N_{\mathcal{C}_j(z_{-j}(\cdot))}(z_j(\cdot)) + \mathcal{M}_j,$$

where $\mathcal{M}_j$ represents the complementary subspace of $\mathcal{N}_j$. Therefore, under CQ, (3.1) is equivalent to

$$\text{Find } z_j(\cdot) \in \mathcal{N}_j, w_j(\cdot) \in \mathcal{M}_j \text{ s.t. } -\partial\mathcal{G}_j(z_j(\cdot); z_{-j}(\cdot)) - w_j(\cdot) \in N_{\mathcal{C}_j(z_{-j}(\cdot))}(z_j(\cdot)). \tag{3.3}$$

Specifically for $\mathcal{C}_j(z_{-j}(\cdot))$ in form (2.15), condition (3.3) is equivalent to the Karush-Kuhn-Tucker (KKT) condition of problem (2.16), which can be written as follows

$$0 \leq \begin{pmatrix} z_j(\xi) \\ \eta_j(\xi) \end{pmatrix} \perp \begin{pmatrix} \bar{Q}_j(\xi) & -\bar{A}_j^T(\xi) \\ \bar{A}_j(\xi) & 0 \end{pmatrix} \begin{pmatrix} z_j(\xi) \\ \eta_j(\xi) \end{pmatrix} + \begin{pmatrix} \bar{c}_j(\xi) + \bar{R}_{-j}(\xi)z_{-j}(\xi) \\ \bar{A}_{-j}(\xi)z_{-j}(\xi) - \bar{b}_j(\xi) \end{pmatrix} + \begin{pmatrix} w_j(\xi) \\ 0 \end{pmatrix} \geq 0, \tag{3.4}$$

where "$\perp$" means "is perpendicular to" and $\eta_j(\xi)$ is a dual vector. If $\bar{Q}_j(\xi)$ is positive semidefinite for all ξ and the optimal value of (2.16) is finite, then the KKT condition (3.4) is also sufficient for the existence of optimal $z_j(\cdot), w_j(\cdot)$, and $\eta_j(\cdot)$. However, we do not assume $\bar{Q}_j$ to be positive semi-definite in the following analysis.

Now let

$$\bar{R}_{-j}(\xi)z_{-j}(\xi) = \sum_{k\neq j} \bar{R}_{jk}(\xi)z_k(\xi), \quad \bar{A}_{-j}(\xi)z_{-j}(\xi) = \sum_{k\neq j} \bar{A}_{jk}(\xi)z_k(\xi)$$

with

$$\bar{R}_{jk}(\xi) = \begin{pmatrix} R_{jk} & 0 \\ P_{jk}(\xi) & O_{jk}(\xi) \end{pmatrix} \text{ and } \bar{A}_{jk}(\xi) = \begin{pmatrix} A_k & 0 \\ D_{jk}(\xi) & B_{jk}(\xi) \end{pmatrix}.$$

The Nash equilibrium of the game requires condition (3.4) to hold for all suppliers, writing all such conditions together, then the necessary conditions of the Nash equilibrium of the suppliers' game under uncertainty is to find

$$v(\xi) := (z_1(\xi), \eta_1(\xi), \ldots, z_N(\xi), \eta_N(\xi))^T \in \hat{\mathcal{N}} \text{ and } (w_1(\xi), 0, \ldots, w_N(\xi), 0)^T \in \hat{\mathcal{M}}$$

such that $\forall \xi \in \Xi$,

$$0 \leq v(\xi) \perp \begin{pmatrix} V_{11}(\xi) & \ldots & V_{1N}(\xi) \\ \vdots & & \vdots \\ V_{N1}(\xi) & \ldots & V_{NN}(\xi) \end{pmatrix} v(\xi) + \begin{pmatrix} \bar{c}_1(\xi) \\ -\bar{b}_1(\xi) \\ \vdots \\ \bar{c}_N(\xi) \\ -\bar{b}_N(\xi) \end{pmatrix} + \begin{pmatrix} w_1(\xi) \\ 0 \\ \vdots \\ w_N(\xi) \\ 0 \end{pmatrix} \geq 0, \tag{3.5}$$

where $\hat{\mathcal{N}} = \{v(\cdot) : \text{ The } x\text{-part of } v(\cdot) \text{ is independent of } \xi\}$, $\hat{\mathcal{M}} = \hat{\mathcal{N}}^{\perp}$ and

$$V_{jj}(\xi) = \begin{pmatrix} \bar{Q}_j(\xi) & -\bar{A}_j(\xi)^T \\ \bar{A}_j(\xi) & 0 \end{pmatrix}, \ \forall j \in [N] \text{ and } V_{jk}(\xi) = \begin{pmatrix} \bar{R}_{jk}(\xi) & 0 \\ \bar{A}_{jk}(\xi) & 0 \end{pmatrix}, \ \forall j \neq k, \ j,k \in [N].$$

To obtain a matrix with an easier-understood structure, we arrange the order of the variables as follows. Put all dual variables corresponding to the first-stage constraints together, followed by the dual variables corresponding to the second-stage constraints and denote the entire dual vector as $\eta(\cdot)$. In addition, denote

$$w(\xi) = (w_1(\xi), \ldots, w_N(\xi))^T.$$

Then (3.5) becomes

$$\begin{aligned} &\exists\ z(\cdot) \in \mathcal{N} \text{ and } w(\cdot) \in \mathcal{M} \text{ with dual variable } \eta(\cdot) \text{ such that } \forall \xi \in \Xi, \\ &0 \leq \begin{pmatrix} z(\xi) \\ \eta(\xi) \end{pmatrix} \perp \begin{pmatrix} H_{11}(\xi) & H_{12}(\xi) \\ H_{21}(\xi) & 0 \end{pmatrix} \begin{pmatrix} z(\xi) \\ \eta(\xi) \end{pmatrix} + \begin{pmatrix} \bar{c}(\xi) \\ -\bar{b}(\xi) \end{pmatrix} + \begin{pmatrix} w(\xi) \\ 0 \end{pmatrix} \geq 0, \end{aligned} \tag{3.6}$$

where

$$H_{11}(\xi) = \begin{pmatrix} 0 & R_{12} & \ldots & R_{1N} & P_{11}(\xi)^T & & & \\ R_{21} & 0 & \ldots & R_{2N} & & P_{22}(\xi)^T & & \\ \vdots & \vdots & \vdots & \vdots & & & \ddots & \\ R_{N1} & R_{N2} & \ldots & 0 & & & & P_{NN}(\xi)^T \\ P_{11}(\xi) & P_{12}(\xi) & \ldots & P_{1N}(\xi) & O_{11}(\xi) & O_{12}(\xi) & \ldots & O_{1N}(\xi) \\ P_{21}(\xi) & P_{22}(\xi) & \ldots & P_{2N}(\xi) & O_{21}(\xi) & O_{22}(\xi) & \ldots & O_{2N}(\xi) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{N1}(\xi) & P_{N2}(\xi) & \ldots & P_{NN}(\xi) & O_{N1}(\xi) & O_{N2}(\xi) & \ldots & O_{NN}(\xi) \end{pmatrix}, \tag{3.7}$$

