



Article

# Solving Dual-Channel Supply Chain Pricing Strategy Problem with Multi-Level Programming Based on Improved Simplified Swarm Optimization

Wei-Chang Yeh \* , Zhenyao Liu , Yu-Cheng Yang and Shi-Yi Tan

Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan; liuzhenyao49@gmail.com (Z.L.); catherineyang@gapp.nthu.edu.tw (Y.-C.Y.); s108034871@m108.nthu.edu.tw (S.-Y.T.)

\* Correspondence: yeh@iee.org

**Abstract:** With the evolution of the Internet and the introduction of third-party platforms, a diversified supply chain has gradually emerged. In contrast to the traditional single sales channel, companies can also increase their revenue by selling through multiple channels, such as dual-channel sales: adding a sales channel for direct sales through online third-party platforms. However, due to the complexity of the supply chain structure, previous studies have rarely discussed and analyzed the capital-constrained dual-channel supply chain model, which is more relevant to the actual situation. To solve more complex and realistic supply chain decision problems, this paper uses the concept of game theory to describe the pricing negotiation procedures among the capital-constrained manufacturers and other parties in the dual-channel supply chain by applying the Stackelberg game theory to describe the supply chain structure as a hierarchical multi-level mathematical model to solve the optimal pricing strategy for different financing options to achieve the common benefit of the supply chain. In this study, we propose a Multi-level Improved Simplified Swarm Optimization (MLiSSO) method, which uses the improved, simplified swarm optimization (iSSO) for the Multi-level Programming Problem (MLPP). It is applied to this pricing strategy model of the supply chain and experiments with three related MLPPs in the past studies to verify the effectiveness of the method. The results show that the MLiSSO algorithm is effective, qualitative, and stable and can be used to solve the pricing strategy problem for supply chain models; furthermore, the algorithm can also be applied to other MLPPs.

**Keywords:** dual-channel supply chain; pricing strategy; Stackelberg game; multi-level programming; improved simplified swarm optimization



**Citation:** Yeh, W.-C.; Liu, Z.; Yang, Y.-C.; Tan, S.-Y. Solving Dual-Channel Supply Chain Pricing Strategy Problem with Multi-Level Programming Based on Improved Simplified Swarm Optimization. *Technologies* **2022**, *10*, 73. <https://doi.org/10.3390/technologies10030073>

Academic Editors: Manoj Gupta, Eugene Wong and Gwanggil Jeon

Received: 20 May 2022

Accepted: 9 June 2022

Published: 11 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Supply chain systems have been progressively diversifying besides conventional retailing manners. Nowadays, the increased competition and globalization of the market have become necessary for different individuals in the supply chain to cooperate to achieve mutual benefits. The competitions within the supply chain have catching researchers' attention [1,2].

Supply chain management (SCM) handles the entire production flow of a good or service, which is a network that moves the product along from the suppliers of raw materials to those organizations that deal directly with users. In addition, due to the invention and growth of the internet, the prosperity of third-party platforms—online retailing, has gradually increased; companies can engage an additional sales channel in direct selling their products to customers and create various ways of sales to increase income. The appearance of multi-channel supply chain management issues is due to the rapid growth of e-commerce, which has led some manufacturers to sell their products online and increase their sales channels to remain competitive and increase the accessibility of

their products. The increase in sales channels represents the complexity of their competition as well as coordination. The market demand is sensitive to the selling price set by the seller, so in a supply chain system, pricing strategy is a complex and tedious decision with numerous factors that affect it.

With all these buying and selling behaviors in business activities, many variables are considered by either the seller or the buyer. Each party is expected to achieve its desired benefit or goal in conducting the activity. Under this premise, the parties coordinate to reach a consensus and gain equilibrium through repeated communication for possible requests to achieve compromises. Therefore, replacing traditional corporate goals with overall value maximization through a holistic approach to the supply chain is a key issue for companies to consider nowadays. Decentralized decision-making occurs when there is a conflict between decision-makers. A hierarchical structure of decentralized decision-making should be carried out according to organizational departments. The objectives of the decision-makers are independent, and they are aimed at maximizing their own profits.

