

https://dspace.org/

Intelligent Optimization Algorithms: A Stochastic Closed-Loop Supply Chain Network Problem Involving Oligopolistic Competition for Multiproducts and Their Product Flow Routings.

Zhou, Y.; Chan, C.K.; Wong, K.H.; Lee, Y.C.E.

2015-10-26
Hindawi Publishing Corporation
http://hdl.handle.net/10539/20667
Zhou, Y. et al. 2015. Mathematical Problems in Engineering. 918705.

Downloaded from a DSpace repository. https://dspace.org/

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 918705, 22 pages
http://dx.doi.org/10.1155/2015/918705

Research Article

Intelligent Optimization Algorithms: A Stochastic Closed-Loop Supply Chain Network Problem Involving Oligopolistic Competition for Multiproducts and Their Product Flow Routings

Yan Zhou,[1] Chi Kin Chan,[2] Kar Hung Wong,[3] and Y. C. E. Lee[2]

[1]*Department of Management Science and Engineering, Qingdao University, Qingdao 266071, China*
[2]*Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
[3]*School of Computational and Applied Mathematics, University of the Witwatersrand, Johannesburg 2050, South Africa*

Correspondence should be addressed to Yan Zhou; yanyanz22@hotmail.com

Received 29 March 2015; Revised 20 July 2015; Accepted 26 July 2015

Academic Editor: Giovanni Falsone

Copyright © 2015 Yan Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, the first oligopolistic competition model of the closed-loop supply chain network involving uncertain demand and return has been established. This model belongs to the context of oligopolistic firms that compete noncooperatively in a Cournot-Nash framework. In this paper, we modify the above model in two different directions. (i) For each returned product from demand market to firm in the reverse logistics, we calculate the percentage of its optimal product flows in each individual path connecting the demand market to the firm. This modification provides the optimal product flow routings for each product in the supply chain and increases the optimal profit of each firm at the Cournot-Nash equilibrium. (ii) Our model extends the method of finding the Cournot-Nash equilibrium involving smooth objective functions to problems involving nondifferentiable objective functions. This modification caters for more real-life applications as a lot of supply chain problems involve nonsmooth functions. Existence of the Cournot-Nash equilibrium is established without the assumption of differentiability of the given functions. Intelligent algorithms, such as the particle swarm optimization algorithm and the genetic algorithm, are applied to find the Cournot-Nash equilibrium for such nonsmooth problems. Numerical examples are solved to illustrate the efficiency of these algorithms.

1. Introduction

In the past decade, the perfect competition equilibrium models of the supply chain network (SCN) have been widely studied. For instance, a perfect competition deterministic equilibrium model and a stochastic equilibrium model involving a lot of decision-makers were first established by Nagurney et al. [1] and Dong et al. [2], respectively. On the other hand, a perfect competition equilibrium model of a reverse SCN was constructed by Nagurney and Toyasaki [3] for the optimal management of the electronic wastes. Moreover, perfect competition equilibrium models of a closed-loop supply chain (CLSC) network were established by Hammond and Beullens [4] and Yang et al. [5] for the optimal management of waste electrical/electronic equipment and raw material, respectively. Qiang et al. [6] were the first to investigate a stochastic equilibrium model of a CLSC network, which involved uncertainties in demands, but not uncertainties in returns. The equilibrium conditions of all the above papers were obtained by the theory of variational inequality and were solved by the modified projection method (Korpelevich [7]).

Nowadays, more firms are aware of the importance of integrating the supply chain as a whole, consisting of all the marketing activities of all the competitors. The integration of the entire supply chain generates oligopolistic competition among firms. As a consequence, research works were recently extended from the perfect competition markets model to the oligopolistic firms model in the forward SCN. The oligopolistic competition among firms in the forward SCN was considered in a lot of real-life situations. For instance, oligopolistic

competition equilibrium models of a forward SCN were established by Masoumi et al. [8] and Yu and Nagurney [9] for the optimal management of perishable products such as pharmacies and fresh foods and by Nagurney and Yu [10] for the minimization of emission-generations. The equilibrium conditions of these models were also obtained by the theory of variational inequality and were solved by the Euler method (Dupuis and Nagurney [11]).

However, for both perfect competition and oligopolistic competition, the CLSC integrating the forward and reverse supply chain is more important than the forward supply chain alone due to the government legislation (such as the paper recycling directive and WEEE within the European Union [12]). Moreover, the process used in the CLSC to recycle used products (such as papers, glass, building wastes, and electronic and electrical equipment) for minimizing resource wastage also leads to people's understanding of the green supply chain management (GSCM) (Sheu and Talley [13] and Seuring [14]). Research work on oligopolistic competition model of the CLSC network only began very recently. The first oligopolistic competition equilibrium model of the CLSC network was established recently by Zhou et al. [15].

The model developed by Zhou et al. [15] belongs to the context of oligopolistic firms that compete noncooperatively in a Cournot-Nash framework in a stochastic environment. Since the demands and returns are uncertain, two types of risk, namely, the overstocking and understocking of multiproducts in the forward supply chain, are considered. By using the quantities of each new product and the path flows of each product on the forward supply chain as the decision variables, every oligopolistic firm's expected profit can be maximized at the Cournot-Nash equilibrium. The equilibrium condition of this model was also obtained by the theory of variational inequality; the variational inequalities in this paper were solved by the logarithmic quadratic proximal prediction-correction method (He et al. [16]).

As mentioned in the abstract, we modify the above model in two different directions as follows:

(i) The model of Zhou et al. [15] can calculate the optimal product flow in each path from firms to demand markets in the forward logistics, but not in paths from demand markets to firms in the reverse logistics because the quantity of the returned products from each demand market is a random variable. In this paper, for each returned product j from demand market R_k to firm i in the reverse logistics, we can calculate the percentage of its optimal product flows in each individual path connecting the above demand market to the above firm. This modification provides the optimal product flow routings for each product in the entire supply chain and hence can increase the optimal profit of each of the firms at the Cournot-Nash equilibrium.

(ii) Our proposed model extends the method of finding the Cournot-Nash equilibrium for CLSC problems involving smooth objective functions to problems involving nondifferentiable objective functions. This modification caters for more real-life applications as a lot of supply chain problems involve nonsmooth objective functions.

In this paper, we establish the Cournot-Nash equilibrium without the assumption of differentiability on the given functions and use intelligent algorithms, such as the genetic algorithm and the particle swarm optimization (PSO) algorithm, for finding the Cournot-Nash equilibrium for nonsmooth CLSC problems.

Genetic algorithm (Holland [17]) is a common intelligent optimization algorithm, which can find the Nash (Nash [18, 19]) equilibrium effectively. For instance, genetic algorithm has been used for finding the Stackelberg- [20] Nash equilibrium of a nonlinear, nonconvex, nondifferentiable multilevel programming model (Liu [21]). Complicated SNC problems involving the design of the hierarchical spanning tree network (Kim et al. [22]) and that of a vendor managed inventory SNC network (Yu and Huang [23]) were also successfully solved by the genetic algorithm. In these papers, the efficiency of the genetic algorithm for solving complicated combinational problems, in terms of both speed and accuracy, has been demonstrated.

The PSO algorithm, originally proposed by Eberhart and Kennedy [24], is a member of the swarm intelligence methods (Kennedy and Eberhart [25]) for solving global optimization problems. Similar to a lot of intelligent optimization algorithms, the PSO algorithm does not require the gradient information of both the objective functions and the constraint functions, but only their values. Thus, it is easily implemented and computationally inexpensive and has been successfully applied to solve continuous optimization problems as well as discrete optimization problems (Goksal et al. [26]). Numerical results have shown that the PSO algorithm is more efficient than the genetic algorithm, especially for solving problems involving continuous solution space. For instance, Kadadevaramath et al. [27] and Govindan et al. [28] solved, respectively, a three-echelon SCN problem and an optimization problem involving both the economic and environmental benefits of a perishable food SCN by both the genetic algorithm and the PSO algorithm. Numerical results indicated that the PSO algorithm is more efficient than the genetic algorithm, in terms of the accuracy of the optimal solutions.

In Section 5 of this paper, two numerical examples are solved to compare the efficiencies of the PSO algorithm, the genetic algorithm, and an algorithm based on variational inequalities for finding the Cournot-Nash equilibrium. In one numerical example (Example 1), the results show that when all the given functions are differentiable, the efficiencies of the PSO algorithm, the genetic algorithm, and the algorithm based on variational inequality are almost the same, in terms of the accuracy of the computed equilibrium. However, the PSO algorithm is just as efficient as the algorithm based on variational inequality but more efficient than the genetic algorithm, in terms of the total computational time required to obtain the equilibrium. In another numerical example (Example 2), the results show that, for problems involving nonsmooth objective functions, the PSO algorithm and the genetic algorithm can still find the Cournot-Nash

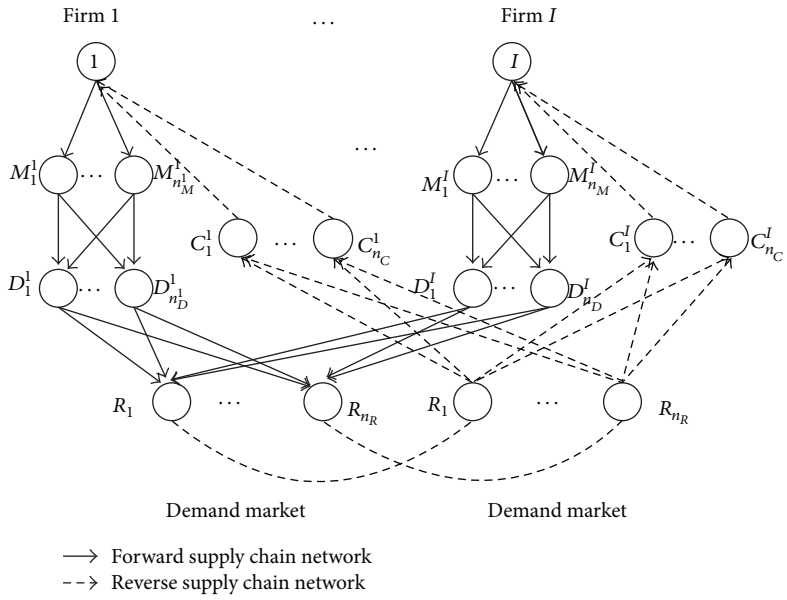


Figure 1: The CLSC network topology.

equilibrium efficiently, but the algorithm based on variational inequality is not efficient.

The rest of this paper is as follows. Section 2 develops the oligopolistic CLSC network model with multiproducts and uncertain demands and returns and constructs the Cournot-Nash equilibrium conditions for our model. Section 3 proves the existence of the Cournot-Nash equilibrium for our model. Section 4 presents a PSO algorithm and a genetic algorithm to find the Cournot-Nash equilibrium. In Section 5, numerical examples are solved to compare the effectiveness of the PSO algorithm, the genetic algorithm, and an algorithm based on variational inequality for finding the Cournot-Nash equilibrium. Section 6 presents our summary and conclusion.

2. An Equilibrium Model of a CLSC Network under Oligopolistic Competition among Firms Involving Product Flow Routings in Both Forward and Reverse Logistics

Zhou et al. [15] have established the first oligopolistic competition model of the closed-loop supply chain network (CLSC) involving uncertain demand and return. This model belongs to the context of oligopolistic firms that compete noncooperatively in a Cournot-Nash framework. In this model, they maximize every oligopolistic firm's expected profit by deciding the optimal production qualities of each new product as well as the amount of product flows in each individual path containing firms to demand markets in the forward logistics. Due to the fact that the quantities of returned products are random variables, they are unable to calculate the optimal amount of product flows in the reverse logistics. In this paper, we modify the above model by including the percentage of product flows in each path in the reverse logistics as a decision variable.

The topology of the network of our model is shown in Figure 1. Each firm i ($i = 1, \ldots, I$) produces J products. In order to satisfy the demand, the firms either manufacture new products or remanufacture used products through recycling used components obtained from the previous production period. Both the demands and returns are random variables having uniform distributions. As mentioned in the previous paragraph, each firm determines the optimal production quantities of the new products, the amount of product flows in each path in the forward logistics, and the percentage of product flows in each path in the reverse logistics to maximize its profit.

In the closed-loop supply chain, each firm is represented as a network of its economic activities. In the forward logistics, firm i ($i = 1, \ldots, I$) has n_M^i manufacturing faculties/plants and n_D^i distribution centers and serves the same n_R demand markets. In the reverse logistics, firm i has n_C^i recovery centers. The activities in the forward supply chain involve the production, remanufacturing, delivery, and distribution of products. The activities in the reverse supply chain involve the disposal, recycling, and returning of products.

In Figure 1, the links from the top-tiered i ($i = 1, \ldots, I$) of the forward logistics representing the individual firm are connected to the manufacturing nodes of the respective firm i, which are denoted by $M_1^i, \ldots, M_{n_M^i}^i$, respectively. The links from the manufacturing nodes are, in turn, connected to the distribution center nodes of firm i, which are denoted by $D_1^i, \ldots, D_{n_D^i}^i$, respectively. The links from the distribution centers nodes are, in turn, connected to the demand market nodes, denoted by $R_1, \ldots, R_{n_R}$, respectively; each of these demand markets is served by all the firms.

Also, in Figure 1, the links from every demand market node R_k ($k = 1, \ldots, n_R$) of the reverse logistics are connected

to all the recovery center nodes of firm i, which are denoted by $C_1^i, \ldots, C_{n_C^i}^i$, respectively. Finally, the links from the recovery centers nodes are connected to firm i to complete a single loop.

Thus, our CLSC network is one which incorporates the forward supply chain processes (such as production, remanufacturing, delivery, and distribution) with the reverse supply chain processes (such as disposal, recycling, and returning) to form a closed-loop network.

We wish to emphasize that the network topology in Figure 1 is for demonstration purpose only. In fact, the model can handle any prospective closed-loop chain network topology, provided that there are top-tiered nodes to represent each firm and bottom-tiered nodes to represent the demand markets with two sequences of directed links containing each top-tiered node to each bottom-tiered node in the forward and reverse directions, respectively.

In the forward supply chain, any set of correlated links that can connect a firm to a demand market via a manufacturer and a distribution center form a forward path p^f. Similarly, in the reverse supply chain, any set of correlated links that can connect a demand market to a firm via the recovery center form a reverse path p^r. From the discussion in the previous paragraph, the structure of our model can be completed specified by its forward and reverse paths.

We now give a few assumptions together with their justifications. These are common assumptions in the CLSC literature, except possibly for (A8). (See Hammond and Beullens [4] and Zhou et al. [15] for details.)

Assumption A. (A1) The demand for product j ($j = 1, \ldots, J$) of firm i ($i = 1, \ldots, I$) at demand market R_k ($k = 1, \ldots, n_R$) in one production period, denoted by $\widehat{d}_{ijk}$, is a random variable. Any two demands $\widehat{d}_{i_1 j_1 k_1}$ and $\widehat{d}_{i_2 j_2 k_2}$ with $(i_1, j_1, k_1) \neq (i_2, j_2, k_2)$ are independent of each other. This assumption is due to the fact that all the buyers are independent.

(A2) A fraction of the used products which is in good condition will be returned to the firms for remanufacturing; the remainder of used products which is not in good condition will be sent to the landfill for disposal. Thus, we assume that, for any used product j ($j = 1, \ldots, J$) which is in good condition, a cost of ρ_j^{Re} per item will be charged to the firms for the purchase of these used products. For any used product j ($j = 1, \ldots, J$) which is not in good condition, a cost of $\overline{\rho}$ (independent of j) per item will be charged to the firms for the disposal of these used products at the landfall.

(A3) All the returned products are remanufacturable. This assumption is a direct consequence of (A2).