$$H_{12}(\xi) = \begin{pmatrix} -A_1^T & & & & -D_1(\xi)^T & & & \\ & -A_2^T & & & & -D_2(\xi)^T & & \\ & & \ddots & & & & \ddots & \\ & & & -A_N^T & & & & -D_N(\xi)^T \\ & & & & -B_1(\xi)^T & & & \\ & & & & & -B_2(\xi)^T & & \\ & & & & & & \ddots & \\ & & & & & & & -B_N(\xi)^T \end{pmatrix},$$

and

$$H_{21}(\xi) = \begin{pmatrix} A_1 & A_2 & \ldots & A_N & & & & \\ A_1 & A_2 & \ldots & A_N & & & & \\ \vdots & \vdots & \vdots & \vdots & & & & \\ A_1 & A_2 & \ldots & A_N & & & & \\ D_1(\xi) & D_{12}(\xi) & \ldots & D_{1N}(\xi) & B_1(\xi) & B_{12}(\xi) & \ldots & B_{1N}(\xi) \\ D_{21}(\xi) & D_2(\xi) & \ldots & D_{2N}(\xi) & B_{21}(\xi) & B_2(\xi) & \ldots & B_{2N}(\xi) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{N1}(\xi) & D_{N2}(\xi) & \ldots & D_N(\xi) & B_{N1}(\xi) & B_{N2}(\xi) & \ldots & B_N(\xi) \end{pmatrix}.$$

The blank parts of the matrices are all zeros.

To summarize, we have shown the following result. Let $\mathcal{C} = \{z(\cdot) \in \mathscr{L} : z_j(\cdot) \in \mathcal{C}_j(z_{-j}), \forall j\}$ and $\mathcal{N} = \mathcal{N}_1 \times \ldots \times \mathcal{N}_N$.

Theorem 3.1. *Under the constraint qualification that $\mathcal{C} \cap \mathcal{N} \neq \emptyset$, the problem of finding a Nash equilibrium of supplier j can be converted to a stochastic linear complementarity problem (3.6).*

For the existence of solution to monotone SLC problem, we have the following result.

Theorem 3.2 (Theorem 3 of [31])**.** *When applied to a monotone stochastic linear complementarity problem, the progressive hedging algorithm (see next section for detail) can be executed by solving in each iteration a strongly monotone linear complementarity problem having the special property that the sequences $\{z^\nu(\cdot)\}$ and $\{w^\nu(\cdot)\}$, $\nu = 1, 2, ...$, thereby generated in $\mathcal{N}$ and $\mathcal{M}$ are sure to converge to a solution pair $(z^*(\cdot), w^*(\cdot))$ of a Nash equilibrium at a linear rate.*

Remark 3.3. Theorem 3.1 and Theorem 3.2 says that CQ plus monotonicity will guarantee the existence of a solution and the solution can be found by the progressive hedging algorithm. To guarantee the monotonicity, at least the matrix $H_{11}(\xi)$ has to be positive semidefinite.

It is generally not easy to obtain a sufficient condition for $H_{11}(\xi)$ to be positive semidefinite in practice. However, some specially structured $H_{11}(\xi)$ may be easily argued to be positive semidefinite. Consider the four blocks of $H_{11}(\xi)$. Notice that, if $p_{tj} - \gamma_{tj} + \beta_{tj}$ are all equal for every $t = 1, \ldots, M$, then $R_{jk} = -R_{kj}$, which leads the upper-left block of matrix $H_{11}(\xi)$ to be skew-symmetric. Actually, this condition is reasonable since the price that each supplier offers is determined by her own cost, and every one follows the same rule to offer her selling price. Moreover, if $P_{jk}(\xi) = 0$ for every $j, k \in [N]$, then $H_{11}(\xi)$ is positive semidefinite when its lower-right block is positive semidefinite.

Remark 3.4. Particularly, if the recourse problem is a linear program, then

$$\phi_j(y_j(\xi); x(\xi), y_{-j}(\xi), \xi) = d_j(\xi)^T y_j(\xi),$$

and only the upper-left block of $H_{11}(\xi)$ is nonzero. In this case, if $R_{jk} = -R_{kj}$, then one can observe that $H_{11}(\xi)$ is positive semidefinite, and as a result, SLC problem (3.6) is a monotone problem.

Remark 3.5. Specifically, if the second-stage constraint (2.10) has the following form:

$$Y_j(x, y_{-j}(\xi), \xi) = \left\{ y_j \in \mathbb{R}^M_+ : \begin{array}{c} F_j(\xi)x_j + G_j(\xi)y_j(\xi) \geq f_j(\xi), \\ \sum\limits_{k=1}^N S_k(\xi)x_k + \sum\limits_{k=1}^N T_k(\xi)y_k(\xi) \geq g(\xi) \end{array} \right\}, \tag{3.8}$$

in which the second constraint involving strategies of all the players has a uniform formulation for every j, together with the first-stage constraint (2.14), then for every ξ the whole constraint of $z_j(\xi)$ (2.15) becomes

$$C_j(z_{-j}(\xi), \xi) = \left\{ z_j(\xi) : z_j(\xi) \in \bar{C}_j(\xi), \bar{S}(\xi)z(\xi) \geq \bar{g}(\xi) \right\},$$

where $\bar{C}_j(\xi) := \{z_j(\xi) : z_j(\xi) \geq 0,\ [F_j(\xi) \quad G_j(\xi)]z_j(\xi) \geq f_j(\xi)\}$, and

$$\bar{S}(\xi) = \begin{pmatrix} A_1 & \ldots & A_N & 0 & \ldots & 0 \\ S_1(\xi) & \ldots & S_N(\xi) & T_1(\xi) & \ldots & T_N(\xi) \end{pmatrix},\ \bar{g}(\xi) = \begin{pmatrix} a \\ g(\xi) \end{pmatrix}.$$

In such case, $H_{12}(\xi)$ and $H_{21}(\xi)$ in condition (3.6) have the following specific structures:

$$H_{21}(\xi) = -H_{12}(\xi)^T = \begin{pmatrix} A_1 & \dots & A_N & 0 & & 0 \\ S_1(\xi) & \dots & S_N(\xi) & T_1(\xi) & \dots & T_N(\xi) \\ F_1(\xi) & & & G_1(\xi) & & \\ & \ddots & & & \ddots & \\ & & F_N(\xi) & & & G_N(\xi) \end{pmatrix},$$

and $\bar{b}(\xi)$ becomes

$$\bar{b}(\xi) = (a^T, g(\xi)^T, f_1(\xi), \dots, f_N(\xi))^T.$$

Therefore, the semidefiniteness of matrix $H_{11}(\xi)$ determines the monotonicity of the SLC problem (3.6).

$\boxed{4}$ Finding an Equilibrium via Progressive Hedging

The progressive hedging algorithm (PHA for short) was originally designed by [33] for multi-stage stochastic minimization problems and it has been used in various fields such as financial engineering [3], supply chain network design [17, 26] and surgery planning [14]. Recently, it has been extended to the monotone SVI problems in [31, 32] that is aimed at the inclusion problem in $\mathscr{L}$

$$\text{Find } z(\cdot) \in \mathcal{N},\ w(\cdot) \in \mathcal{M},\ \text{such that } \ -\mathcal{F}(z(\cdot)) \in N_{\mathcal{C}}(z(\cdot)) + w(\cdot), \tag{4.1}$$

where $\mathcal{F}$ is a point-to-set continuous mapping. The SLC problem is a special case of (4.1), where $\mathcal{C} \equiv$ the nonnegative orthant and $\mathcal{F}$ is defined by the linear mapping $\xi \to S(\xi)z(\xi) + q(\xi)$. Then the Problem (4.1) becomes the SLC problem

$$\text{Find } z(\cdot) \in \mathcal{N},\ w(\cdot) \in \mathcal{M},\ \text{such that } \ 0 \le z(\cdot) \perp \mathcal{F}(z(\cdot)) + w(\cdot) \ge 0, \tag{4.2}$$

where $\mathcal{F}$ is a linear mapping of $\mathscr{L} \to \mathscr{L}$. We notice that our suppliers' game (3.6) is in particular in this form.

$\boxed{4.1}$ The ideas for solving the SLC problem

Since the dimension of $z(\cdot)$ is very large, it is almost impossible to directly solve the problem (4.2). The progressive hedging algorithm is a scenario-based decomposition scheme with a projection step for resuming nonanticipativity [37]. Let us note that, by ignoring the nonanticipativity, for given $w(\cdot)$, it is easy to find $z(\cdot)$ such that

$$0 \le z(\cdot) \perp \mathcal{F}(z(\cdot)) + w(\cdot) \ge 0, \tag{4.3}$$

because the task is equivalent to that for each ξ find the solution to the problem

$$0 \le z(\xi) \perp S(\xi)z(\xi) + q(\xi) + w(\xi) \ge 0. \tag{4.4}$$

Since (4.4) is of normal size, it is easy to solve for every ξ. This is the basic decomposition idea of progressive hedging. However, even if $S(\xi)$ is positive semidefinite, there may not be a solution to (4.4); therefore, the PHA adds another trick to this decomposition scheme to at least guarantee a unique solution to (4.4); that is, adding a proximal term to it by solving

$$0 \le z(\xi) \perp S(\xi)z(\xi) + q(\xi) + w(\xi) + \sigma(z(\xi) - z^{\nu}(\xi)) \ge 0, \tag{4.5}$$

where $\sigma > 0$ is a fixed parameter and $z^\nu(\xi)$ is the current iteration in the iterative procedure of PHA.

The solution $\hat{z}(\cdot)$ obtained by separately solving (4.5) for all $\xi \in \Xi$, may not be in the space $\mathcal{N}$. Therefore, the next step of PHA is to restore the nonanticipativity by projecting $\hat{z}(\cdot)$ to space $\mathcal{N}$, i.e.

$$z^{\nu+1}(\cdot) = \mathbb{P}_{\mathcal{N}}(\hat{z}(\cdot)) \text{ and } w^{\nu+1}(\cdot) = w^\nu(\cdot) + \sigma \mathbb{P}_{\mathcal{M}}(\hat{z}(\cdot)), \tag{4.6}$$

where $\mathbb{P}$ is the projection operator. The projection $\mathbb{P}_{\mathcal{N}}$ accounts to computing the expectation of $\hat{z}(\xi)$ in the two-stage setting and is therefore an easy job, while the projection $\mathbb{P}_{\mathcal{M}} = \mathbb{I} - \mathbb{P}_{\mathcal{N}}$, where $\mathbb{I}$ is the identity operator. With the new pair of $(z^{\nu+1}(\cdot), w^{\nu+1}(\cdot))$, the PHA proceeds as an iterative method until the generated sequence converges.

$\boxed{4.2}$ The PHAs

The PHA, when applied to monotone SLC problems, is fairly fast as demonstrated in [31]. Zhang et al. [40] studied the quadratic two-stage N-person noncooperative game under uncertainty, which include our model (3.6) as a special case. The problem of finding a Nash equilibrium of the game is shown to be equivalent to a generally nonmonotone SLC problem. They proposed specially designed progressive hedging algorithms to solve SLC problems in both monotone and nonmonotone but “elicited monotone” cases. We next present their results without proof.

Algorithm 1. PHA for SLC problem (3.6) in monotone case

Initiation. Let parameter $\sigma > 0$. Set $z^0(\xi) = 0$, $\eta^0(\xi) = 0$, $w^0(\xi) = 0$ for all ξ, and $\nu = 0$.

Iterations.

Step 1. For each $\xi \in \Xi$, obtain $\hat{z}^\nu(\xi) = (\hat{x}^\nu(\xi), \hat{y}^\nu(\xi))$ and $\hat{\eta}^\nu(\xi))$ via solving the following linear complementarity problem

$$0 \le \begin{pmatrix} z \\ \eta \end{pmatrix} \perp \begin{pmatrix} H_{11}(\xi) & H_{12}(\xi) \\ H_{21}(\xi) & 0 \end{pmatrix} \begin{pmatrix} z \\ \eta \end{pmatrix} + \begin{pmatrix} \bar{c}(\xi) \\ -\bar{b}(\xi) \end{pmatrix} + \begin{pmatrix} w^\nu(\xi) \\ 0 \end{pmatrix} + \sigma \begin{pmatrix} z - z^\nu(\xi) \\ \eta - \eta^\nu(\xi) \end{pmatrix} \ge 0.$$

Step 2. (Primal Update) For each $\xi \in \Xi$,

$$z^{\nu+1}(\xi) = \begin{pmatrix} \mathbb{E}_\xi(\hat{x}^\nu(\xi)) \\ \hat{y}^\nu(\xi) \end{pmatrix}, \ \eta^{\nu+1}(\xi) = \hat{\eta}^\nu(\xi).$$

Step 3. (Dual Update) $w^{\nu+1}(\xi) = w^\nu(\xi) + \sigma \left[\hat{z}^\nu(\xi) - z^{\nu+1}(\xi)\right]$.

Set $\nu := \nu + 1$, **repeat** until a stopping criterion is met.

We need a definition to present the convergence result for the nonmonotone case.

Definition 4.1 ([30]). Monotonicity of $\mathcal{R}$ is said to be elicitable (or elicited) at level $\rho > 0$:

- globally if $\mathcal{R} + \rho\mathbb{P}_{\mathcal{M}}$ is maximal monotone globally, where $\mathbb{P}_{\mathcal{M}}$ is the projection onto subspace $\mathcal{M}$, and
- locally around $(z, y) \in \operatorname{graph}(\mathcal{R})$ with $z \in \mathcal{N}, y \in \mathcal{M}$, if $\mathcal{R}+\rho\mathbb{P}_{\mathcal{M}}$ is maximal monotone locally around (z, y).

[40] obtained the following result for nonmonotone SLC problems. Denote

$$H(\xi)=\begin{pmatrix}H_{11}(\xi) & H_{12}(\xi)\\ H_{21}(\xi) & 0\end{pmatrix}.$$

Let $\operatorname{diag}(H(\xi))$ be the block-diagonal matrix, consisting of diagonal blocks $H(\xi)$ for all $\xi\in\Xi$ and let $\operatorname{diag}(\mathbf{H}(\xi))$ be its symmetric part, i.e.,

$$\operatorname{diag}(\mathbf{H}(\xi)) := [\operatorname{diag}(H(\xi)) + \operatorname{diag}(H(\xi))^T]/2.$$

From Corollary 3.6 in [40], if $\operatorname{diag}(\mathbf{H}(\xi))$ is positive definite on $\mathcal{N}$, then $\mathcal{F}+N_{\mathcal{C}}+\rho\mathbb{P}_{\mathcal{M}}$ is maximal monotone for some large $\rho>0$, thus SLC problem (3.6) is globally elicitable and the following Algorithm 2 will produce series $\{(z^\nu(\cdot),w^\nu(\cdot))\}$ that converges linearly to a solution $(z^*(\cdot),w^*(\cdot))$ with respect to the (σ,ρ)-norm defined by

$$\|(z(\cdot),w(\cdot))\|^2_{\sigma,\rho}=\|z(\cdot)\|^2+\frac{1}{\sigma(\sigma-\rho)}\|w(\cdot)\|^2,$$

if the game has a solution and satisfies the constraint qualification.