(A4) The return of used product j ($j = 1, \ldots, J$) from demand market R_k ($k = 1, \ldots, n_R$) to firm i ($i = 1, \ldots, I$) in one production period, denoted by $\widehat{r}_{kji}$, is a random variable. Any two returns $\widehat{r}_{k_1 j_1 i_1}$ and $\widehat{r}_{k_2 j_2 i_2}$ with $(k_1, j_1, i_1) \neq (k_2, j_2, i_2)$ are independent of each other. This assumption is due to the fact that all the customers are independent.

(A5) Due to the existence of modern manufacturing technologies, manufacturers can easily transform the remanufactured products into as-new products. Thus, we assume that there is no distinction between the qualities of the brand new products and the remanufactured products and they are sold to the markets at the same price.

(A6) The operational cost of product j on any link a in the forward logistics, denoted by $\widehat{c}_{ja}$, is a continuous function of all the link flows of this product in the forward supply chain. In other words, $\widehat{c}_{ja}$ depends continuously not only on the product flow in link a, but also on the product flows in all the manufacturer links and all the shipment links and all the distribution links. Consequently, the operation cost of product j on any path p^f in the forward supply chain, denoted by $\widehat{c}_{jp^f}$, is also a continuous function of all the path flows of its product in the forward supply chain. This assumption is due to the fact that competition for business exists among all the companies involving economic activities in the forward logistics.

(A7) The operational cost of product j on any path p^r in the reverse logistics, denoted by $\widehat{c}_{jp^r}$, is a continuous function of the path flow of this product on p^r only.

(A8) For product j ($j = 1, \ldots, J$) of firm i ($i = 1, \ldots, I$), the cost of manufacturing z items of new product j by firm i is $\overline{f}_{ij}(z)$, and the cost of remanufacturing z items of returned product j by firm i is $\overline{f}_{ij}^{\mathrm{Re}}(z)$, where $\overline{f}_{ij}$ and $\overline{f}_{ij}^{\mathrm{Re}}$ are given continuous functions. Due to the operational advantages, the manufacturing costs are higher than the remanufacturing costs; that is,

$$\overline{f}_{ij}(z) > \overline{f}_{ij}^{\mathrm{Re}}(z), \tag{1}$$

for $i = 1, \ldots, I$ and $j = 1, \ldots, J$.

(Note that, due to the logistics advantages, the optimal strategies of any firm i ($i = 1, \ldots, I$) at the Cournot-Nash equilibrium must automatically satisfy the following relationship: total reverse logistics costs of firm i (i.e., total purchase costs of returned products + total transportation costs of returned products) + total remanufacturing costs of firm i < total manufacturing costs of firm i.)

(A9) Due to the limitation of the facilities at the manufacturers' factories, the quantity of new product j ($j = 1, \ldots, J$) manufactured by firm i ($i = 1, \ldots, I$) in one production period cannot exceed $\overline{x}$, where $\overline{x}$ is a positive integer.

In order to formulate the problem of finding the Cournot-Nash equilibrium for our CLSC network, we first formulate three types of decision variables, namely, the quantity of new manufactured product, the amount of product flows in each path in the forward logistics, and the percentage of product flows in each path in the reverse logistics.

For any firm i ($i = 1, \ldots, I$), there are n_M^i manufacturers and n_D^i distribution centers served for firm i in the forward logistics. Thus, there are altogether $n_M^i \times n_D^i$ different paths which can connect firm i to demand market R_k ($k = 1, \ldots, n_R$) via the manufacturers and distribution centers. Let P_k^i denote the set of all the forward paths associated with firm i and demand market R_k and let P denote the set of all the forward paths associated with all the firms and all the demand markets. For each $p^f \in P_k^i$, let x_{jp^f} be the amount of product flows of product j in the forward path p^f.

For any firm i ($i = 1,\ldots,I$), there are n_C^i recovery centers served for firm i in the reverse logistics. Thus, there are altogether n_C^i different paths which can connect demand market R_k ($k = 1,\ldots,n_R$) to firm i via the recovery centers. Let $\widehat{P}_k^i$ denote the set of all the reverse paths associated with firm i and demand market R_k. For each returned product j from demand market R_k to firm i in the reverse logistics, let y_{jp^r} be the percentage of its optimal product flows in the path p^r connecting the above demand market to the above firm.

For any firm i ($i = 1,\ldots,I$) and product j ($j = 1,\ldots,J$), let x_{ij}^{New} be the quantity of new product j manufactured by firm i in one production period.

Thus, there are altogether $(n_M^i \times n_D^i) \times n_R + n_C^i \times n_R + 1$ decision variables associated with firm i and product j for any $i = 1,\ldots,I$, $j = 1,\ldots,J$, where the $(n_M^i \times n_D^i) \times n_R$ decision variables are of the form x_{jp^f}, which represent the amount of the flows of product j on all the forward paths $p^f \in P_k^i$ ($k = 1,\ldots,n_R$), the $n_C^i \times n_R$ decision variables are of the form y_{jp^r}, which represent the percentage of the flows of product j in all the reverse paths $p^r \in \widehat{P}_k^i$ ($k = 1,\ldots,n_R$), and the other decision variable is of the form x_{ij}^{New}, which represents the quantity of new product j manufactured by firm i.

Let

$$X_{ij} \equiv \left\{\left(x_{jp^f}, y_{jp^r}, x_{ij}^{\text{New}}\right) \mid p^f \in P_k^i,\ p^r \in \widehat{P}_k^i,\ k = 1,\ldots,n_R\right\} \in \mathbb{R}_+^{(n_M^i \times n_D^i + n_C^i) \times n_R + 1} \tag{2}$$

be the vector representing all the decision variables associated with firm i and product j.

Let

$$X_i \equiv \left(X_{i1},\ldots,X_{iJ}\right)^T \in \mathbb{R}_+^{(n_M^i \times n_D^i + n_C^i) \times n_R J + J} \tag{3}$$

be the strategy vector representing the overall decision variables associated with firm i.

Then

$$X \equiv \left(X_1,\ldots,X_i,\ldots,X_I\right)^T \in K \tag{4}$$

is the overall decision variables for the entire CLSC network, where $K \equiv \mathbb{R}_+^{\sum_{i=1}^{I}(n_M^i \times n_D^i + n_C^i) \times n_R J + IJ}$.

We now formulate all the constraints in the reverse logistics. From the definition of y_{jp^r}, we have

$$0 \le y_{jp^r} \le 1,\quad j = 1,\ldots,J,\ p^r \in \widehat{P}_k^i\ \left(i = 1,\ldots,I,\ k = 1,\ldots,n_R\right), \tag{5}$$

$$\sum_{p^r \in \widehat{P}_k^i} y_{jp^r} = 1,\quad i = 1,\ldots,I,\ j = 1,\ldots,J,\ k = 1,\ldots,n_R. \tag{6}$$

Thus (5) provides upper and lower bounds for the product flow y_{jp^r}, $j = 1,\ldots,J$, $p^r \in \widehat{P}_k^i$ ($i = 1,\ldots,I$, $k = 1,\ldots,n_R$) in the reverse logistics.

Since the total amount of returned product j from demand market R_k to firm i is $\widehat{r}_{kji}$, it is clear that the amount of product flow in the reverse path p^r ($p^r \in \widehat{P}_k^i$) is equal to x_{jp^r}, where

$$x_{jp^r} = y_{jp^r}\widehat{r}_{kji},\quad j = 1,\ldots,J,\ \forall p^r \in \widehat{P}_k^i,\ \left(i = 1,\ldots,I,\ k = 1,\ldots,n_R\right). \tag{7}$$

We now formulate all the constraints in the forward logistics.

In view of (A9), we have

$$x_{ij}^{\text{New}} \le \overline{x}. \tag{8}$$

Constraint (8) provides an upper bound for the quantity of new product j ($j = 1,\ldots,J$) manufactured by firm i ($i = 1,\ldots,I$).

Let x_{ij}^{Re} be the random variable representing the quantity of product j remanufactured by firm i in one production period. Then in view of (A3) and (A4), we have

$$x_{ij}^{\text{Re}} = \sum_{k=1}^{n_R} \widehat{r}_{kji}, \tag{9}$$

In view of (9), the following flow inequality, which provides an upper bound for the total amount of flows of product j ($j = 1,\ldots,J$) from firm i ($i = 1,\ldots,I$) to all the demand markets in the forward supply chain, must hold:

$$\sum_{k=1}^{n_R} \sum_{p^f \in P_k^i} x_{jp^f} \le x_{ij}^{\text{New}} + E\left(\sum_{k=1}^{n_R} \widehat{r}_{kji}\right). \tag{10}$$

Furthermore, the following flow inequality, which provides a lower bound for the amount of flows of product j ($j = 1,\ldots,J$) from firm i ($i = 1,\ldots,I$) to the demand market k ($k = 1,\ldots,n_R$) in the forward supply chain, must also hold:

$$E\left(\widehat{r}_{kji}\right) \le \sum_{p^f \in P_k^i} x_{jp^f}. \tag{11}$$

We now formulate different cost functions for our CLSC network. We first formulate the expected penalty cost (due to excessive or insufficient supply).

Consider the forward supply chain involving firm i ($i = 1,\ldots,I$), product j ($j = 1,\ldots,J$), and demand market R_k ($k = 1,\ldots,n_R$). Let v_{ijk} denote the quantity of product j supplied by firm i to demand market R_k in one production period. Then the above quantity is equal to the total amount of flows of product j on paths connecting firm i to demand market R_k. Thus,

$$v_{ijk} = \sum_{p^f \in P_k^i} x_{jp^f}. \tag{12}$$

Let $\widehat{d}_{ijk}$ be the total demand associated with firm i, product j, and demand market R_k in one production period. Let $\mathscr{F}_{ijk}$ be the probability density function of $\widehat{d}_{ijk}$. Then

$$\Pr\left(\widehat{d}_{ijk} \le x\right) = \int_0^x \mathscr{F}_{ijk}\left(\widehat{d}_{ijk}\right) \mathrm{d}\widehat{d}_{ijk}, \tag{13}$$

where Pr denote the probability. For each product j, let

$$\widehat{d}_j = \left(\widehat{d}_{1j1}, \ldots, \widehat{d}_{1jn_R}, \ldots, \widehat{d}_{Ij1}, \ldots, \widehat{d}_{Ijn_R}\right)^T \in R^{I \times n_R} \quad (14)$$

denote the vector consisting of all the demands of product j in the entire CLSC network.

Now, the quantity of product j supplied by firm i to demand market R_k cannot exceed the minimum of v_{ijk} and $\widehat{d}_{ijk}$. In other words, the actual sale of these products is equal to $\min\{v_{ijk}, \widehat{d}_{ijk}\}$. Let

$$\begin{aligned} \Delta^+_{ijk} &= \max\left\{0, v_{ijk} - \widehat{d}_{ijk}\right\}, \\ \Delta^-_{ijk} &= \max\left\{0, \widehat{d}_{ijk} - v_{ijk}\right\}, \end{aligned} \quad (15)$$

denote, respectively, the quantity of the overstocking and the understocking of product j associated with firm i and demand market R_k. The expected values of Δ^+_{ijk} and Δ^-_{ijk} are given by

$$E\left(\Delta^+_{ijk}\right) = \int_0^{v_{ijk}} \left(v_{ijk} - \widehat{d}_{ijk}\right) \mathcal{F}_{ijk}\left(\widehat{d}_{ijk}\right) \mathrm{d}\widehat{d}_{ijk}, \quad (16)$$

$$E\left(\Delta^-_{ijk}\right) = \int_{v_{ijk}}^{+\infty} \left(\widehat{d}_{ijk} - v_{ijk}\right) \mathcal{F}_{ijk}\left(\widehat{d}_{ijk}\right) \mathrm{d}\widehat{d}_{ijk}. \quad (17)$$

Assume that the unit penalty incurred on firm i due to excessive supply of product j to demand market R_k is θ^+_{ijk} and the unit penalty incurred on firm i due to insufficient supply of product j to demand market R_k is θ^-_{ijk}, where $\theta^+_{ijk} \geq 0$ and $\theta^-_{ijk} \geq 0$. Then, the total expected penalty incurred on firm i associated with product j and demand market R_k is given by

$$E\left(\theta^+_{ijk}\Delta^+_{ijk} + \theta^-_{ijk}\Delta^-_{ijk}\right) = \theta^+_{ijk}E\left(\Delta^+_{ijk}\right) + \theta^-_{ijk}E\left(\Delta^-_{ijk}\right). \quad (18)$$

Apart from the penalty costs given by (18) (due to excessive or insufficient supply), the following additional costs are also charged for firm i ($i = 1, \ldots, I$):

(i) From (A6), the total operational cost of product j ($j = 1, \ldots, J$) on the forward path is

$$\sum_{k=1}^{n_R} \sum_{p^f \in P^i_k} \widehat{c}_{jp^f}\left(x_{jP}\right), \quad (19)$$

where x_{jP} is the vector representing the amount of product flows of product j on all the forward paths.

(ii) From (A7) and (7), the total operational cost of product j ($j = 1, \ldots, J$) on all the reverse path is

$$\sum_{k=1}^{n_R} \sum_{p^f \in \widetilde{P}^i_k} \widehat{c}_{jp^r}\left(y_{jp^r}\widehat{r}_{kji}\right). \quad (20)$$

(iii) From (A2), the total cost related to the purchase of returned product j is

$$\rho^{\mathrm{Re}}_j \sum_{k=1}^{n_R} \widehat{r}_{kji}. \quad (21)$$

(iv) Also from (A2), the total cost related to the disposal of uncollected product j ($j = 1, \ldots, J$) at the landfill site is

$$\overline{\rho} \sum_{k=1}^{n_R} \left(\sum_{p^f \in P^i_k} x_{jp^f} - \widehat{r}_{kji} \right). \quad (22)$$

(v) From (A8) and (9), the cost of manufacturing new product j ($j = 1, \ldots, J$) is

$$\overline{f}_{ij}\left(x^{\mathrm{New}}_{ij}\right), \quad (23)$$

and the cost of remanufacturing the returned product j ($j = 1, \ldots, J$) is

$$\overline{f}^{\mathrm{Re}}_{ij}\left(\sum_{k=1}^{n_R} \widehat{r}_{kji}\right). \quad (24)$$

By virtue of (18) and (19)–(24), the total expected cost incurred on firm i ($i = 1, \ldots, I$) is given by

$$\begin{aligned} &\sum_{j=1}^{J}\sum_{k=1}^{n_R} E\left(\theta^+_{ijk}\Delta^+_{ijk} + \theta^-_{ijk}\Delta^-_{ijk}\right) + \sum_{j=1}^{J} \overline{f}_{ij}\left(x^{\mathrm{New}}_{ij}\right) \\ &\quad + \sum_{j=1}^{J}\sum_{k=1}^{n_R} \sum_{p^f \in P^i_k} \widehat{c}_{jp^f}\left(x_{jP}\right) \\ &\quad + \sum_{j=1}^{J}\sum_{k=1}^{n_R} \sum_{p^f \in \widetilde{P}^i_k} E\left[\widehat{c}_{jp^r}\left(y_{jp^r}\widehat{r}_{kji}\right)\right] + \sum_{j=1}^{J}\sum_{k=1}^{n_R} \rho^{\mathrm{Re}}_j E\left(\widehat{r}_{kji}\right) \\ &\quad + \sum_{j=1}^{J} E\left(\overline{f}^{\mathrm{Re}}_{ij}\left(\sum_{k=1}^{n_R} \widehat{r}_{kji}\right)\right) \\ &\quad + \overline{\rho} \sum_{j=1}^{J}\sum_{k=1}^{n_R} E\left(\sum_{p^f \in P^i_k} x_{jp^f} - \widehat{r}_{kji} \right). \end{aligned} \quad (25)$$

Now, we formulate the total revenue received by firm i ($i = 1, \ldots, I$). In order to capture competition for demand in the entire CLSC network, we assumed that, for each product j, the demand price function ρ_{ijk} ($i = 1, \ldots, I, k = 1, \ldots, n_R$) is a continuous function of all the demands associated with product j in the entire CLSC network; that is,

$$\rho_{ijk} = \rho_{ijk}\left(\widehat{d}_j\right). \quad (26)$$

In other words, the price function ρ_{ijk} depends continuously not only on $\widehat{d}_{ijk}$, but also on all the demand associated with product j. Masoumi et al. [8] and Nagurney and Yu [10] used such demand price function in the study of forward supply chain network involving oligopolistic competition among firms. Then the expected revenue received by firm i ($i = 1, \ldots, I$) is given by

$$\sum_{j=1}^{J}\sum_{k=1}^{n_R} E\left(\rho_{ijk}\left(\widehat{d}_j\right) \min\left\{v_{ijk}, \widehat{d}_{ijk}\right\}\right). \quad (27)$$

By virtue of the revenue, the cost of the forward supply chain, and the cost of the reverse supply chain, the expected profit function of firm i ($i = 1, \ldots, I$), denoted by U_i, can be expressed as follows:

$$\begin{aligned} U_i = & \sum_{j=1}^{J} \sum_{k=1}^{n_R} E\left(\rho_{ijk}\left(\widehat{d}_j\right) \min\left\{v_{ijk}, \widehat{d}_{ijk}\right\}\right) \\ & - \sum_{j=1}^{J} \sum_{k=1}^{n_R} E\left(\theta_{ijk}^{+}\Delta_{ijk}^{+} + \theta_{ijk}^{-}\Delta_{ijk}^{-}\right) - \sum_{j=1}^{J} \overline{f}_{ij}\left(x_{ij}^{\mathrm{New}}\right) \\ & - \sum_{j=1}^{J} \sum_{k=1}^{n_R} \sum_{p^f \in P_k^i} \widehat{c}_{jp^f}\left(x_{jP}\right) \\ & - \sum_{j=1}^{J} \sum_{k=1}^{n_R} \sum_{p^f \in \widehat{P}_k^i} E\left[\widehat{c}_{jp^r}\left(y_{jp^r}\widehat{r}_{kji}\right)\right] \\ & - \sum_{j=1}^{J} \sum_{k=1}^{n_R} \rho_j^{\mathrm{Re}} E\left(\widehat{r}_{kji}\right) - \sum_{j=1}^{J} E\left(\overline{f}_{ij}^{\mathrm{Re}}\left(\sum_{k=1}^{n_R} \widehat{r}_{kji}\right)\right) \\ & - \overline{\rho} \sum_{j=1}^{J} \sum_{k=1}^{n_R} E\left(\sum_{p^f \in P_k^i} x_{jp^f} - \widehat{r}_{kji}\right), \end{aligned} \tag{28}$$

where (12) holds for all decision variables defined by (4). By virtue of (16), (17), (19)–(24), and (26), we know that U_i is a function of the strategy vector of all firms in the entire CLSC network. That is,

$$U_i = U_i(X). \tag{29}$$

Now, in order to define the Cournot-Nash equilibrium of the CLSC network, we need to consider the following constraints involving firm i and product j ($i = 1, \ldots, I$, $j = 1, \ldots, J$):

$$x_{ij}^{\mathrm{New}} \le \overline{x}, \tag{30}$$

$$\sum_{k=1}^{n_R} \sum_{p^f \in P_k^i} x_{jp^f} \le x_{ij}^{\mathrm{New}} + E\left(\sum_{k=1}^{n_R} \widehat{r}_{kji}\right), \tag{31}$$

$$E\left(\widehat{r}_{kji}\right) \le \sum_{p^f \in P_k^i} x_{jp^f}, \quad k = 1, \ldots, n_R, \tag{32}$$

$$\sum_{p^r \in \widehat{P}_k^i} y_{jp^r} = 1, \quad k = 1, \ldots, n_R, \tag{33}$$

$$0 \le y_{jp^r} \le 1, \ p^r \in \widehat{P}_k^i, \ (k = 1, \ldots, n_R), \tag{34}$$

$$\begin{aligned} & x_{jp^f} \ge 0, \ p^f \in P_k^i, \ (k = 1, \ldots, n_R), \\ & x_{ij}^{\mathrm{New}} \ge 0, \end{aligned} \tag{35}$$

where constraints (30), (31), (32), (33), and (34) are directly obtained from (8), (1), (11), (6), and (5), respectively; constraints (34) and (35) ensure that all the decision variables are nonnegative.

In this CLSC network, there are I oligopolistic firms competing noncooperatively to maximize their own expected profits and selecting their strategy vectors till an equilibrium is established. So the game is based on oligopolistic Cournot pricing in a Cournot-Nash framework.

Now, we can define the Cournot-Nash equilibrium of the CLSC network according to Definitions 1, 2, and 3 given below.

Definition 1 (a feasible strategy vector). For each i ($i = 1, \ldots, I$), a strategy vector $X_i \in \mathbb{R}_{+}^{(n_M^i \times n_D^i + n_C^i) n_R J + J}$ is said to be a feasible strategy vector, if X_i satisfies constraints (30)–(35).

Definition 2 (the set of all feasible strategy pattern). Let $\mathcal{X}$ be the set of all feasible strategy pattern defined by

$$\mathcal{X} \equiv \{(X_1, X_2, \ldots, X_I)\}, \tag{36}$$

where X_i ($i = 1, \ldots, I$) is a feasible strategy vector defined by Definition 1.

Definition 3 (the Cournot-Nash equilibrium of the CLSC network). A feasible strategy pattern $X^* \in \mathcal{X}$ constitutes a Cournot-Nash equilibrium of the CLSC network, if the following inequality holds for all i and for all feasible strategy vectors X_i:

$$U_i\left(X_i^*, \widehat{X}_i^*\right) \ge U_i\left(X_i, \widehat{X}_i^*\right), \tag{37}$$

where $\widehat{X}_i^* \equiv (X_1^*, \ldots, X_{i-1}^*, X_{i+1}^*, \ldots, X_I^*)$.

Thus, an equilibrium is established if no firm in the CLSC network can unilaterally increase its expected profit (without violating feasibility) by changing any of its strategy, given that the strategies of the other firms do not change.

3. Existence Results

In this section, we establish the existence of the Cournot-Nash equilibrium of the CLSC network defined by Definition 3.

To guarantee the existence of the Cournot-Nash equilibrium of the CLSC network, the following additional assumption is held for $U_i(X)$ ($i = 1, \ldots, I$).

Assumption B. (B1) The operational cost of product j in any path p^f in the forward logistics, denoted by $\widehat{c}_{jp^f}$, is both a continuous and a convex function of x_{jP}, the vector consisting of all forward path flows of product j.

(B2) The operational cost of product j in any path p^r in the reverse logistics, denoted by $\widehat{c}_{jp^r}$, is both a continuous and a convex function of x_{jp^r}, the product flow of this product in the reverse path p^r.

(B3) The production cost $\overline{f}_{ij}$ is a continuous and convex function of x_{ij}^{New}, the new product j manufactured by firm i.

Remark 4. Note that Assumption (B1) and Assumption (B3) in this paper are similar to Assumption (B1) and Assumption (B2) by Zhou et al. [15], except that there is no assumption of differentiability of the given functions in this paper.

In the following discussion, Lemma 5 provides us with sufficient conditions under which the set of all feasible strategy pattern $\mathcal{X}$ has a Cournot-Nash equilibrium, whereas in Lemmas 6–8 we prove that, under Assumptions A and B, the set of all feasible strategy pattern $\mathcal{X}$ indeed satisfies the sufficient condition for the existence of the Cournot-Nash equilibrium.

Lemma 5. *Suppose the set of all feasible strategy pattern $\mathcal{X}$ defined by Definition 2 is both a compact and convex set of K. Suppose the expected profit function $U_i(\cdot)$ $(i = 1, \ldots, I)$ defined by (28) is continuous and concave with respect to the decision variable X defined by (4). Then there exists a Cournot-Nash equilibrium of the CLSC network defined by Definition 3.*

Proof. By virtue of the fact that all the conditions imposed in this lemma are almost the same as those imposed in Theorem 1, page 639 of Nishimura and Friedman's paper [29], except that the concavity condition in this lemma is more restrictive than that imposed in Theorem 1 of Nishimura and Friedman's paper [29] (see condition (A5) on page 639 of Nishimura and Friedman's paper [29]), the proof of this lemma follows easily from that of Theorem 1 of Nishimura and Friedman's paper [29]. □

Lemma 6. *Suppose that Assumption A is satisfied; then the set of all feasible strategy pattern $\mathcal{X}$ defined by Definition 2 is both a compact and a convex set of K.*

Proof. The fact that $\mathcal{X}$ is bounded follows easily from (35), (31), (30), and (34). The fact that $\mathcal{X}$ is closed and convex follows easily from (30), (31), (32), (33), (34), and (35). Hence, $\mathcal{X}$ is both a compact and a convex set of K. □

Lemma 7. *Suppose that Assumptions A and B are satisfied; then the expected profit function $U_i(\cdot)$ $(i = 1, \ldots, I)$ defined by (28) is continuous with respect to the decision variable X being defined by (4).*

Proof. Firstly, from (28), (12), (16)–(19), and (B1), we know that the first, second, and fourth terms of $U_i(\cdot)$ are continuous with respect to x_{jp^f}, for each j $(j = 1, \ldots, J)$ and $p^f \in P_k^i$ $(k = 1, \ldots, n_R)$. From (B3), we know that the third term of $U_i(\cdot)$ is a continuous function of x_{ij}^{New}, for each j $(j = 1, \ldots, J)$. The sixth, seventh, and eighth terms of $U_i(\cdot)$ $(i = 1, \ldots, I)$ are constants, because $\widehat{r}_{kji}$ and x_{ij}^{Re} are random variables, for each k $(k = 1, \ldots, n_R)$ and j $(j = 1, \ldots, J)$. By virtue of the fact that $\widehat{r}_{kji}$ is a random variable, we know that the last term of $U_i(\cdot)$ is also a continuous function of x_{jp^f}, for each j $(j = 1, \ldots, J)$ and $p^f \in P_k^i$, $k = 1, \ldots, n_R$. From (B2), we know that the fifth term of $U_i(\cdot)$ is a continuous function of y_{jp^r}, for each j $(j = 1, \ldots, J)$ and $p^r \in \widehat{P}_k^i$ $(k = 1, \ldots, n_R)$. Thus, every term of $U_i(\cdot)$ is continuous with respect to the decision variable X defined by (4). The proof is complete. □

Lemma 8. *Suppose that Assumptions A and B are satisfied; then the expected profit function $U_i(\cdot)$ $(i = 1, \ldots, I)$ defined by (28) is concave with respect to the decision variable X defined by (4).*

Proof. In order to prove that the first and second terms of $U_i(\cdot)$ $(i = 1, \ldots, I)$ are concave with respect to the decision variable X defined by (4), we calculate the second partial derivative of these terms with respect to the decision variable x_{jp^f} as follows.

By virtue of the fact that any two demands are independent of each other (from (A1)), the first term of $U_i(\cdot)$ $(i = 1, \ldots, I)$ can be expressed as

$$\begin{aligned} &E\left(\rho_{ijk}\left(\widehat{d}_j\right)\min\left\{v_{ijk}, \widehat{d}_{ijk}\right\}\right) \\ &\quad = \int_0^{v_{ijk}} \widehat{d}_{ijk}\overline{G}\left(\widehat{d}_{ijk}\right)\mathcal{F}_{ijk}\left(\widehat{d}_{ijk}\right)\mathrm{d}\widehat{d}_{ijk} \\ &\qquad + \int_{v_{ijk}}^{+\infty} v_{ijk}\overline{G}\left(\widehat{d}_{ijk}\right)\mathcal{F}_{ijk}\left(\widehat{d}_{ijk}\right)\mathrm{d}\widehat{d}_{ijk}, \end{aligned} \tag{38}$$

where

$$\begin{aligned} \overline{G}\left(\widehat{d}_{ijk}\right) &= \int_0^{+\infty}\cdots\int_0^{+\infty}\rho_{ijk}\left(\widehat{d}_{111}, \ldots, \widehat{d}_{ij(k-1)}, \widehat{d}_{ijk}, \right. \\ &\quad \left.\widehat{d}_{ij(k+1)}, \ldots, \widehat{d}_{Ijn_R}\right)\mathcal{F}_{111}\left(\widehat{d}_{111}\right) \\ &\quad \cdots\mathcal{F}_{ij(k-1)}\left(\widehat{d}_{ij(k-1)}\right)\mathcal{F}_{ij(k+1)}\left(\widehat{d}_{ij(k+1)}\right) \\ &\quad \cdots\mathcal{F}_{Ijn_R}\left(\widehat{d}_{Ijn_R}\right)\mathrm{d}\widehat{d}_{111}\cdots\mathrm{d}\widehat{d}_{ij(k-1)}\mathrm{d}\widehat{d}_{ij(k+1)}\cdots\mathrm{d}\widehat{d}_{Ijn_R}. \end{aligned} \tag{39}$$

From (38), we have

$$\begin{aligned} &\frac{\partial E\left(\rho_{ijk}\left(\widehat{d}_j\right)\min\left\{v_{ijk}, \widehat{d}_{ijk}\right\}\right)}{\partial v_{ijk}} \\ &\quad = \int_{v_{ijk}}^{+\infty}\overline{G}\left(\widehat{d}_{ijk}\right)\mathcal{F}_{ijk}\left(\widehat{d}_{ijk}\right)\mathrm{d}\widehat{d}_{ijk}, \end{aligned} \tag{40}$$

$$\begin{aligned} &\frac{\partial^2 E\left(\rho_{ijk}\left(\widehat{d}_j\right)\min\left\{v_{ijk}, \widehat{d}_{ijk}\right\}\right)}{\partial v_{ijk}^2} \\ &\quad = -\overline{G}\left(v_{ijk}\right)\mathcal{F}_{ijk}\left(v_{ijk}\right). \end{aligned} \tag{41}$$

But, from (12), we know that

$$\frac{\partial v_{ijk}}{\partial x_{jp^f}} = 1, \tag{42}$$

for each j $(j = 1, \ldots, J)$ and $p^f \in P_k^i$ $(k = 1, \ldots, n_R)$.

Thus, from (41) and (42), the second partial derivative of the first term of $U_i(\cdot)$ becomes

$$\begin{aligned} &\frac{\partial^2 E\left(\rho_{ijk}\left(\widehat{d}_j\right)\min\left\{v_{ijk}, \widehat{d}_{ijk}\right\}\right)}{\partial x_{jp^f}^2} \\ &\quad = -\overline{G}\left(v_{ijk}\right)\mathcal{F}_{ijk}\left(v_{ijk}\right) < 0, \end{aligned} \tag{43}$$

for each j $(j = 1, \ldots, J)$ and $p^f \in P_k^i$ $(k = 1, \ldots, n_R)$.