Algorithm 2. Elicited PHA for SLC problem (3.6)

Initiation. Let parameter $\sigma>\rho>0$. Set $z^0(\xi)=0$, $\eta^0(\xi)=0$, $w^0(\xi)=0$ for all ξ, and $\nu=0$.

Iterations.

Step 1. For each $\xi\in\Xi$, obtain $\hat{z}^\nu(\xi)=(\hat{x}^\nu(\xi),\hat{y}^\nu(\xi))$ and $\hat{\eta}^\nu(\xi))$ via the following LCP

$$0\le\begin{pmatrix}z\\ \eta\end{pmatrix}\perp\begin{pmatrix}H_{11}(\xi) & H_{12}(\xi)\\ H_{21}(\xi) & 0\end{pmatrix}\begin{pmatrix}z\\ \eta\end{pmatrix}+\begin{pmatrix}\bar{c}(\xi)\\ -\bar{b}(\xi)\end{pmatrix}+\begin{pmatrix}w^\nu(\xi)\\ 0\end{pmatrix}+\sigma\begin{pmatrix}z-z^\nu(\xi)\\ \eta-\eta^\nu(\xi)\end{pmatrix}\ge 0.$$

Step 2. (Primal Update) For each $\xi\in\Xi$,

$$z^{\nu+1}(\xi)=\begin{pmatrix}\mathbb{E}_\xi(\hat{x}^\nu(\xi))\\ \hat{y}^\nu(\xi)\end{pmatrix},\ \eta^{\nu+1}(\xi)=\hat{\eta}^\nu(\xi).$$

Step 3. (Dual Update) $\quad w^{\nu+1}(\xi)=w^\nu(\xi)+(\sigma-\rho)\left[\hat{z}^\nu(\xi)-z^{\nu+1}(\xi)\right].$

Set $\nu:=\nu+1$, **repeat** until a stopping criterion is met.

Theorem 4.2. (Convergence of Algorithm 2, Theorem 3.3 of [40] and Theorem 2.1 of [38]) *Suppose that $\mathcal{F}+N_{\mathcal{C}}$ is globally elicitable at level ρ. Then in the case that $\mathcal{F}$ is linear, if $\mathcal{N}\cap\mathscr{L}_+\neq\emptyset$ ($\mathscr{L}_+$ is the nonnegative orthant of $\mathscr{L}$) and the SLC problem has a solution, then the sequence $\{z^\nu(\cdot),w^\nu(\cdot)\}$ generated by Algorithm 2 will globally converge to some pair $\{z^*(\cdot),w^*(\cdot)\}$ with $(z^*(\cdot),w^*(\cdot))$ being a solution to (4.1) at linear rate with respect to the (σ,ρ)-norm.*

5 Numerical Experiments

In this section, to test the effectiveness of PHAs in Algorithms 1 and 2, we conduct the following three experiments for both monotone problems and nonmonotone-but-elicited-monotone (nonmonotone for short) problems.

- Explore the choices of parameters in PHAs;

- Test how PHAs work when the number of scenarios increases;
- Test how PHAs work when the number of players ($[M, N]$) increases.

All numerical experiments are coded in Matlab R2015b and run on a laptop with an Intel(R) Core(TM) I7-7500U 2.90GHz CPU and 16 GB of RAM under WINDOWS 10 operating system.

As we indicted in Section 1, the purpose of the computational test is to show that the PHAs work effectively for randomly generated medium sized manufacturer-supplier games. Since PHA is the first algorithm proposed for SVI problems, currently, there is no other method to compare with. Some well-known algorithms in stochastic optimization such as the L-shaped method [39,41] appears not applicable to equilibrium problems for our comparison, in particular for the nonlinear and nonconvex setting of the proposed manufacturer-supplier games.

5.1 Data generation

In order to construct the monotonicity of the test problems, we generate the manufacturer-supplier game (2.16) according to Remarks 1 and 3 in Section 3. Specifically, the suppliers' second-stage constraints is generated in the form of (3.8), and the relevant data are randomly generated by the rules in Table 1, in which the expression $v = \text{rand}(l, u)$ (or $V = \text{rand}(l, u)$) means v (or V) is randomly generated as a vector (or matrix) with its entries being uniformly distributed in the interval (l, u).

Table 1. List of data values

Parameters of the first stage
$\Delta_i = 100$, $r_i \in (2N, 5N)$, $h_i \in (0.1, 0.5)$, $\forall\ i \in [M]$
$p_{ij} \in (2, 4)$, $p_{ij} - \gamma_{ij} - \beta_{ij} \in (1, 2)$, $\Gamma_{ij} \in (0.5, 1)$, $\forall\ i \in [M]$, $j \in [N]$
ε =1e-6
Parameters of the second stage (for every $\xi \in \Xi$)
$P_{jk}(\xi) = 0$, $\forall\ j, k \in [N]$
the lower-right block of $H_{11}(\xi)$ is generated as a positive definite matrix
by $\hat{O}(\xi)^T \hat{O}(\xi)$, where $\hat{O}(\xi) = \text{rand}(0, 1)$ is in size $MN \times MN$
$d_j(\xi) = \text{rand}(-1, 1)$, $\forall\ j \in [N]$
$F_j(\xi) = \text{rand}(-1, 0)$, $G_j(\xi) = \text{rand}(0, 1)$ are in size $(\lfloor N/2 \rfloor + 1) \times M$, $\forall\ j \in [N]$
$S_j(\xi) = \text{rand}(-1, 1), T_j(\xi) = \text{rand}(-1, 1)$ are in size $(\lfloor N/2 \rfloor + 1) \times M$, $\forall\ j \in [N]$
$f_j(\xi)$, $g(\xi)$ are random vectors to guarantee the feasibility, $\forall\ j \in [N]$

Following Remarks 1 and 3, when $P_{jk}(\xi) = 0\ \forall\ j, k \in [N]$ and the lower-right block of $H_{11}(\xi)$ is positive semidefinite, we can obtain monotone problems by letting

$$p_{i1} - \gamma_{i1} - \beta_{i1} = \cdots = p_{iN} - \gamma_{iN} - \beta_{iN} \text{ for all } i \in [M]. \tag{5.1}$$

Besides, when $P_{jk}(\xi) = 0\ \forall\ j, k \in [N]$ and the lower-right block of $H_{11}(\xi)$ is positive semidefinite, it is not difficult to see $H_{11}(\xi)$ is elicitable monotone at some level $\rho > 0$. Therefore, in our experiments, we generate nonmonotone but elicitable monotone problems following the rules in Table 1; as well as monotone problems following the rules in Table

1 and make (5.1) holds, particularly with $O_{jk}(\xi) = 0$ for $j \neq k$ and $O_{jj}(\xi)$ generated by $\hat{O}_{jj}(\xi)^T \hat{O}_{jj}(\xi)$, where $\hat{O}_{jj}(\xi) = \text{rand}(0,1)$ is of size $M \times M$.