From (16), we have

$$\frac{\partial E\left(\Delta_{ijk}^{+}\right)}{\partial v_{ijk}} = \frac{\partial}{\partial v_{ijk}} \int_{0}^{v_{ijk}} \left(v_{ijk} - \widehat{d}_{ijk}\right) \mathscr{F}_{ijk}\left(\widehat{d}_{ijk}\right) \mathrm{d}\widehat{d}_{ijk} = \int_{0}^{v_{ijk}} \mathscr{F}_{ijk}\left(\widehat{d}_{ijk}\right) \mathrm{d}\widehat{d}_{ijk}, \tag{44}$$

$$\frac{\partial^2 E\left(\Delta_{ijk}^{+}\right)}{\partial v_{ijk}^2} = \mathscr{F}_{ijk}\left(v_{ijk}\right). \tag{45}$$

From (17), we have

$$\frac{\partial E\left(\Delta_{ijk}^{-}\right)}{\partial v_{ijk}} = \frac{\partial}{\partial v_{ijk}} \int_{v_{ijk}}^{+\infty} \left(\widehat{d}_{ijk} - v_{ijk}\right) \mathscr{F}_{ijk}\left(\widehat{d}_{ijk}\right) \mathrm{d}\widehat{d}_{ijk} = -\int_{v_{ijk}}^{+\infty} \mathscr{F}_{ijk}\left(\widehat{d}_{ijk}\right) \mathrm{d}\widehat{d}_{ijk}, \tag{46}$$

$$\frac{\partial^2 E\left(\Delta_{ijk}^{-}\right)}{\partial v_{ijk}^2} = \mathscr{F}_{ijk}\left(v_{ijk}\right). \tag{47}$$

Thus, from (18), (42), (45), and (47), the second partial derivative of the second term of $U_i(\cdot)$ becomes

$$\frac{\partial^2 E\left(\theta_{ijk}^{+}\Delta_{ijk}^{+} + \theta_{ijk}^{-}\Delta_{ijk}^{-}\right)}{\partial x_{jp^f}^2} = \left(\theta_{ijk}^{+} + \theta_{ijk}^{-}\right) \mathscr{F}_{ijk}\left(v_{ijk}\right) > 0, \tag{48}$$

for each j ($j = 1, \ldots, J$) and $p^f \in P_k^i$, $k = 1, \ldots, n_R$.

By virtue of (B1), we know that $(-\widehat{c}_{jp^f}(x_{jP}))$ is concave with respect to x_{jp^f} for each j ($j = 1, \ldots, J$) and $p^f \in P_k^i$ ($k = 1, \ldots, n_R$).

Now, by virtue of (43), (48), the fact that the third, fifth, sixth, and seventh terms of (28) are independent of x_{jp^f}, the fact that the fourth term of (28) is concave with respect to x_{jp^f}, and the fact that the last term of (28) is a linear function of x_{jp^f}, we conclude that $U_i(\cdot)$ ($i = 1, \ldots, I$) is concave with respect to x_{jp^f} for each j ($j = 1, \ldots, J$) and $p^f \in P_k^i$ ($k = 1, \ldots, n_R$).

From (B3), we know that $(-\overline{f}_{ij}(x_{ij}^{\mathrm{New}}))$ is concave with respect to x_{ij}^{New}. Since $(-\overline{f}_{ij}(x_{ij}^{\mathrm{New}}))$ is the only term in (28) involving x_{ij}^{New}, we conclude that $U_i(\cdot)$ ($i = 1, \ldots, I$) is concave with respect to x_{ij}^{New} for each j ($j = 1, \ldots, J$).

By virtue of (B2), we know that $-E[\widehat{c}_{jp^r}(y_{jp^r}\widehat{r}_{ijk})]$ is concave with respect to y_{jp^r} for each j ($j = 1, \ldots, J$) and $p^r \in \widehat{P}_k^i$ ($k = 1, \ldots, n_R$).

Hence, we conclude that the expected profit function $U_i(\cdot)$ ($i = 1, \ldots, I$) defined by (28) is concave with respect to the decision variable X defined by (4). □

Theorem 9. *Suppose that Assumptions A and B are satisfied; then there exists a Cournot-Nash equilibrium of the CLSC network defined by Definition 3.*

Proof. The proof follows easily from Lemmas 5, 6, 7, and 8. □

4. Computing the Cournot-Nash Equilibrium

In this section, we develop two intelligent optimization algorithms, namely, the particle swarm algorithm (proposed by Eberhart and Kennedy [24]) and the Nash genetic algorithm (proposed by Sefrioui and Periaux [30]), denoted by PSO algorithm and GA, respectively, for finding the Cournot-Nash equilibrium of the CLSC network defined by Definition 3. For this purpose, we first need to transform the problem of finding the Cournot-Nash equilibrium into solving sequences of optimization problems with a continuous solution space as follows.

Let X_i^l denote the potential strategy vector of firm i ($i = 1, \ldots, I$) obtained at iteration l of the algorithm (to be presented in the next paragraph). Let

$$X^l = \left(X_1^l, \ldots, X_i^l, \ldots, X_I^l\right) \tag{49}$$

be the strategy vector of all firms obtained at iteration l. In order to obtain the Cournot-Nash equilibrium as defined in Definition 3, at iteration l of the algorithm, we need to find a feasible strategy vector X_i^{l*} ($i = 1, \ldots, I$) which satisfies

$$U_i\left(X_i^{l*}, \widehat{X}_i^{(l-1)*}\right) \geq U_i\left(X_i^l, \widehat{X}_i^{(l-1)*}\right), \tag{50}$$

for all feasible strategy vector X_i^l, where

$$\widehat{X}_i^0 = \left(\overline{X}_1^0, \ldots, \overline{X}_{i-1}^0, \overline{X}_{i+1}^0, \ldots, \overline{X}_I^0\right),$$
$$\widehat{X}_i^{(l-1)*} = \left(X_1^{(l-1)*}, \ldots, X_{i-1}^{(l-1)*}, X_{i+1}^{(l-1)*}, \ldots, X_I^{(l-1)*}\right), \quad l > 1, \tag{51}$$

where $\overline{X}_i^0$ is chosen arbitrarily and $X_i^{(l-1)*}$ denotes the best strategy of firm i obtained at iteration $l-1$ of the algorithm.

Thus, we need to find the feasible strategy vector X_i^{l*} which maximizes the expected profit function $U_i(X_i^l, \widehat{X}_i^{(l-1)*})$ over the set of all the feasible vectors X_i^l. For this purpose, we need to define a fitness function for any strategy vector X_i^l, denoted by fitness (X_i^l), which is obtained by appending a penalty function penal (X_i^l) to the expected profit function $U_i(X_i^l, \widehat{X}_i^{(l-1)*})$. The penalty function penal (X_i^l) is defined by

$$\mathrm{penal}\left(X_i^l\right) \equiv M \cdot G\left(X_i^l\right), \tag{52}$$

where M is a large number and $G(X_i^l)$ denotes the total amount of constraints violations for all the constraints (30)–(34) of the strategy vector X_i^l. (Note that when X_i^l is a feasible strategy vector, then $G(X_i^l) = 0$.) Thus, the fitness function for the strategy vector X_i^l at iteration l of the algorithm can be defined as follows:

$$\mathrm{fitness}\left(X_i^l, \widehat{X}_i^{(l-1)*}\right) \equiv U_i\left(X_i^l, \widehat{X}_i^{(l-1)*}\right) - \mathrm{penal}\left(X_i^l\right). \tag{53}$$

Thus, (53) states that, at iteration l ($l > 0$), the fitness of the strategy vector X_i^l for firm i is defined as "the expected profit obtained by firm i by using the strategy X_i^l, under

the condition that all the other firms are using their best strategies obtained at iteration $(l-1)$ minus the amount of penalty due to the constraints violations of the strategy vector X_i^l."

Thus, the problem of finding the Cournot-Nash equilibrium of the CLSC network is equivalent to solving the following sequences of optimization problems with a continuous solution space, denoted by P_i^l $(l = 1, 2, \ldots)$.

Problem(P_i^l) is as follows:

$$\max \text{ fitness}\left(X_i^l, \widehat{X}_i^{(l-1)*}\right),\ i = 1, \ldots, I, \tag{54}$$

where $\widehat{X}_i^{0*}$ and $\widehat{X}_i^{(l-1)*}$ are as defined in (51). For the sake of simplicity, we replace fitness $(X_i^l, \widehat{X}_i^{(l-1)*})$ by fitness (X_i^l).

Let

$$\Delta\left(X^{l*}\right) = \sum_{i=1}^{I} \left\| \text{fitness}\left(X_i^{l*}\right) - \text{fitness}\left(X_i^{(l-1)*}\right) \right\| \tag{55}$$

be the error function, where $X^{l*} = (X_1^{l*}, \ldots, X_I^{l*})^T$. Suppose that $\Delta(X^{l*}) = 0$; then X^{l*} is the Cournot-Nash equilibrium of the CLSC network. In other words, for all $i = 1, 2, \ldots, I$, when the optimal solutions of two consecutive optimization problems P_i^{l-1} and P_i^l are sufficiently close to each other, then we arrive at the Cournot-Nash equilibrium of the CLSC network.

We will use both PSO algorithm and GA to solve the above sequences of optimization problems. We now define the following terms for PSO algorithm.

Let

$$p^{\text{IB}} \equiv \left(p_1^{\text{IB}}, \ldots, p_i^{\text{IB}}, \ldots, p_I^{\text{IB}}\right)^T \tag{56}$$

be the individual best strategy vector found by a particle, where p_i^{IB} is the best strategy vector for firm i found by this particle. Let

$$p^{\text{GB}} \equiv \left(p_1^{\text{GB}}, \ldots, p_i^{\text{GB}}, \ldots, p_I^{\text{GB}}\right)^T \tag{57}$$

be the global best strategy vector among all the particles in the swarm, where p_i^{GB} is the best strategy vector for firm i among all the particles in the swarm. Let

$$v \equiv \left(v_1, \ldots, v_i, \ldots, v_I\right)^T \tag{58}$$

be the velocity vector which represents both the distance and the direction that should be traveled by the particle from its current position. Let swarm_size denote the swarm size of the particle swarm. Let ω denote the weight representing the trade-off between global exploration and local exploitation abilities of the swarm. Let c_1 and c_2 denote the weight representing the stochastic acceleration terms that pull each particle toward p^{IB} and p^{GB} positions of PSO algorithm, respectively. Let ε be a small number. Let N_{end} be the number of successive iterations that criteria $\Delta(X^{l*}) < \varepsilon$ needed to be satisfied before we can ensure the convergence of PSO algorithm. PSO algorithm can now be formally stated as follows.

PSO Algorithm. Input parameters swarm_size, ω, c_1, c_2, $v_{\min} \in \mathbb{R}^{\sum_{i=1}^{I}(n_M^i \times n_D^i + n_C^i)n_R J + IJ}$, $v_{\max} \in \mathbb{R}^{\sum_{i=1}^{I}(n_M^i \times n_D^i + n_C^i)n_R J + IJ}$, ε, N_{end}, and $x_{\max,i}$ $(i = 1, \ldots, I)$ with $0 \le x_{\max,i} \le \overline{M}_i$, where $\overline{M}_i$ is a large number.

Iteration 0. Choose initial velocity vectors v^0 $(v_{\min} \le v^0 \le v_{\max})$. For each $i = 1, \ldots, I$, choose initial swarm_size strategy vectors to form the initial swarm swarm_i^0. For each particle $X_i^0 \in \text{swarm}_i^0$, let $p_i^{\text{IB}} = X_i^0$ be the best strategy vector found by the above particle. Then

$$p^{\text{IB}} \equiv \left(p_1^{\text{IB}}, \ldots, p_i^{\text{IB}}, \ldots, p_I^{\text{IB}}\right)^T \tag{59}$$

is the individual best strategy vector found by the above particle.

Compute fitness (X_i^0) for each $X_i^0 \in \text{swarm}_i^0$ to obtain the best strategy vector X_i^{0*} for firm i.

Let $p_i^{\text{GB}} = X_i^{0*}$. Then

$$p^{\text{GB}} \equiv \left(p_1^{\text{GB}}, \ldots, p_i^{\text{GB}}, \ldots, p_I^{\text{GB}}\right)^T \tag{60}$$

is the global best strategy vector among all particles in the swarm.

Let $l = 1$.

Iteration l

Step 1. For each $i = 1, \ldots, I$, update each particle in the swarm $\text{swarm}_i^{(l-1)}$ by using the following:

$$\begin{aligned}
v^l &= \omega \times v^{l-1} + c_1\xi_1\left(p^{\text{IB}} - X^{l-1}\right) \\
&\quad + c_2\xi_2\left(p^{\text{GB}} - X^{l-1}\right), \\
&\text{If } v^l > v_{\max}, \quad \text{then } v^l = v_{\max}, \\
&\text{if } v^l < v_{\min}, \quad \text{then } v^l = v_{\min}, \\
v^l &= \left(v_1^l, v_2^l, \ldots, v_I^l\right), \\
X_i^l &= X_i^{l-1} + v_i^l, \\
&\text{if } X_i^l > x_{\max,i}, \quad \text{then } X_i^l = x_{\max,i}, \\
&\text{if } X_i^l < 0, \quad \text{then } X_i^l = 0,
\end{aligned} \tag{61}$$

where $\xi_1, \xi_2 \in U[0, 1]$ are pseudorandom number. The new swarm obtained is denoted by swarm_i^l.

Step 2. For each $i = 1, \ldots, I$, compute fitness (X_i^l) for each $X_i^l \in \text{swarm}_i^l$. If

$$\text{fitness}\left(X_i^l\right) > \text{fitness}\left(p_i^{\text{IB}}\right), \tag{62}$$

then $p_i^{\mathrm{IB}} = X_i^l$. Update the individual best vector found by each particle in the swarm as follows:

$$p^{\mathrm{IB}} \equiv \left(p_1^{\mathrm{IB}}, \ldots, p_i^{\mathrm{IB}}, \ldots, p_I^{\mathrm{IB}}\right)^T. \tag{63}$$

Step 3. For each $i = 1, \ldots, I$, compare fitness (X_i^l) for each $X_i^l \in \mathrm{swarm}_i^l$ to obtain the best strategy vector X_i^{l*}. Hence obtain the best strategy vector $X^{l*} = (X_1^{l*}, \ldots, X_I^{l*})^T$. If

$$\mathrm{fitness}\left(X_i^{l*}\right) > \mathrm{fitness}\left(p_i^{\mathrm{GB}}\right), \tag{64}$$

then $p_i^{\mathrm{GB}} = X_i^{l*}$. Update the global best strategy vector found by all particles in the swarm as follows:

$$p^{\mathrm{GB}} \equiv \left(p_1^{\mathrm{GB}}, \ldots, p_i^{\mathrm{GB}}, \ldots, p_I^{\mathrm{GB}}\right)^T. \tag{65}$$

Step 4. If $l < N_{\mathrm{end}}$, let $l = l+1$ and repeat iteration l; otherwise go to Step 5.

Step 5. Compute $\Delta(X^{l*})$. If $\Delta(X^{\bar{l}*}) < \varepsilon$ for all $\bar{l} = l - N_{\mathrm{end}} + 1, \ldots, l$, the Cournot-Nash equilibrium of the CLSC network is reached, which is given by $X^* = X^{l*}$. For each $i = 1, \ldots, I$, compute fitness (X_i^*) to obtain the optimal expected profit of firm i, stop; otherwise, let $l = l + 1$ and repeat iteration l.

We now define the following parameters for GA.

Let pop_size, P_c, and P_m denote the population size, the crossover probability, and the mutation probability of the genetic algorithm, respectively. Let ε be a small number. Let N_{end} be the number of successive iterations that criteria $\Delta(X^{l*}) < \varepsilon$ needed to be satisfied before we can ensure the convergence of GA. GA can now be formally stated as follows.

GA. Input parameters pop_size, P_c, P_m, ε, and N_{end}.

Iteration 0. For each $i = 1, \ldots, I$, choose initial pop_size strategy vectors to form the initial population Pop_i^0 based on real-coded genetic algorithm. Compute fitness (X_i^0) for each $X_i^0 \in \mathrm{Pop}_i^0$ to obtain the best strategy vector X_i^{0*}. Let $l = 1$.

Iteration l

Step 1. For each $i = 1, \ldots, I$, update the population $\mathrm{Pop}_i^{(l-1)}$ by spinning the roulette wheel with a bias towards selecting fitter strategy vectors to form the new population.

Step 2. For each $i = 1, \ldots, I$, update the population $\mathrm{Pop}_i^{(l-1)}$ by using the crossover operation and the mutation operation given in Appendix A. The new population obtained is denoted by Pop_i^l.

Step 3. For each $i = 1, \ldots, I$, compute fitness (X_i^l) for each $X_i^l \in \mathrm{Pop}_i^l$ to obtain the best strategy vector X_i^{l*}. Hence obtain the best strategy vector for all the firms $X^{l*} = (X_1^{l*}, \ldots, X_I^{l*})^T$.

Step 4. If $l < N_{\mathrm{end}}$, let $l = l+1$ and repeat iteration l; otherwise go to Step 5.

Step 5. Compute $\Delta(X^{l*})$. If $\Delta(X^{\bar{l}*}) < \varepsilon$, for all $\bar{l} = l - N_{\mathrm{end}} + 1, \ldots, l$, the Cournot-Nash equilibrium of the CLSC network is reached, which is given by $X^* = X^{l*}$. For each $i = 1, \ldots, I$, compute fitness (X_i^*), which is the optimal expected profit of firm i, stop; otherwise, let $l = l + 1$ and repeat iteration l.

Note that PSO algorithm and GA developed in this paper can always find the Cournot-Nash equilibrium in a certain number of iterations, even when the expected profit function $U_i(\cdot)$ $(i = 1, \ldots, I)$ defined by (28) is nondifferentiable.

5. Numerical Examples

In this section, two numerical examples are used to compare the efficiencies of the PSO algorithm, the genetic algorithm, and an algorithm based on variational inequalities for finding the Cournot-Nash equilibrium of the CLSC network. (The theory of variational inequality method is given in Appendix B.) In the first example (Example 1), all the given cost functions are differentiable. Thus, we can find the Cournot-Nash equilibrium by PSO algorithm, GA, and Euler algorithm (Dupuis and Nagurney [11]) based on variational inequality (Euler algorithm is given in Appendix C). In the second example (Example 2), the given cost functions corresponding to the manufacturing of the new products are not everywhere differentiable. We use this example to illustrate that both PSO algorithm and GA can still solve nonsmooth optimization problem efficiently, but the algorithm based on variational inequality is not efficient.

We consider a CLSC network involving oligopolistic competition among four firms. Each firm manufactures two products and has two manufacturers and two distribution centers to supply goods to three demand markets. Each firm also has two recovery centers for recycling the used products. For each of the examples solved in this section, we use the same demand price functions as those given by Zhou et al. [15].

Example 1 (comparison of the efficiencies of PSO algorithm, GA, and the Euler algorithm based on variational inequality for solving problems with smooth cost functions). In this example, the demand price functions are as given by Zhou et al. [15].

The operation cost of product j in the forward logistics (which is a function of the product flow of product j on all the paths in the forward logistics) is as follows:

$$\begin{aligned}\widehat{c}_{jp^f}\left(x_{jP}\right) &= 2\left(x_{jp^f}\right)^2 + [0.2i + 0.5k]\, x_{jp^f} \\ &\quad + \sum_{i=1}^{I}\sum_{k=1}^{n_R}\sum_{p^f \in P_k^i} x_{jp^f},\end{aligned} \tag{66}$$

where $p^f \in P_k^i$ $(i = 1, 2, 3, 4,\ j = 1, 2,$ and $k = 1, 2, 3)$. The above operation cost is modified from that of Zhou et al.'s paper [15] by adding the last two terms of (66) to capture the competition among firms for the optimal product flows in the forward logistics.

The operation cost of product j in the reverse logistics is as follows:

$$\widehat{c}_{jp^r}\left(x_{jp^r}\right) = 0.2\left(x_{jp^r}\right)^2 + [0.7i + 0.3k]\, x_{jp^r}, \quad (67)$$

for any path p^r connecting demand market R_k to firm i via the recovery center C_1^i and

$$\widehat{c}_{jp^r}\left(x_{jp^r}\right) = 0.2\left(x_{jp^r}\right)^2 + [0.3i + 0.7k]\, x_{jp^r}, \quad (68)$$

for any path p^r connecting demand market R_k to firm i via the recovery center C_2^i, where $p^r \in \widehat{P}_k^i$ ($i = 1, 2, 3, 4$ and $k = 1, 2, 3$).

The manufacturing cost of new products and the remanufacturing cost of returned products, which are exactly the same as those in Zhou et al.'s paper [15], are as follows:

$$\begin{aligned} \overline{f}_{ij}\left(x_{ij}^{\mathrm{New}}\right) &= 2.5\left(x_{ij}^{\mathrm{New}}\right)^2 + 2x_{ij}^{\mathrm{New}}, \\ \overline{f}_{ij}^{\mathrm{Re}}\left(x_{ij}^{\mathrm{Re}}\right) &= \left(x_{ij}^{\mathrm{Re}}\right)^2 + 0.5x_{ij}^{\mathrm{Re}}. \end{aligned} \quad (69)$$

where $i = 1, 2, 3, 4$, $j = 1, 2$, and $k = 1, 2, 3$.

Assume that $\widehat{d}_{ijk}$ is uniformly distributed in $[0, \tau_{ijk}]$ with probability density function given by

$$\mathscr{F}_{ijk}(x) = \begin{cases} \dfrac{1}{\tau_{ijk}}, & \text{if } x \in \left[0, \tau_{ijk}\right], \\ 0, & \text{if } x \in \left(\tau_{ijk}, +\infty\right), \end{cases} \quad (70)$$

where, for product $j = 1$, $\tau_{1jk} = 28$, $\tau_{2jk} = 27$, $\tau_{3jk} = 26$, and $\tau_{4jk} = 25$ ($k = 1, 2, 3$) and, for product $j = 2$, $\tau_{1jk} = 20$, $\tau_{2jk} = 19$, $\tau_{3jk} = 18$, and $\tau_{4jk} = 17$ ($k = 1, 2, 3$).

Assume that $\widehat{r}_{kji}$ is uniformly distributed in $[0, \widehat{\tau}_{kji}]$ with probability density function given by

$$\mathscr{F}_{kji}^r(x) = \begin{cases} \dfrac{1}{\widehat{\tau}_{kji}}, & \text{if } x \in \left[0, \widehat{\tau}_{kji}\right], \\ 0, & \text{if } x \in \left(\widehat{\tau}_{kji}, +\infty\right), \end{cases} \quad (71)$$

where, for product $j = 1$, $\widehat{\tau}_{kji} = 8$ and, for product $j = 2$, $\widehat{\tau}_{kji} = 6$ ($k = 1, 2, 3$ and $i = 1, 2, 3, 4$).

The values of the parameters are as follows:

θ_{ijk}^+ (unit penalty incurred on firm i due to excessive supply of product j to demand market R_k) = 20,

θ_{ijk}^- (unit penalty incurred on firm i due to insufficient supply of product j to demand market R_k) = 20,

ρ_j^{Re} (purchase cost per item of returned product j from demand market R_k to firm i) = 10,

$\overline{\rho}$ (disposal fee per item of the used products at the landfill site) = 10,

$\overline{x}$ (the maximum quantity of new product j manufactured by firm i) = 50,

where $i = 1, 2, 3, 4$, $j = 1, 2$, and $k = 1, 2, 3$.

We have solved Example 1 by PSO algorithm, GA, and the Euler algorithm based on the variational inequality. All the optimal strategies in the forward logistics and the reverse logistics at the Cournot-Nash equilibrium are depicted in Tables 1 and 2, respectively.

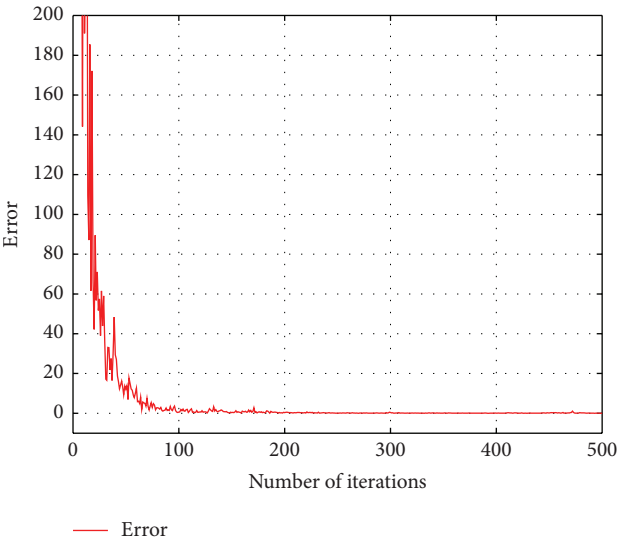


Figure 2: Values of the error function for Example 1 by PSO algorithm.

The fourth column of Tables 1 and 2 represents the mean values of the optimal strategies of all the firms obtained by PSO algorithm, using the following input parameters:

$$\begin{aligned} \text{swarm_size} &= 100, \\ \omega &= 0.6, \\ c_1 &= 2, \\ c_2 &= 2, \\ v_{\max} &= 4, \\ v_{\min} &= -4, \\ \varepsilon &= 0.01, \\ N_{\mathrm{end}} &= 100. \end{aligned} \quad (72)$$

The mean values of the expected profits obtained by firm i ($i = 1, 2, 3, 4$) at the Cournot-Nash equilibrium, denoted by $U_{\mathrm{PSO},i}^*$, are

$$\begin{aligned} U_{\mathrm{PSO},1}^* &= 15716, \\ U_{\mathrm{PSO},2}^* &= 15395, \\ U_{\mathrm{PSO},3}^* &= 14636, \\ U_{\mathrm{PSO},4}^* &= 14451. \end{aligned} \quad (73)$$

Hence, the mean value of the total expected profit of all firms is 60234.

Figure 2 shows the values of the error function $\Delta(X^{l*})$ and Figure 3 shows all firms' fitness (i.e., fitness (X_1^l), fitness (X_2^l),

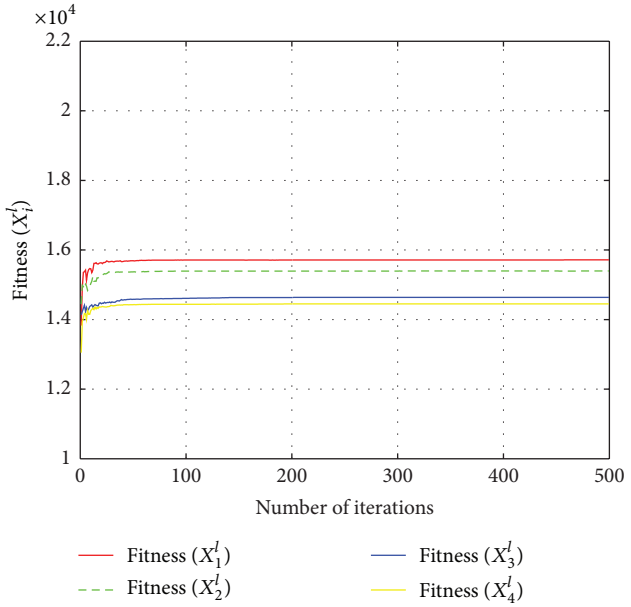


FIGURE 3: All firms' fitness for Example 1 by PSO algorithm.

fitness (X_3^l), and fitness (X_4^l)) from iteration 0 to 500 for a particular result generated by PSO algorithm.

The sixth column of Tables 1 and 2 represents the mean values of the optimal strategies of all the firms obtained by GA, using the following input parameters:

$$\begin{aligned} \text{pop_size} &= 100, \\ P_c &= 0.6, \\ P_m &= 0.01, \\ \varepsilon &= 0.01, \\ N_{\text{end}} &= 100. \end{aligned} \tag{74}$$

The mean values of the expected profits obtained by firm i ($i = 1, 2, 3, 4$) at the Cournot-Nash equilibrium, denoted by $U^*_{\text{GA},i}$, are

$$\begin{aligned} U^*_{\text{GA},1} &= 15712, \\ U^*_{\text{GA},2} &= 15361, \\ U^*_{\text{GA},3} &= 14701, \\ U^*_{\text{GA},4} &= 14417. \end{aligned} \tag{75}$$

Hence, the mean value of the total expected profit of all firms is 60191.

Figure 4 shows the values of the error function $\Delta(X^{l*})$ and Figure 5 shows all firms' fitness (i.e., fitness (X_1^l), fitness (X_2^l), fitness (X_3^l), and fitness (X_4^l)) from iteration 0 to 1500 for a particular result generated by GA.

The eighth column of Tables 1 and 2 represents the optimal strategies of all the firms obtained by the Euler algorithm based on the variational inequality. The expected

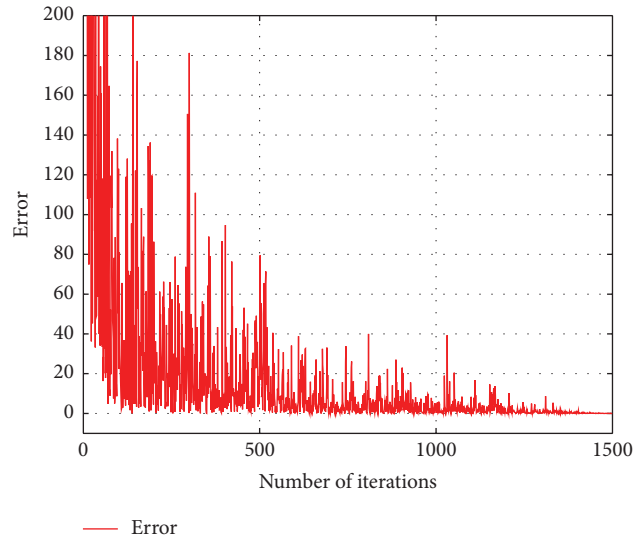


FIGURE 4: Values of the error function for Example 1 by GA.

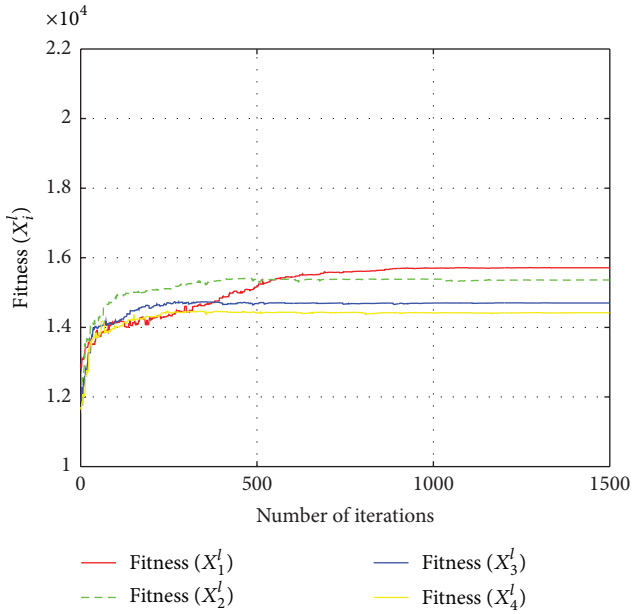


FIGURE 5: All firms' fitness for Example 1 by GA.

profits obtained by firm i ($i = 1, 2, 3, 4$) at the Cournot-Nash equilibrium, denoted by $U^*_{E,i}$, are

$$\begin{aligned} U^*_{E,1} &= 15729, \\ U^*_{E,2} &= 15430, \\ U^*_{E,3} &= 14715, \\ U^*_{E,4} &= 14474. \end{aligned} \tag{76}$$

Hence, the total expected profit of all firms is 60348.

Figure 6 shows all firms' fitness (i.e., fitness (X_1^l), fitness (X_2^l), fitness (X_3^l), and fitness (X_4^l)) obtained by the Euler algorithm from iteration 0 to 1500.

Table 1: Comparison of the optimal strategies obtained by the Euler algorithm with the optimal strategies obtained by PSO algorithm and GA in the forward logistics of Example 1.