5.2 Stopping criteria

Re-partition matrix $H(\xi)$ into four blocks $\begin{pmatrix} M_{11} & M_{12}(\xi) \\ M_{21}(\xi) & M_{22}(\xi) \end{pmatrix}$ with M_{11} being the upper-left block of $H_{11}(\xi)$ without the $O_{ij}(\xi)$s, which is a deterministic submatrix. The reason why we isolate M_{11} is to reflect the structure of nonanticipativity constraint and to separate the first-stage vector x, which is supposed to be a constant for every ξ. The vector $q(\xi)$ is divided into two blocks (q_1 and $q_2(\xi)$) according to the partition of $H(\xi)$. Set

$$\text{rel.err} = \max\{\text{rel.err}_1, \text{rel.err}_2\},$$

where

$$\text{rel.err}_1 = \frac{\|x - \prod_{\geq 0}(x - (M_{11}x + \mathbb{E}_\xi[M_{12}(\xi)\bar{y}(\xi)] + q_1))\|}{1 + \|x\|},$$

$$\text{rel.err}_2 = \max_\xi \left\{ \frac{\|\bar{y}(\xi) - \prod_{\geq 0}(\bar{y}(\xi) - (M_{21}(\xi)x + M_{22}(\xi)\bar{y}(\xi) + q_2(\xi)))\|}{1 + \|\bar{y}(\xi)\|} \right\},$$

with $(\prod_{\geq 0}(a))_j = \max\{a_j, 0\}$. Set the tolerance to be 10^{-5}, and the maximal iterations to be 2000, i.e., if rel.err $\leq 10^{-5}$ or iteration number ≥ 2000, the algorithm stops.

5.3 Numerical results for the choice of parameters

In addition to the parameter σ used in Algorithm 1, a step length τ is added in the dual update step, namely

$$w^{\nu+1}(\xi) = w^\nu(\xi) + \tau\sigma(\hat{z}^\nu(\xi) - z^{\nu+1}(\xi)). \tag{5.2}$$

Similarly, for Algorithm 2, we revise Step 3 as follows,

$$w^{\nu+1}(\xi) = w^\nu(\xi) + \tau(\sigma - \rho)\left[\hat{z}^\nu(\xi) - z^{\nu+1}(\xi)\right]. \tag{5.3}$$

It can be seen that (5.2) and (5.3) reduce to Step 3 in Algorithms 1 and 2 when $\tau = 1$. However, the choice of $\tau = 1.618$, which has been successfully used in Douglas-Rachford splitting methods, has achieved better performance in the numerical experiments of [40]. Thus, in this subsection, we will first check the impact of parameter σ and τ in Algorithm 1 for monotone problems, then fix τ and check the impact of parameters σ and ρ in Algorithm 2 for nonmonotone problems.

We set $[M, N] = [5, 5]$ and the number of scenarios (sn for short) at 10, then generate 10 monotone problems and 10 nonmonotone problems by the rules stated in Subsection 5.1. Fix step length $\tau = 1$ and $\tau = 1.618$ respectively, then we apply Algorithm 1 for solving the randomly generated monotone problems with parameter σ to be 0.5, 1, 2.5, $\sqrt{10}$ and 5, and record the number of iterations for convergence and the time (counting by seconds) for each problem. The average number of converging iteration (avg-iter for short) and the average time (avg-time) for convergence among these 10 problems are listed in Table 2 and drawn in Figure 1.

It can be seen from Table 2 and Figure 1 that the choice of σ is crucial for the speed of convergence. The default value $\sigma = 1$ results in nearly 2 times more iterations than a suitable selection of σ, while a larger or smaller σ slows down the convergence as well. For

Table 2: Performance of Algorithm 1 with different parameters ([M,N]=[5,5], sn=10)

σ	$\tau = 1$		$\tau = 1.618$	
	avg-iter	avg-time(s)	avg-iter	avg-time(s)
0.5	214	52.2	138	35.4
1	114	24.9	69	16.2
2.5	63	14.0	54	12.4
$\sqrt{10}$	69	17.1	67	15.7
5	104	24.3	103	22.8

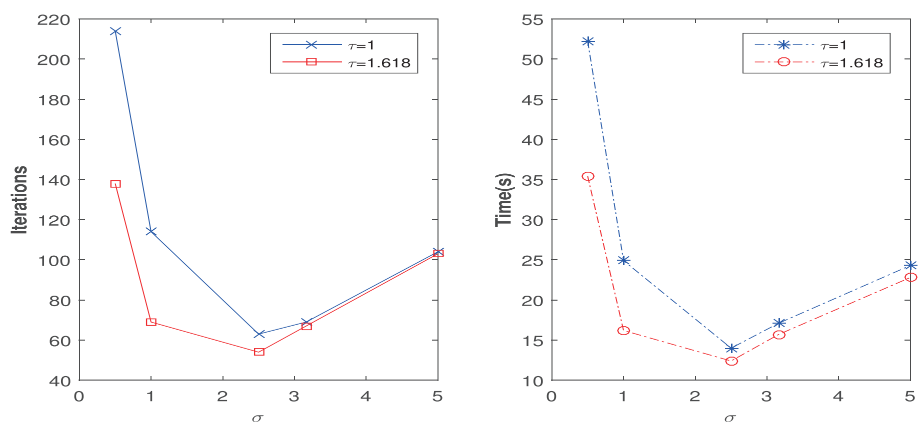


Figure 1: Performance of Algorithm 1 with different parameters

this group of monotone problems, a good heuristic for the selection of σ is 2.5, which seems to be $N/2$, i.e. half number of suppliers. Besides, Table 2 and Figure 1 show that the value of τ has limited impact on the speed of convergence. Generally, $\tau = 1.618$ can speed up 10% than $\tau = 1$ for this group of problems. Therefore, in the following experiments, we adopt $\tau = 1.618$ and use $\sigma = N/2$ when applying Algorithm 1 to solving monotone problems.

Fix $\tau = 1.618$, we apply Algorithm 2 for solving the randomly generated nonmonotone problems with different values of σ and ρ:

- fix $\rho = 25$, and set σ to be 50, 75, 100, 125, respectively;
- fix $\sigma = 50$, and set ρ to be 0, 5, 15, 25, respectively,

to see the impact of σ and ρ in Algorithm 2 on the speed of convergence for nonmonotne problems. Notice that the choice of σ in Algorithm 2 is much bigger than the suitable selection for monotone problems in Algorithm 1. This is because the matrix $H_{11}(\xi)$ is no longer semidefinite and the value of σ needs to contribute for the elicited monotonicity. The average number of converging iteration and the average time for convergence among these 10 nonmonotone problems are listed in Table 3 and are drawn in Figure 2.

Figure 2 indicates that the choice of σ has more influence than the choice of ρ on the performance of Algorithm 2 for nonmonotone problems. When fixing $\rho = 25$, larger σ makes slower convergence; while fixing $\sigma = 50$, different values of ρ have little difference on the convergence time and number of iterations. However, based on the number of iterations and

Table 3: Performance of Algorithm 2 with different parameters
([M,N]=[5,5], sn=10, $\tau = 1.618$)

$\rho = 25$			$\sigma = 50$		
σ	avg-iter	avg-time(s)	ρ	avg-iter	avg-time(s)
50	105	21.8	0	96	20.2
75	127	26.1	5	100	20.8
100	154	31.7	15	103	21.7
125	195	40.0	25	105	21.8

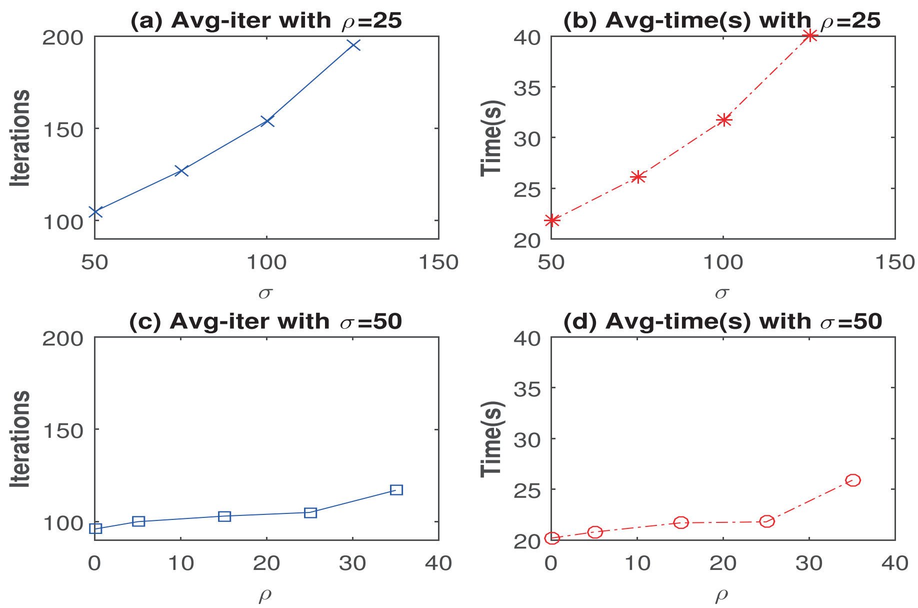


Figure 2: Performance of Algorithm 2 with different parameters

time among the ten nonmonotone problems, the best choice of σ differs from each other. In the following experiments, we use $\sigma = 10N$ and $\rho = \sigma/2$ when applying Algorithm 2 to solving nonmonotone problems.

$\boxed{5.4}$ Numerical results when the number of scenarios increases

In this subsection, we set $[M, N] = [5, 5]$ and sn=10, 20, 50, 100, 200, respectively, then for each setting of sn, generate 10 monotone problems and 10 nonmonotone problems by the rules stated in Subsection 5.1.

(1) Choosing $\sigma = 2.5$ and $\tau = 1.618$, apply Algorithm 1 for the randomly generated monotone problems and record the number of iterations and time for convergence, with corresponding results being listed in Table 4 and Figure 3;

(2) Choosing $\sigma = 50$, $\rho = 25$ and $\tau = 1.618$, apply Algorithm 2 for the randomly generated nonmonotone problems and record the number of iterations and time for convergence, with corresponding results being listed in Table 5 and Figure 4.

Table 4: Monotone results while sn increases
([M,N]=[5,5], $\sigma = 2.5$, $\tau = 1.618$)

sn	avg-iter	avg-time(s)
10	54	12.4
20	54	25.1
50	63	69.8
100	71	159.5
200	95	409.8

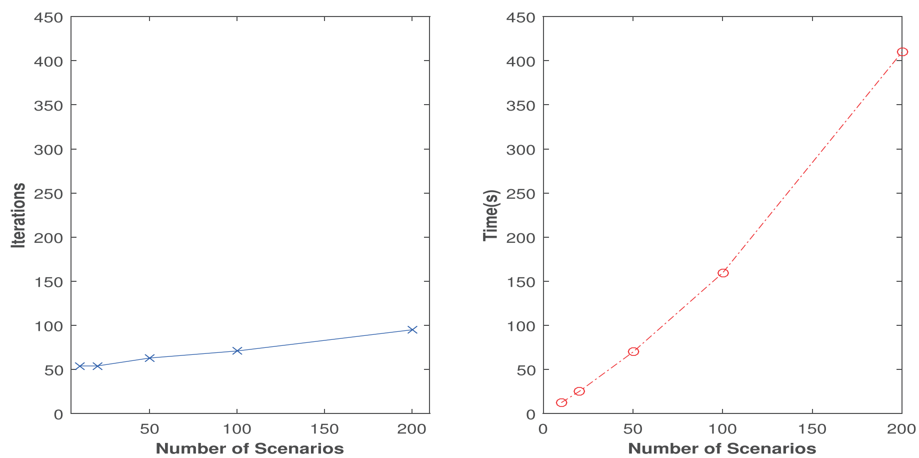


Figure 3: Convergence results of Algorithm 1 when number of scenarios increases

Table 4 and Figure 3 show that both number of iterations and time for convergence increase near linearly while the number of scenarios grows. Besides, it can be observed that the growth rate of number of iterations is comparatively steady, which coincides with the results in [40]. Hence, the reason for the rise of convergence time may be caused by the proportional growth of the number of subproblems for each iteration when sn increases. Thus, if using parallel computation for subproblems, which is a possible choice of future experiments, then the time for convergence could be expected to be stable.

Table 5: Nonmonotone results while sn increases
([M,N]=[5,5], $\sigma = 50$, $\rho = 25$, $\tau = 1.618$)

sn	avg-iter	avg-time(s)
10	105	21.8
20	108	43.5
50	136	138.9
100	139	284.8
200	158	649.6

It can be seen that the nonmonotone case in Figure 4 presents similar trend to the monotone case in Figure 3. The number of iterations for convergence is relatively stable, while the convergence time rises at a near linear rate when sn grows. However, compared with the monotone results, both number of iterations and time for converging in the nonmonotone

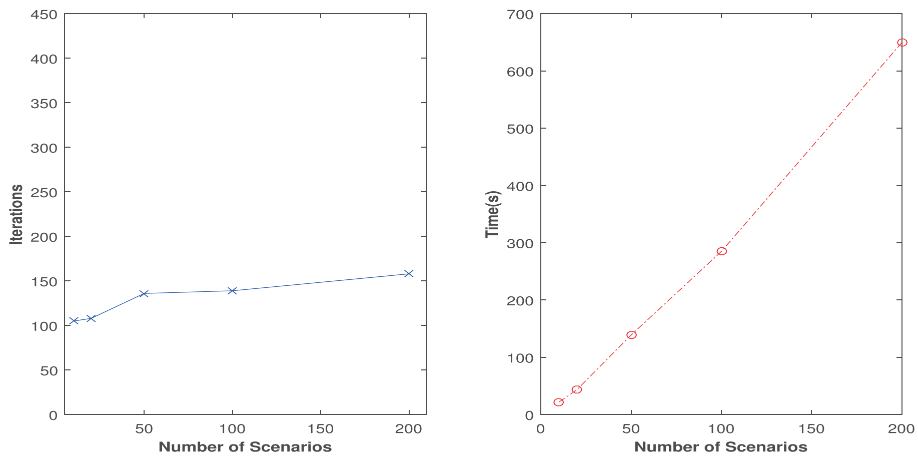


Figure 4: Convergence results of Algorithm 2 when number of scenarios increases

case are consistently nearly twice larger than those in the monotone case.