			PSO		GA		Euler
			Mean	Stdev.	Mean	Stdev.	
Firm 1	Optimal quantity of product 1 supplied by firm 1 to demand market R_k	$k = 1$	15.4812	0.1763	15.2751	0.3265	15.2807
		$k = 2$	15.3944	0.2543	15.8214	0.3163	15.4540
		$k = 3$	15.1247	0.3744	14.9799	0.5525	15.2798
	Optimal quantity of new manufactured product 1		34.0004	0.2496	34.0874	0.0301	34.0124
	Optimal quantity of product 2 supplied by firm 1 to demand market R_k	$k = 1$	11.8935	0.2569	11.2828	0.7107	12.0464
		$k = 2$	11.6125	0.3073	12.3592	0.5675	12.1290
		$k = 3$	12.5527	0.3890	12.5719	0.7039	12.0301
	Optimal quantity of new manufactured product 2		24.0588	0.4450	24.2513	0.0708	24.2025
Firm 2	Optimal quantity of product 1 supplied by firm 2 to demand market R_k	$k = 1$	14.8362	0.4743	14.4733	0.3827	15.0069
		$k = 2$	14.4819	0.5303	14.7227	0.2718	15.0784
		$k = 3$	14.8862	0.3080	14.9882	0.2961	14.8107
	Optimal quantity of new manufactured product 1		32.2042	0.2463	32.1870	0.0825	32.8937
	Optimal quantity of product 2 supplied by firm 2 to demand market R_k	$k = 1$	11.8696	0.2370	11.4820	0.7778	11.8671
		$k = 2$	11.6569	0.2564	11.9422	0.4798	11.8944
		$k = 3$	11.4962	0.3879	11.7069	0.8136	11.7576
	Optimal quantity of new manufactured product 2		23.0228	0.3726	23.3518	0.2146	23.5153
Firm 3	Optimal quantity of product 1 supplied by firm 3 to demand market R_k	$k = 1$	15.2002	0.3042	14.7245	0.6152	14.5303
		$k = 2$	13.6609	0.5233	14.3239	0.7908	14.5972
		$k = 3$	14.1390	0.3938	14.5945	0.8268	14.3372
	Optimal quantity of new manufactured product 1		31.0001	0.1009	31.6438	0.4290	31.4625
	Optimal quantity of product 2 supplied by firm 3 to demand market R_k	$k = 1$	11.7620	0.3609	10.7958	0.4134	11.5539
		$k = 2$	11.9596	0.3486	11.8798	0.4157	11.5755
		$k = 3$	11.5913	0.3544	11.4596	0.7349	11.4546
	Optimal quantity of new manufactured product 2		23.3130	0.4206	22.1509	0.0989	22.5796
Firm 4	Optimal quantity of product 1 supplied by firm 4 to demand market R_k	$k = 1$	13.7444	0.3978	13.5301	0.4678	14.1712
		$k = 2$	14.1173	0.2761	14.1522	0.8036	14.1888
		$k = 3$	14.2150	0.4706	13.7095	0.7046	14.0771
	Optimal quantity of new manufactured product 1		30.0767	0.5645	30.4098	0.1273	30.4346
	Optimal quantity of product 2 supplied by firm 4 to demand market R_k	$k = 1$	11.1741	0.2702	10.2537	0.8623	11.2665
		$k = 2$	11.2078	0.2615	11.8534	0.7726	11.2646
		$k = 3$	12.0043	0.2985	11.4950	0.7313	11.2136
	Optimal quantity of new manufactured product 2		22.3862	0.5094	21.6579	0.2226	21.7392

From the values of $U^*_{\mathrm{PSO},i}$, $U^*_{\mathrm{GA},i}$, and $U^*_{E,i}$ given in the previous paragraph, we observe that, on the average, the optimal expected profits of each of the firms obtained by each of the three algorithms are almost the same, with the total optimal expected profit (of all firms) obtained by the Euler algorithm being marginally higher than that obtained by PSO algorithm (by 0.33%) and by GA (by 0.41%). Thus, on the average, the efficiency of the PSO algorithm, the genetic algorithm, and the Euler algorithm (based on the variational inequality) is almost the same, in terms of the accuracy of the computed equilibrium.

From Figure 6, we observe that the Euler algorithm (based on the variational inequality) converges to the Cournot-Nash equilibrium of the CLSC network in 1500

Table 2: Comparison of the optimal strategies obtained by the Euler algorithm with the optimal strategies obtained by PSO algorithm and GA in the reverse logistics of Example 1.

			PSO		GA		Euler
			Mean	Stdev.	Mean	Stdev.	
Firm 1	Optimal percentage of product 1 returning from demand market R_k to firm 1 via recovery center C_1^1	$k = 1$	0.5014	0.0289	0.5036	0.0187	0.5000
		$k = 2$	0.5240	0.0243	0.5545	0.0121	0.5937
		$k = 3$	0.6922	0.0339	0.6201	0.0172	0.6875
	Optimal percentage of product 2 returning from demand market R_k to firm 1 via recovery center C_1^1	$k = 1$	0.4938	0.0313	0.5258	0.0183	0.5000
		$k = 2$	0.6306	0.0212	0.6133	0.0253	0.6250
		$k = 3$	0.7990	0.0300	0.7686	0.0242	0.7500
Firm 2	Optimal percentage of product 1 returning from demand market R_k to firm 2 via recovery center C_1^2	$k = 1$	0.4124	0.0281	0.4216	0.0161	0.4063
		$k = 2$	0.5515	0.0162	0.5235	0.0103	0.5000
		$k = 3$	0.5958	0.0331	0.5309	0.0090	0.5937
	Optimal percentage of product 2 returning from demand market R_k to firm 2 via recovery center C_1^2	$k = 1$	0.3626	0.0235	0.3118	0.0213	0.3750
		$k = 2$	0.5039	0.0339	0.5100	0.0159	0.5000
		$k = 3$	0.6709	0.0338	0.6086	0.0219	0.6250
Firm 3	Optimal percentage of product 1 returning from demand market R_k to firm 3 via recovery center C_1^3	$k = 1$	0.3098	0.0296	0.3770	0.0175	0.3125
		$k = 2$	0.3992	0.0285	0.4551	0.0108	0.4063
		$k = 3$	0.5000	0.0237	0.5092	0.0122	0.5000
	Optimal percentage of product 2 returning from demand market R_k to firm 3 via recovery center C_1^3	$k = 1$	0.2012	0.0236	0.2647	0.0182	0.2500
		$k = 2$	0.3534	0.0321	0.3434	0.0156	0.3750
		$k = 3$	0.5028	0.0258	0.5078	0.0154	0.5000
Firm 4	Optimal percentage of product 1 returning from demand market R_k to firm 4 via recovery center C_1^4	$k = 1$	0.2128	0.0170	0.2099	0.0186	0.2188
		$k = 2$	0.3749	0.0231	0.3623	0.0199	0.3125
		$k = 3$	0.4598	0.0315	0.4481	0.0100	0.4063
	Optimal percentage of product 2 returning from demand market R_k to firm 4 via recovery center C_1^4	$k = 1$	0.1238	0.0317	0.1118	0.0317	0.1250
		$k = 2$	0.2219	0.0224	0.2399	0.0133	0.2500
		$k = 3$	0.3221	0.0224	0.3182	0.0182	0.3750

Note: optimal percentage of product j returning from demand market R_k to firm i via recovery center C_2^i = 1 − optimal percentage of product j returning from demand market R_k to firm i via recovery center C_1^i ($\forall i = 1, 2, 3, 4$, $j = 1, 2$, and $k = 1, 2, 3$).

iterations. Such convergence of the sequence of solutions obtained by the Euler algorithm to the optimal solution of variational inequality has been proved by Dupuis and Nagurney [11] and Nagurney and Zhang [31]. On the other hand, the convergence of the sequence of solutions obtained by PSO algorithm (which converges in 500 iterations) and by GA (which converges in 1500 iterations) to the Cournot-Nash equilibrium is illustrated in Figure 2 to Figure 5. In terms of computational effort, the computational time required by the Euler algorithm to obtain the equilibrium of the CLSC network is 30.75 seconds, whereas the average computational time required by PSO algorithm and GA is 39.04 seconds and 115.18 seconds, respectively. Thus, on the average, the computational time required by PSO algorithm is slightly longer than that required by the Euler algorithm but is much shorter than that required by GA to obtain the equilibrium. Thus, the PSO algorithm is just as efficient as the Euler algorithm but is more efficient than the genetic algorithm, in terms of the computational time required to obtain the equilibrium.

From this example, the two heuristics (PSO and GA) are effective in searching for a near-optimal solution despite the additional computational time as compared to the Euler algorithm. However, in reality, the production cost functions are not everywhere differentiable and, hence, the Euler algorithm is not applicable. In the next example, Example 2, some given functions are not everywhere differentiable and therefore we cannot find the Cournot-Nash equilibrium of the CLSC network by the Euler algorithm. PSO algorithm and GA, which do not require the gradient information of both the objective functions and the constraint functions, can still obtain the Cournot-Nash equilibrium efficiently.

Example 2 (comparison of the efficiencies of PSO algorithm and GA for solving problems with nonsmooth cost functions). It is the same as Example 1, except that the production cost functions $\overline{f}_{ij}$ in Example 1 are now being replaced by

$$\overline{f}_{ij}\left(x_{ij}^{\text{New}}\right) = \begin{cases} 3.5\left(x_{ij}^{\text{New}}\right)^2 + 2.5x_{ij}^{\text{New}}, & \text{if } x_{ij}^{\text{New}} \le 10,\ i = 1,\ldots,4,\ j = 1,2; \\ 3.5\left(x_{ij}^{\text{New}}\right)^2 + x_{ij}^{\text{New}} + 15, & \text{if } x_{ij}^{\text{New}} > 10,\ i = 1,\ldots,4,\ j = 1,2. \end{cases} \tag{77}$$

(In other words, the differentiable production cost functions in Example 1 are now being replaced by production cost functions which are not everywhere differentiable.) Thus, the main purpose of this example is to test the efficiency of PSO algorithm and GA for finding the Cournot-Nash equilibrium for problems involving nondifferentiable cost functions.

With the same input parameters as those used in Example 1, we run PSO algorithm and GA to obtain the near-optimal strategies. All the optimal strategies in the forward logistics and the reverse logistics at the Cournot-Nash equilibrium are depicted in Tables 3 and 4, respectively.

The mean values of the expected profits obtained by firm i ($i = 1, 2, 3, 4$) at the Cournot-Nash equilibrium, denoted by $U_{\text{PSO},i}^*$ and $U_{\text{GA},i}^*$, respectively, are

$$\begin{aligned} U_{\text{PSO},1}^* &= 14377, \\ U_{\text{PSO},2}^* &= 14140, \\ U_{\text{PSO},3}^* &= 13550, \\ U_{\text{PSO},4}^* &= 13376, \\ U_{\text{GA},1}^* &= 14364, \\ U_{\text{GA},2}^* &= 14138, \\ U_{\text{GA},3}^* &= 13500, \\ U_{\text{GA},4}^* &= 13367. \end{aligned} \tag{78}$$

Hence, the mean value of the total expected profit of all firms obtained by PSO algorithm is 55443 and that obtained by GA is 55369.

From the values of $U_{\text{PSO},i}^*$ and $U_{\text{GA},i}^*$ given in the previous paragraph, we observe that, on the average, the optimal expected profits of each of the firms obtained by each of the two algorithms are almost the same, with the total optimal expected profit (of all the firms) obtained by PSO algorithm being marginally higher (by 0.13%) than that obtained by GA.

Figures 7 and 9 show the values of the error functions obtained by PSO algorithm and GA, respectively, whereas Figures 8 and 10 show all firms' fitness obtained by the above two algorithms, respectively. From Figures 7 and 9, we observe that PSO algorithm and GA converge to the Cournot-Nash equilibrium of the CLSC network in 500 iterations and 2000 iterations, respectively. In terms of computational effort, the average computational time required by PSO algorithm and GA is 35.59 seconds and 121.51 seconds, respectively. Hence, the average computational time required by the PSO algorithm to obtain the Cournot-Nash equilibrium of the CLSC network is much shorter than that required by the genetic algorithm. Similar to the conclusion of Example 1, the PSO algorithm is more efficient than the genetic algorithm, in terms of computational effort.

Remark 3. From Examples 1 and 2, we conclude that as long as the given functions are continuous (it does not matter whether they are differentiable everywhere or not), on the average, the efficiency of the PSO algorithm and the genetic algorithm is almost the same, in terms of the accuracy of the computed equilibrium; however, the PSO algorithm is more efficient than the genetic algorithm, in terms of the computational time required to obtain the equilibrium. This difference in efficiencies between the two algorithms in terms of speed is probably due to the fact that genetic algorithms are more suitable for solving combinatorial problems than problems with a continuous solution space. (Note that the problem of finding the Cournot-Nash equilibrium of the CLSC network in our paper is transformed into solving sequences of optimization problems with a continuous solution space in Section 4.)

Remark 4. Since the production cost functions in this example are continuous but not everywhere differentiable, we cannot get the variational inequality formulation of this example and thus cannot obtain the Cournot-Nash equilibrium by the variational inequality method. Thus, PSO algorithm and GA of this paper are particularly useful for solving this nonsmooth problem, because they do not require the gradient information of both the objective functions and the constraint functions.

6. Conclusion and Suggestions for Further Studies

In this paper, we develop stochastic multiproducts closed-loop supply chain network equilibrium model involving oligopolistic competition for multiproducts and their product flow routings in the entire supply chain. This model belongs to the context of oligopolistic firms that compete noncooperatively in a Cournot-Nash framework under a stochastic environment. Since the problem of finding the Cournot-Nash for the oligopolistic competition CLSC network model in our paper can be transformed into solving sequences of optimization problems with a continuous solution space, both the PSO algorithm and the genetic algorithm can be used for finding the above equilibrium. The numerical results show that when all the given functions are differentiable,

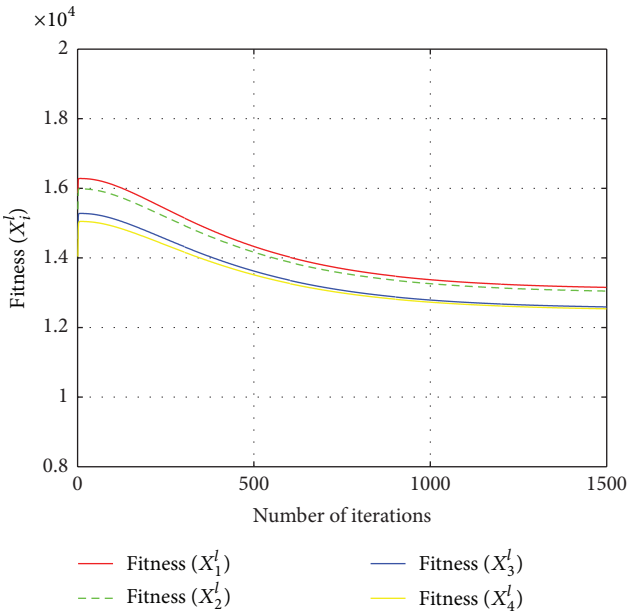


Figure 6: All firms' fitness for Example 1 by Euler algorithm.

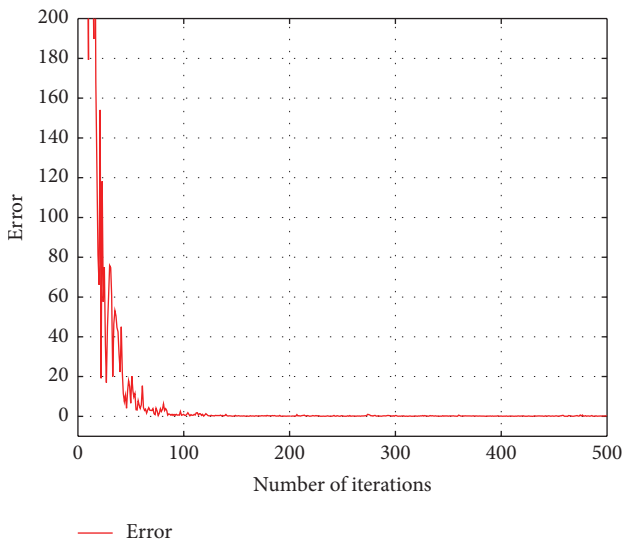


Figure 7: Values of the error function for Example 2 by PSO algorithm.

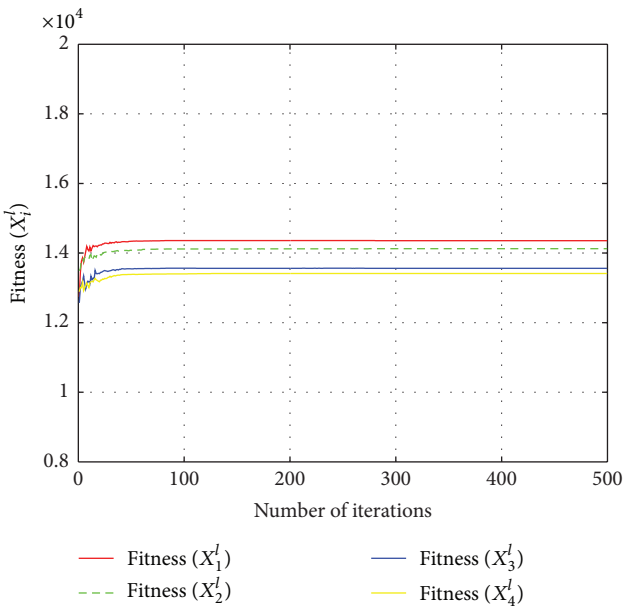


Figure 8: All firms' fitness for Example 2 by PSO algorithm.

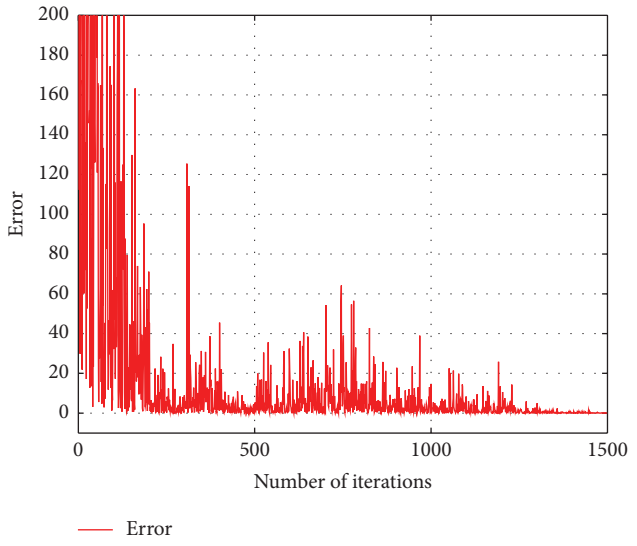


Figure 9: Values of the error function for Example 2 by GA.

the efficiencies of the PSO algorithm, the genetic algorithm, and an algorithm based on variational inequality are almost the same, in terms of the accuracy of the computed equilibrium. However, the PSO algorithm is just as efficient as the algorithm based on variational inequality but more efficient than the genetic algorithm, in terms of the computational time required to obtain the equilibrium. For problems involving nondifferentiable cost functions, the numerical results show that the PSO algorithm and the genetic algorithm can still find the Cournot-Nash equilibrium efficiently, but the algorithm based on variational inequality is not efficient. Thus, we conclude that both the PSO algorithm and the genetic algorithm can solve optimization problems with a continuous solution space efficiently, but the PSO algorithm is faster than the genetic algorithm. In the future, we would like to apply the PSO algorithm and the genetic algorithm to solve problems involving a multiperiod CLSC network under oligopolistic competition among firms.

Appendices

A. The Crossover Operation and the Mutation Operation

In the following crossover and mutation operation, the term "chromosome" is used to represent the strategy vector of firm i (denoted by X_i, where $X_i \in K_i$).

Table 3: Comparison of the optimal strategies obtained by PSO algorithm and GA in the forward logistics of Example 2.

			PSO		GA	
			Mean	Stdev.	Mean	Stdev.
Firm 1	Optimal quantity of product 1 supplied by firm 1 to demand market R_k	$k = 1$	13.0209	0.3023	14.4351	0.7513
		$k = 2$	14.0660	0.3063	13.3885	0.4320
		$k = 3$	13.2913	0.2571	13.5359	0.6898
	Optimal quantity of new manufactured product 1		28.3783	0.3691	29.3675	0.2781
	Optimal quantity of product 2 supplied by firm 1 to demand market R_k	$k = 1$	10.6559	0.2370	10.6977	0.5947
		$k = 2$	10.1936	0.3070	11.1715	0.7511
		$k = 3$	11.7843	0.3214	10.9480	0.3460
	Optimal quantity of new manufactured product 2		20.6338	0.3454	20.8259	0.2688
Firm 2	Optimal quantity of product 1 supplied by firm 2 to demand market R_k	$k = 1$	13.0987	0.1915	12.7234	0.6973
		$k = 2$	12.9814	0.2434	13.7705	0.0993
		$k = 3$	12.9200	0.2846	13.5463	0.2621
	Optimal quantity of new manufactured product 1		27.0000	0.4155	28.0405	0.6897
	Optimal quantity of product 2 supplied by firm 2 to demand market R_k	$k = 1$	11.3123	0.3327	9.8867	0.7544
		$k = 2$	10.6753	0.3787	11.4666	0.9035
		$k = 3$	10.3289	0.3804	11.2320	0.8300
	Optimal quantity of new manufactured product 2		20.3165	0.3663	20.5914	0.5499
Firm 3	Optimal quantity of product 1 supplied by firm 3 to demand market R_k	$k = 1$	12.5400	0.4467	12.9380	0.6420
		$k = 2$	13.0791	0.4328	13.0966	0.8916
		$k = 3$	12.7047	0.5142	12.0838	0.3959
	Optimal quantity of new manufactured product 1		26.3238	0.3406	26.1193	0.7190
	Optimal quantity of product 2 supplied by firm 3 to demand market R_k	$k = 1$	10.4219	0.4377	10.7817	0.4922
		$k = 2$	10.4076	0.3780	9.8720	0.3852
		$k = 3$	10.6253	0.2417	10.7148	0.3356
	Optimal quantity of new manufactured product 2		19.4547	0.3097	19.3687	0.1064
Firm 4	Optimal quantity of product 1 supplied by firm 4 to demand market R_k	$k = 1$	12.5708	0.3300	11.2457	0.8963
		$k = 2$	13.3946	0.5604	13.1968	0.4776
		$k = 3$	12.1992	0.2689	13.2656	0.0695
	Optimal quantity of new manufactured product 1		26.1646	0.5159	25.7109	0.7179
	Optimal quantity of product 2 supplied by firm 4 to demand market R_k	$k = 1$	10.5663	0.3677	11.0721	0.6788
		$k = 2$	10.0228	0.3526	10.3900	0.6832
		$k = 3$	10.7935	0.3971	11.1403	0.8083
	Optimal quantity of new manufactured product 2		19.3827	0.2282	19.6067	0.7179

Crossover Operation. Consider the following.

Step 1. Among all the chromosomes in a given population of size pop_size, we choose $m_1 = \text{pop_size} \times P_c$ chromosome as parents (where $P_c < 1$), with the probability of each chromosome being selected equal to P_c.

Step 2. We group all the selected parents in Step 1 into pairs and denoted them by $(\text{par}_{1,j}, \text{par}_{2,j})$, $j = 1, \ldots, m_1/2$. Then for each pair of parents, we generate a random number, denoted by rand_1, from the interval $(0, 1)$ and create a pair of children according to these formulae:

$$\begin{aligned} \text{child}_{1,j} &= \text{rand}_1 * \text{par}_{1,j} + (1 - \text{rand}_1) * \text{par}_{2,j}, \\ &\qquad j = 1, \ldots, \frac{m_1}{2}, \\ \text{child}_{2,j} &= (1 - \text{rand}_1) * \text{par}_{1,j} + \text{rand}_1 * \text{par}_{2,j}, \\ &\qquad j = 1, \ldots, \frac{m_1}{2}. \end{aligned} \tag{A.1}$$

Table 4: Comparison of the optimal strategies obtained by PSO algorithm and GA in the reverse logistics of Example 2.

			PSO		GA	
			Mean	Stdev.	Mean	Stdev.
Firm 1	Optimal percentage of product 1 returning from demand market R_k to firm 1 via recovery center C_1^1	$k = 1$	0.5709	0.0269	0.5151	0.0110
		$k = 2$	0.5927	0.0250	0.5211	0.0226
		$k = 3$	0.6977	0.0224	0.6428	0.0145
	Optimal percentage of product 2 returning from demand market R_k to firm 1 via recovery center C_1^1	$k = 1$	0.5790	0.0317	0.5413	0.0043
		$k = 2$	0.6262	0.0218	0.6666	0.0135
		$k = 3$	0.7497	0.0204	0.7562	0.0134
Firm 2	Optimal percentage of product 1 returning from demand market R_k to firm 2 via recovery center C_1^2	$k = 1$	0.4084	0.0352	0.4199	0.0152
		$k = 2$	0.5694	0.0244	0.5242	0.0190
		$k = 3$	0.5859	0.0313	0.5452	0.0180
	Optimal percentage of product 2 returning from demand market R_k to firm 2 via recovery center C_1^2	$k = 1$	0.4000	0.0326	0.3517	0.0158
		$k = 2$	0.5401	0.0231	0.5403	0.0182
		$k = 3$	0.6085	0.0366	0.6510	0.0066
Firm 3	Optimal percentage of product 1 returning from demand market R_k to firm 3 via recovery center C_1^3	$k = 1$	0.3107	0.0235	0.3166	0.0080
		$k = 2$	0.4213	0.0175	0.4384	0.0138
		$k = 3$	0.5590	0.0317	0.5069	0.0249
	Optimal percentage of product 2 returning from demand market R_k to firm 3 via recovery center C_1^3	$k = 1$	0.2819	0.0215	0.2287	0.0071
		$k = 2$	0.3262	0.0318	0.3523	0.0152
		$k = 3$	0.5463	0.0176	0.5463	0.0229
Firm 4	Optimal percentage of product 1 returning from demand market R_k to firm 4 via recovery center C_1^4	$k = 1$	0.2110	0.0315	0.2177	0.0143
		$k = 2$	0.3224	0.0293	0.3159	0.0152
		$k = 3$	0.4405	0.0297	0.4358	0.0094
	Optimal percentage of product 2 returning from demand market R_k to firm 4 via recovery center C_1^4	$k = 1$	0.1726	0.0214	0.1219	0.0080
		$k = 2$	0.2561	0.0221	0.2699	0.0220
		$k = 3$	0.3390	0.0334	0.3169	0.0162

Note: optimal percentage of product j returning from demand market R_k to firm i via recovery center C_2^i = 1 − optimal percentage of product j returning from demand market R_k to firm i via recovery center C_1^i ($\forall i = 1, 2, 3, 4$, $j = 1, 2$, and $k = 1, 2, 3$).

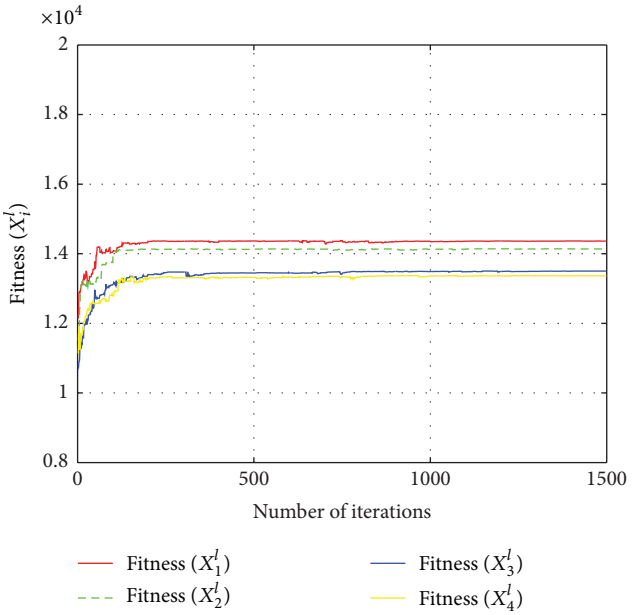


Figure 10: All firms' fitness for Example 2 by GA.

Replace these m_1 parents by their children to obtain the new population.

Mutation Operation. Consider the following.

Step 1. Among all the chromosomes in a given population of size pop_size, we choose m_2 = pop_size × P_m chromosome as parents (where $P_m < 1$), with the probability of each chromosome being selected equal to P_m.

Step 2. Denote each selected parent in Step 1 by par_j, $j = 1, \ldots, m_2$. Then, for each par_j, we choose a feasible direction d_j and create a child according to this formula:

$$\text{child}_j = \text{par}_j + \text{rand}_2 * d_j, \quad j = 1, \ldots, m_2, \tag{A.2}$$

where rand_2 is a random number generated in the interval $(0, 1)$. Replace these m_2 parents by their children to obtain the new population.

B. The Theorem of Variational Inequality Method

Theorem C. *Suppose that Assumptions A and B are satisfied, and suppose that the expected profit function $U_i(X)$ ($i = 1, \ldots, I$) defined by (28) is differentiable with respect to the decision variable X defined by (4). Then the strategy vector of all firms $X^* \in \mathbb{R}_+^{\sum_{i=1}^{I}(n_M^i \times n_D^i + n_C^i)n_R J + IJ} \bigcup \mathbb{R}^{IJK}$ is a Cournot-Nash equilibrium of the CLSC network defined by Definition 3 if there exist $\lambda_1^* \in \mathbb{R}_+^{IJ}$, $\lambda_2^* \in \mathbb{R}_+^{IJ}$, $\lambda_3^* \in \mathbb{R}_+^{IJK}$, and $\lambda_4^* \in \mathbb{R}^{IJK}$, such that the following variational inequality holds:*

$$\begin{aligned}
&\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{n_R}\sum_{p^f \in P_k^i}\left[\frac{\partial \widehat{c}_{jp^f}\left(x_{jP}^*\right)}{\partial x_{jp^f}} + \theta_{ijk}^{+}\Pr\left(\widehat{d}_{ijk} \le v_{ijk}^*\right) + \theta_{ijk}^{-}\left(\Pr\left(\widehat{d}_{ijk} \le v_{ijk}^*\right) - 1\right) - \int_{v_{ijk}}^{+\infty} \overline{G}\left(\widehat{d}_{ijk}\right)\mathscr{F}_{ijk}\left(\widehat{d}_{ijk}\right)\mathrm{d}\widehat{d}_{ijk} + \overline{\rho} + \lambda_{2ij}^* - \lambda_{3ijk}^*\right]\\
&\cdot\left[x_{jp^f} - x_{jp^f}^*\right] + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{n_R}\sum_{p^r \in \widehat{P}_k^i}\left[\frac{\partial E\left[\widehat{c}_{jp^r}\left(y_{jp^r}^*\widehat{r}_{kji}\right)\right]}{\partial y_{jp^r}} - \lambda_{4ijk}^*\right]\left[y_{jp^r} - y_{jp^r}^*\right] + \sum_{i=1}^{I}\sum_{j=1}^{J}\left[\frac{\partial \overline{f}_{ij}\left(x_{ij}^{\mathrm{New}*}\right)}{\partial x_{ij}^{\mathrm{New}}} + \lambda_{1ij}^* - \lambda_{2ij}^*\right]\\
&\cdot\left[x_{ij}^{\mathrm{New}} - x_{ij}^{\mathrm{New}*}\right] + \sum_{i=1}^{I}\sum_{j=1}^{J}\left[\overline{x} - x_{ij}^{\mathrm{New}*}\right] \times \left[\lambda_{1ij} - \lambda_{1ij}^*\right] + \sum_{i=1}^{I}\sum_{j=1}^{J}\left[x_{ij}^{\mathrm{New}*} + E\left(x_{ij}^{\mathrm{Re}}\right) - \sum_{k=1}^{n_R}\sum_{p^f \in P_k^i} x_{jp^f}^*\right]\left[\lambda_{2ij} - \lambda_{2ij}^*\right]\\
&+ \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{n_R}\left[\sum_{p^f \in P_k^i} x_{jp^f}^* - E\left(r_{kji}\right)\right]\left[\lambda_{3ijk} - \lambda_{3ijk}^*\right] + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{n_R}\left[\sum_{p^r \in \widehat{P}_k^i} y_{jp^r} - 1\right]\left[\lambda_{4ijk} - \lambda_{4ijk}^*\right] \ge 0,\\
&\qquad \forall\left(X; \lambda_1; \lambda_2; \lambda_3; \lambda_4\right) \in \mathbb{R}_+^{\sum_{i=1}^{I}(n_M^i \times n_D^i)n_R J + 3IJ + IJK} \bigcup \mathbb{R}^{IJK},
\end{aligned} \tag{B.1}$$

where $\overline{G}(\widehat{d}_{ijk})$ is as defined in (39) and λ_{1ij}, λ_{2ij}, λ_{3ijk}, and λ_{4ijk} are Lagrangian multipliers of constraints (30), (31), (32), and (33), respectively.