$\boxed{5.5}$ Numerical results when the number of players grows

In this subsection, we set sn=20 and respectively

- fix $N = 5$, increase M from 2 to 10;
- fix $M = 5$, increase N from 2 to 10.

For each setting of $[M, N]$, generate 10 monotone problems and 10 nonmonotone problems by the rules stated in Subsection 5.1.

(1) Choosing $\sigma = N/2$ and $\tau = 1.618$, apply Algorithm 1 for the randomly generated monotone problems and list results in Table 6 and Figure 5;

(2) Choosing $\sigma = 10N$, $\rho = \sigma/2$ and $\tau = 1.618$, apply Algorithm 2 for the randomly generated nonmonotone problems and list results in Table 7 and Figure 6.

It is shown in Table 6 and Figure 5 that when we fix N and increase M, or vice versa, the number of iterations that Algorithm 1 converges to a solution has no significant change, while the average time for convergence grows slowly.

Table 7 and Figure 6 illustrate that when applying Algorithm 2 to solving nonmonotone problem with parameter $\sigma = 10N$ and $\rho = \sigma/2$, the convergence speed will be slowed down at a bigger rate while rising either M or N. However, it can be observed by comparison of Figure 5 and Figure 6 that the shape of the increase trend is similar between monotone problems solved by Algorithm 1 and nonmonotone problems solved via Algorithm 2. The difference is only on the slope of the increasing trend.

Table 6: Monotone results while [M,N] increases
(sn=20, $\sigma = N/2$, $\tau = 1.618$)

Fixed $N = 5$			Fixed $M = 5$		
M	avg-iter	avg-time(s)	N	avg-iter	avg-time(s)
2	57	19.3	2	3	1.2
3	54	19.3	3	38	12.6
4	50	19.4	4	50	19.9
5	54	25.1	5	54	25.1
6	55	25.6	6	62	27.3
7	69	33.3	7	60	28.5
8	67	33.7	8	62	35.1
9	78	42.1	9	63	37.0
10	91	54.3	10	68	45.7

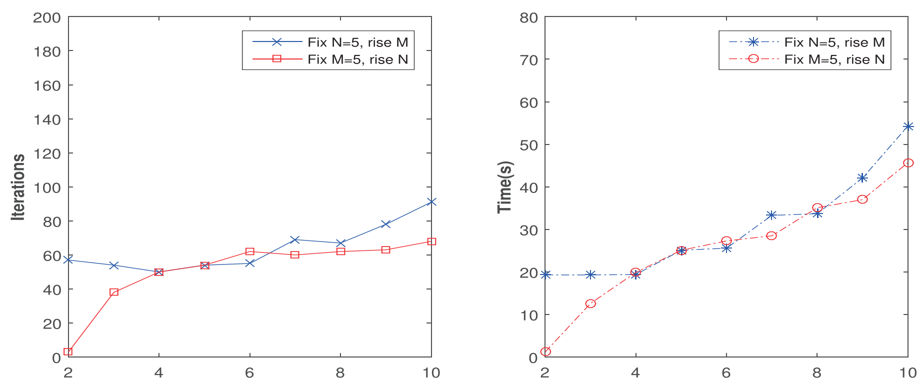


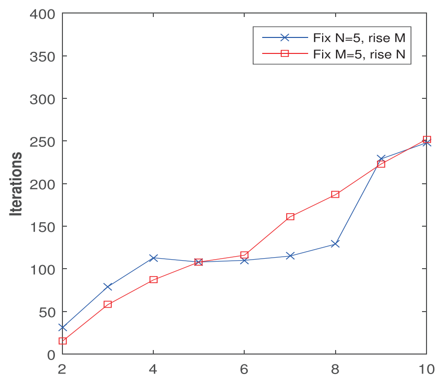
Figure 5: Convergence results of Algorithm 1 when $[M, N]$ increases

Table 7: Nonmonotone results while [M,N] increases
(sn=20, $\sigma = 10N$, $\rho = \sigma/2$, $\tau = 1.618$)

Fixed $N = 5$			Fixed $M = 5$		
M	avg-iter	avg-time(s)	N	avg-iter	avg-time(s)
2	31	10.5	2	15	5.0
3	79	27.6	3	58	18.7
4	113	42.9	4	87	34.3
5	108	43.5	5	108	43.5
6	110	54.3	6	116	64.8
7	115	64.3	7	161	95.7
8	129	79.4	8	187	135.5
9	229	152.5	9	223	188.8
10	248	185.0	10	252	265.5

6 Conclusion

This paper studies a model of quadratic two-stage manufacturer-supplier game under uncertainty. In addition to allowing all parameters in the second stage to be random, this model allows all kind of cross terms to appear in the objective functions of both stages.



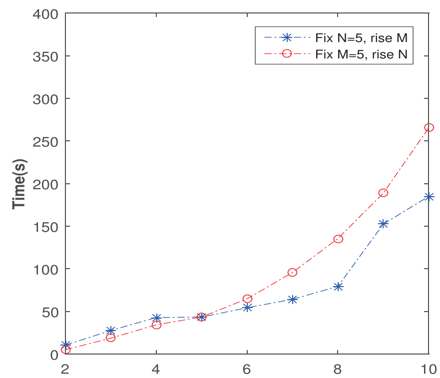


Figure 6: Convergence results of Algorithm 2 when $[M, N]$ increases

The problem of finding a Nash equilibrium of this model is converted to a stochastic linear complementarity problem that appears in the literature of stochastic optimization only recently.

It is explained in detail that the progressive hedging scheme can be used in solving the suppliers' game model for both the monotone case and the elicited monotone case. Preliminary numerical experiments are conducted and the results indicate that the specially tailored progressive hedging algorithms for stochastic linear complementarity problems are effective for the manufacturer-supplier game in moderate size. The experiment results also provided heuristic choices on the parameters of the algorithms.

References

[1] J. Ang, M. Fukushima, F. Meng, T. Noda and J. Sun, Establishing Nash equilibrium of the manufacturer-supplier game in supply chain management, *J. Global Optim.* 56 (2013) 1297–1312.

[2] J. Ang, F. Meng and J. Sun, Two-stage stochastic linear programs with incomplete information on uncertainty, *Eur. J. Oper. Res.* 233 (2014) 16–22.

[3] D. Barro and E. Canestrelli, Dynamic portfolio optimization: Time decomposition using the Maximum Principle with a scenario approach, *Eur. J. Oper. Res.* 163(1) (2005) 217–229.

[4] S. Benjaafar, E. Elahi and K.L. Donohue, Outsourcing via service competition, *Manag. Sci.*, 53 (2007) 241–259.

[5] F. Bernstein and A. Federgruen, Coordination mechanisms for supply chains under price and service competition, *Manufac. Serv. Oper. Manag.* 9 (2007) 242–262.

[6] G. Cachon and P. Harker, Competition and outsourcing with scale economies, *Manag. Sci.* 48 (2002) 1314–1333.