Proof. The proof of Theorem C follows from that of Theorem 1 of [8].

Variational inequality has a standard form.

Find $Z^* \in \mathbb{R}_+^{\sum_{i=1}^{I}(n_M^i \times n_D^i + n_C^i)n_R J + 3IJ + IJK} \bigcup \mathbb{R}^{IJK}$, such that

$$\begin{aligned}
&\left\langle F\left(Z^*\right), Z - Z^*\right\rangle \ge 0,\\
&\qquad \forall Z \in \mathbb{R}_+^{\sum_{i=1}^{I}(n_M^i \times n_D^i + n_C^i)n_R J + 3IJ + IJK} \bigcup \mathbb{R}^{IJK},
\end{aligned} \tag{B.2}$$

where $Z = (X, \lambda_1, \lambda_2, \lambda_3, \lambda_4)$, $F(Z) = (F_1(Z), F_2(Z), F_3(Z), F_4(Z), F_5(Z), F_6(Z), F_7(Z))$, $F_1(Z) \in \mathbb{R}^{\sum_{i=1}^{I}(n_M^i \times n_D^i)n_R J}$, $F_2(Z) \in \mathbb{R}^{\sum_{i=1}^{I} n_C^i n_R J}$, $F_3(Z) \in \mathbb{R}^{IJ}$, $F_4(Z) \in \mathbb{R}^{IJ}$, $F_5(Z) \in \mathbb{R}^{IJ}$, $F_6(Z) \in \mathbb{R}^{IJK}$, $F_7(Z) \in \mathbb{R}^{IJK}$, the component of $F_1(Z)$ corresponding to the variable x_{jp^f} ($\forall p^f \in P_k^i, \forall i, \forall j, \forall k$) is

$$\begin{aligned}
&\frac{\partial \widehat{c}_{jp^f}\left(x_{jP}\right)}{\partial x_{jp^f}} + \theta_{ijk}^{+}\Pr\left(\widehat{d}_{ijk} \le v_{ijk}\right)\\
&+ \theta_{ijk}^{-}\left(\Pr\left(\widehat{d}_{ijk} \le v_{ijk}\right) - 1\right)\\
&- \int_{v_{ijk}}^{+\infty} \overline{G}\left(\widehat{d}_{ijk}\right)\mathscr{F}_{ijk}\left(\widehat{d}_{ijk}\right)\mathrm{d}\widehat{d}_{ijk} + \overline{\rho} + \lambda_{2ij}\\
&- \lambda_{3ijk},
\end{aligned} \tag{B.3}$$

the component of $F_2(Z)$ corresponding to the variable y_{jp^r} ($\forall p^r \in \widehat{P}_k^i, \forall i, \forall j, \forall k$) is

$$\frac{\partial E\left[\widehat{c}_{jp^r}\left(y_{jp^r}\widehat{r}_{kji}\right)\right]}{\partial y_{jp^r}} - \lambda_{4ijk}, \tag{B.4}$$

the component of $F_3(Z)$ corresponding to the variable x_{ij}^{New} ($\forall i, \forall j$) is

$$\frac{\partial \overline{f}_{ij}\left(x_{ij}^{\mathrm{New}}\right)}{\partial x_{ij}^{\mathrm{New}}} + \lambda_{1ij} - \lambda_{2ij}, \tag{B.5}$$

the component of $F_4(Z)$ corresponding to the variable λ_{1ij} ($\forall i, \forall j$) is

$$\overline{x} - x_{ij}^{\mathrm{New}}, \tag{B.6}$$

the component of $F_5(Z)$ corresponding to the variable λ_{2ij} ($\forall i, \forall j$) is

$$x_{ij}^{\mathrm{New}} + E\left(x_{ij}^{\mathrm{Re}}\right) - \sum_{k=1}^{n_R}\sum_{p^f \in P_k^i} x_{jp^f}, \tag{B.7}$$

the component of $F_6(Z)$ corresponding to the variable λ_{3ijk} ($\forall i, \forall j, \forall k$) is

$$\sum_{p^f \in P_k^i} x_{jp^f} - E\left(r_{kji}\right), \tag{B.8}$$

and the component of $F_7(Z)$ corresponding to the variable λ_{4ijk} ($\forall i, \forall j, \forall k$) is

$$\sum_{p^r \in \widehat{P}_k^i} y_{jp^r} - 1. \tag{B.9}$$

□

C. Euler Algorithm

Euler algorithm is induced by the general iterative scheme of [11]. At iteration l of the Euler algorithm, one computes

$$Z^{l+1} = P_K\left(Z^l - \alpha^l F\left(Z^l\right)\right), \quad \text{(C.1)}$$

where P_K is the projection on the set K and F is the function corresponding to the variational inequality in Appendix B.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the General Research Fund (Project no. Poly U 5405/11 H) of the Research Grants Council, Hong Kong, and the Research committee of the Hong Kong Polytechnic University. It was also supported by the National Natural Science Foundation of China (no. 71371102) and the Natural Science Foundation of Shandong Province of China (ZR2013GQ007).

References

[1] A. Nagurney, J. Dong, and D. Zhang, "A supply chain network equilibrium model," *Transportation Research Part E*, vol. 38, no. 5, pp. 281–303, 2002.

[2] J. Dong, D. Zhang, and A. Nagurney, "A supply chain network equilibrium model with random demands," *European Journal of Operational Research*, vol. 156, no. 1, pp. 194–212, 2004.

[3] A. Nagurney and F. Toyasaki, "Reverse supply chain management and electronic waste recycling: a multitiered network equilibrium framework for e-cycling," *Transportation Research Part E: Logistics and Transportation Review*, vol. 41, no. 1, pp. 1–28, 2005.

[4] D. Hammond and P. Beullens, "Closed-loop supply chain network equilibrium under legislation," *European Journal of Operational Research*, vol. 183, no. 2, pp. 895–908, 2007.

[5] G.-F. Yang, Z.-P. Wang, and X.-Q. Li, "The optimization of the closed-loop supply chain network," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 1, pp. 16–28, 2009.

[6] Q. Qiang, K. Ke, T. Anderson, and J. Dong, "The closed-loop supply chain network with competition, distribution channel investment, and uncertainties," *Omega*, vol. 41, no. 2, pp. 186–194, 2013.

[7] G. M. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, pp. 747–756, 1976.

[8] A. H. Masoumi, M. Yu, and A. Nagurney, "A supply chain generalized network oligopoly model for pharmaceuticals under brand differentiation and perishability," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 4, pp. 762–780, 2012.

[9] M. Yu and A. Nagurney, "Competitive food supply chain networks with application to fresh produce," *European Journal of Operational Research*, vol. 224, no. 2, pp. 273–282, 2013.

[10] A. Nagurney and M. Yu, "Sustainable fashion supply chain management under oligopolistic competition and brand differentiation," *International Journal of Production Economics*, vol. 135, no. 2, pp. 532–540, 2012.

[11] P. Dupuis and A. Nagurney, "Dynamical systems and variational inequalities," *Annals of Operations Research*, vol. 44, no. 1–4, pp. 9–42, 1993.

[12] European Union, "Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE)," Official Journal of the European Union, L 37/34.

[13] J.-B. Sheu and W. K. Talley, "Green supply chain management: trends, challenges, and solutions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 6, pp. 791–792, 2011.

[14] S. Seuring, "A review of modeling approaches for sustainable supply chain management," *Decision Support Systems*, vol. 54, no. 4, pp. 1513–1520, 2013.

[15] Y. Zhou, C. K. Chan, K. H. Wong, and Y. C. E. Lee, "Closed-loop supply chain network under oligopolistic competition with multiproducts, uncertain demands, and returns," *Mathematical Problems in Engineering*, vol. 2014, Article ID 912914, 15 pages, 2014.

[16] B.-S. He, Y. Xu, and X.-M. Yuan, "A logarithmic-quadratic proximal prediction-correction method for structured monotone variational inequalities," *Computational Optimization and Applications*, vol. 35, no. 1, pp. 19–46, 2006.

[17] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[18] J. Nash, "Equilibrium points in *n*-person games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, pp. 48–49, 1950.

[19] J. F. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, pp. 286–298, 1951.

[20] H. Stackelberg, "Probleme der unvollkommenen Konkurrenz," *Weltwirtschaftliches Archiv*, vol. 48, pp. 95–141, 1938.

[21] B. D. Liu, "Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms," *Computers & Mathematics with Applications*, vol. 36, no. 7, pp. 79–89, 1998.

[22] J. R. Kim, J. U. Lee, and J.-B. Jo, "Hierarchical spanning tree network design with nash genetic algorithm," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 1040–1052, 2009.

[23] Y. Yu and G. Q. Huang, "Nash game model for optimizing market strategies, configuration of platform products in a Vendor Managed Inventory (VMI) supply chain for a product family," *European Journal of Operational Research*, vol. 206, 2, pp. 361–373, 2010.

[24] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th IEEE International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39–43, Nagoya, Japan, October 1995.

[25] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[26] F. P. Goksal, I. Karaoglan, and F. Altiparmak, "A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery," *Computers & Industrial Engineering*, vol. 65, no. 1, pp. 39–53, 2013.

[27] R. S. Kadadevaramath, J. C. H. Chen, B. L. Shankar, and K. Rameshkumar, "Application of particle swarm intelligence algorithms in supply chain network architecture optimization,"

Expert Systems with Applications, vol. 39, no. 11, pp. 10160–10176, 2012.

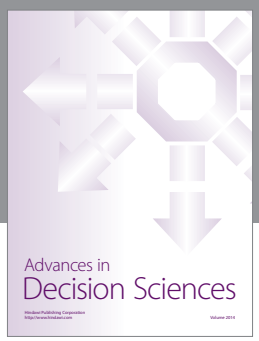
[28] K. Govindan, A. Jafarian, R. Khodaverdi, and K. Devika, "Two-echelon multiple-vehicle location-routing problem with time windows for optimization of sustainable supply chain network of perishable food," *International Journal of Production Economics*, vol. 152, pp. 9–28, 2014.

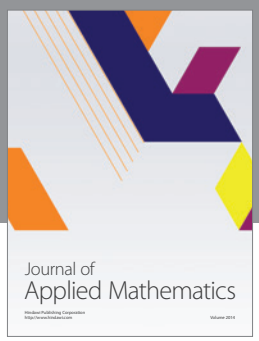
[29] K. Nishimura and J. Friedman, "Existence of Nash equilibrium in n person games without quasi-concavity," *International Economic Review*, vol. 22, no. 3, pp. 637–648, 1981.

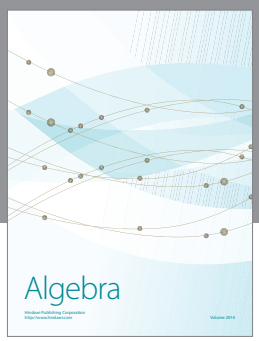
[30] M. Sefrioui and J. Periaux, "Nash genetic algorithms: examples and applications," in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, pp. 509–516, July 2000.

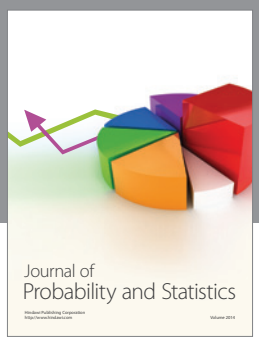
[31] A. Nagurney and D. Zhang, *Projected Dynamical Systems and Variational Inequalities with Applications*, Kluwer Academic, Boston, Mass, USA, 1996.





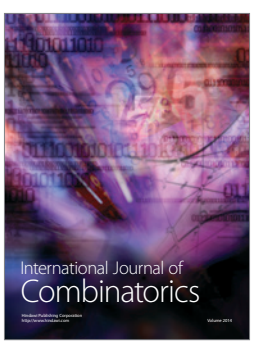


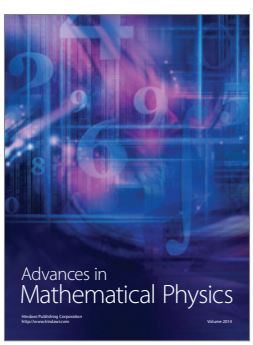


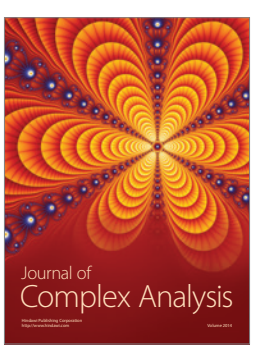


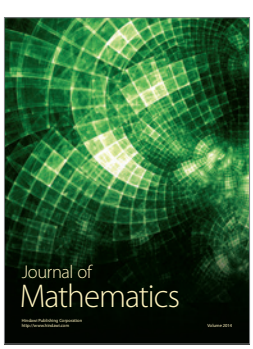


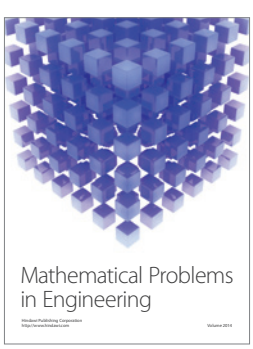


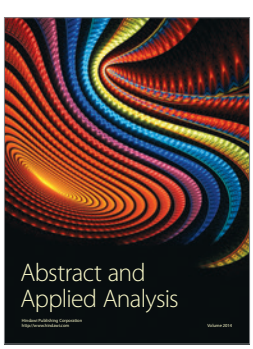


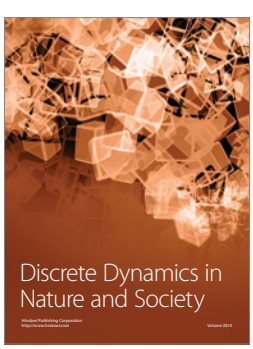


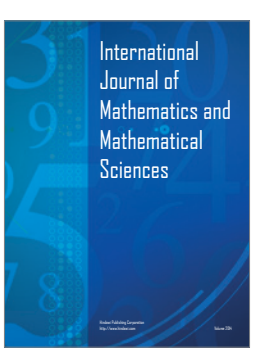




















Hindawi

Submit your manuscripts at
http://www.hindawi.com