[7] G. Cachon and S. Netessine, Game theory in supply chain analysis. In *Handbook of Quantitative Supply Chain Analysis: Modeling in the eBusiness Era*, D. Smichi-Levi, S.D. Wu and Z. Shen (eds.), Kluwer, Boston, 2004, pp 13–66.

[8] G. Cachon and F. Zhang, Obtaining fast service in a queueing system via performance-based allocation of demand, *Manag. Sci.* 53 (2007) 408–420.

[9] V.M. Charitopoulos, V. Dua, J.M. Pinto and L.G. Papageorgiou, A game-theoretic optimisation approach to fair customer allocation in oligopolies, *Optim. Eng.* 21 (2020) 1459–1486.

[10] Y.Z. Chen, X. Cheng, J. Li and L. He, A multi-level programming for shale gas-water supply chains accounting for tradeoffs between economic and environmental concerns, *Comp. Chem. Eng.* 135 (2020) Article Number: 106761.

[11] W. Chen, M. Sim, J. Sun and C.P. Teo, From CVaR to uncertainty set: Implications in joint chance constrained optimization, *Oper. Res.* 58 (2010) 470–485.

[12] A. Federgruen and A. Heching, Combined pricing and inventory control under uncertainty, *Oper. Res.* 47 (1999) 454–475.

[13] J.Y. Gao and F.Q. You, A stochastic game theoretic framework for decentralized optimization of multi-stakeholder supply chains under uncertainty, *Comp. Chem. Eng.* 122 (2019) 31–46.

[14] S. Gul, B.T. Denton and J.W. Fowler, A progressive hedging approach for surgery planning under uncertainty, *INFORMS J. Comput.* 27(4) (2015) 597–808.

[15] A. Ha, L. Li and S.-M. Ng, Price and delivery logistics competition in a supply chain, *Manag. Sci.* 49 (2003) 1139–1153.

[16] J. Hall and E. Porteus, Customer service competition in capacitated systems, *Manufac. Serv. Oper. Manag.* 2 (2000) 144–165.

[17] M. Jeihoonian, M.K. Zanjani and M. Gendreau, Dynamic reverse supply chain network design under uncertainty: mathematical modeling and solution algorithm, *Int. Trans. Oper. Res.* (2020) Published online doi: 10.1111/itor.12865.

[18] E. Kalai, M. Kamien and M. Rubinovitch, Optimal service speeds in a competitive environment, *Manag. Sci.* 38 (1992) 1154–1163.

[19] B. Li, Y. Rong, J. Sun and K.L. Teo, A distributionally robust linear receiver design for multi-access space-time block coded MIMO systems, *IEEE Trans. Wireless Commun.* 16(1) (2017) 464–474.

[20] B. Li, J. Sun, H. Xu and M. Zhang, A class of two-stage distributionally robust games, *J. Ind. Manag. Optim.* 15(1) (2019) 387–400.

[21] L. Li, The role of inventory in delivery-time competition, *Manag. Sci.* 38 (1992) 182–197.

[22] L. Li and Y.S. Lee, Pricing and delivery-time performance in a competitive environment, *Manag. Sci.* 40 (1994) 633–646.

[23] C. Loch, *Time competition is capability competition*, Working Paper, INSEAD, 1994.

[24] S. Namany, T. Al-Ansari and R. Govindan, Optimisation of the energy, water, and food nexus for food security scenarios, *Comp. Chem. Eng.* 129 (2019) Article Number: UNSP 106513.

[25] J. Nicoletti and F.Q. You, Multiobjective economic and environmental optimization of global crude oil purchase and sale planning with noncooperative stakeholders, *Appl. Energ.* 259 (2020) Article Number: 114222.

[26] F. Nur, M. Aboytes-Ojeda, K.K. Castillo-Villar and M. Marufuzzaman, A two-stage stochastic programming model for biofuel supply chain network design with biomass quality implications, *IISE Trans.* 53(8) (2021) 845-–868.

[27] J.-S. Pang, S. Sen, and U. Shanbhag, Two-stage non-cooperative games with risk-averse players, *Math. Program.* 165 (2017) 235–290.

[28] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, NJ, 1970.

[29] R.T. Rockafellar, *Network Flows and Monotropic Optimization*, Weley, New York, 1984.

[30] R.T. Rockafellar, Progressive decoupling of linkages in optimization and variational inequalities with elicitable convexity or monotonicity, *Set-Valued Var. Anal.* 27 (2019) 863–893.

[31] R.T. Rockafellar and J. Sun, Solving monotone stochastic variational inequalities and complementarity problems by progressive hedging, *Math. Program.* 174 (2019) 453–471.

[32] R.T. Rockafellar and J. Sun, Solving Lagrangian variational inequalities with applications to stochastic programming, *Math. Program.* 181 (2020) 435–451.

[33] R.T. Rockafellar and R.J-B. Wets, Scenarios and policy aggregation in optimization under uncertainty, *Math. Oper. Res.* 16 (1991) 119–147.

[34] R.T. Rockafellar and R.J-B. Wets, Stochastic variational inequalities: single-stage to multistage, *Math. Program.* 165 (2017) 291–330.

[35] T. Santoso, S. Ahmed, M. Goetschalckx and A. Shapiro, A stochastic programming approach for supply chain network design under uncertainty, *European J. Oper. Res.* 167 (2005) 96–115.

[36] K. So and J.S. Song, Price, delivery time guarantees and capacity selection, *European J. Oper. Res.* 111 (1998) 28–49.

[37] J. Sun, H. Xu and M. Zhang, A new interpretation of the progressive hedging algorithm for multistage stochastic minimization problems, *J. Industrial Manag. Optim.* 16(4) (2020) 1655–1662.

[38] J. Sun and M. Zhang, The elicited progressive decoupling algorithm: A note on the rate of convergence and a preliminary numerical experiment on the choice of parameters, *Set-Valued Var. Anal.* (2021) https://doi.org/10.1007/s11228-021-00613-0.

[39] R.M. Van Slyke and R.J-B. Wets, L-shaped linear programs with applications to optimal control and stochastic programming, *SIAM J. Appl. Math.* 17 (1969) 638–663.

[40] M. Zhang, J. Sun and H. Xu, Two-stage quadratic game under uncertainty and its solution by progressive hedging algorithm, *SIAM J. Optim.* 29(3) (2019) 1799–1818.

[41] J. Zou, S. Ahmed and X.A. Sun, Stochastic dual dynamic integer programming, *Math. Program.* 175 (2019) 461—502.

Manuscript received 25 November 2021
revised 2 January 2022
accepted for publication 6 January 2022

Yuan Lu

Department of Mathematics and Applied Mathematics
Shenyang University, Shenyang 110044, P.R. China
E-mail address: sylu8126@163.com

Jie Sun
School of EECMS, Curtin University, Bentley 6102, Australia and;
School of Business National University of Singapore
Singapore 119245
E-mail address: jie.sun@curtin.edu.au

Min Zhang
Xinjiang Institute of Ecology and Geography
CAS, Urumuqi 830011, P.R. China and;
University of Chinese Academy of Sciences
Beijing 100049, P.R. China
E-mail address: zhangmin1206@ms.xjb.ac.cn

Ying Zhang
School of Mathematics, Tianjin University
Tianjin 300072, P.R. China
E-mail address: yingzhang@tju.edu.cn