In reality, however, the situation is actually not that simple. There may be some conflict of interest because the asymmetry of the market position causes an asymmetry in the order of decision making for this German economist Heinrich Freiherr von Stackelberg proposed the Stackelberg model in 1934 to describe this situation of priority order decision making with leaders and followers [3]. In addition, the equilibrium point of this problem is determined through the solution of the Stackelberg game. The Nash equilibrium does not guarantee the best resolution for all decision-makers, the result may not be the most favorable situation, but it is an acceptable outcome for all parties. As a result, some studies use the method of multi-level programming (MLP), a mathematical model for solving decentralized decision-making problems, as an extension of the Stackelberg game to find the solution [4–6]. The key feature of this model is that the decision-makers have independent objective functions at each level of the hierarchy and control over the selection of decision variables.

The earliest proofs that a multi-level programming problem (MLPP) is an NP-hard problem are Ben & Blair (1990) [7] and Bard (1991) [8], and the bi-level programming problem (BLPP) they solved and proposed is derived from the MLPP problem, so the more complex MLPP also belongs to the NP-hard problem. Because of its limitations and complexity, it is more difficult to solve large-scale problems by mathematical planning methods. In recent years, researchers have adopted the more efficient meta-heuristic algorithm to obtain approximate solutions [6,9–12], which may not always lead to the best solution, but can handle more complex MLPP problems.

Nowadays, with increasingly complex supply chain relationships, companies need to be equipped with better decision models to manage their own goals. These problems can be solved by proposing suitable algorithms that can solve MLPP in a reasonable time and, at the same time, obtain an acceptable quality of the solution. The relationship between the supply chain network and the logistics distribution scheduling as regards applying swarm optimization algorithms proposed by some scholars, they harnessed the machine learning method, algorithms in the retailing environment in dynamic assessment to determine the users' trends and patterns and grasp customer attitudes and feelings [13–16]. The improved Simplified Swarm Optimization (iSSO) designed by Yeh [17] in 2015 is one of the evolutionary algorithms and stochastic optimization algorithms. It is characterized by the simplicity and efficiency of the iterative method. The algorithm demonstrates its excellent efficiency and generates high-quality solutions in solving most of the continuity problems.

Based on the above-mentioned excellent features, we propose an approach that uses iSSO to further optimize the pricing strategy by constructing an MLPP model that can effectively maximize the profit among all parties.

The purpose of this study is to investigate the use of an MLPP to solve the pricing problem of the supply chain, considering the financing decision options under dual sales channels where different options generate different interactions among the parties in the supply chain that will affect the pricing strategy. Therefore, this paper first uses the study

of Zhen (2020) [18] as the basis of the mathematical model of a dual-channel supply chain system to analyze the profit formula of each party under different financing strategies.

This study aims to develop a method to solve the NP-hard problem of MLPP and construct and investigate the multi-level supply chain system by exploring how the decision-makers in the supply chain system should decide on the best pricing strategy to maximize the profit. By considering various competing influences, the benefits of the supply chain system are maximized by making pricing decisions that satisfy all parties. Thus, the ideal situation is to prioritize the manufacturer's best interests and minimize the costs of all parties while achieving the best interests of the other parties.

The study objectives can be summarized as follows,

1. Build an MLPP model to obtain the equilibrium solution of pricing strategy in the dual-channel supply chain system.
2. Study and analyze the best decision for the manufacturer on finance strategy.
3. Apply the improved, simplified swarm optimization algorithm to multi-level programming problems.

The rest of this article is organized as follows. Section 2 lists the theoretical basis of the research. Section 3 describes the supply chain model, including proposed symbols, assumptions, and mathematical models. In Section 4, we introduce the research method, including the concept of MLPP and iSSO, and discuss the novelty and steps of the proposed MLiSSO. In Section 5, we analyzed the effects of the proposed method and described in detail the results of the above-mentioned supply chain pricing strategy. Finally, our conclusions are given in Section 6.

## 2. Literature Review

### 2.1. Dual-Channel Supply Chain

Nowadays, due to the rapid development of technology, various sales models have emerged in our society. In response to the purchasing habits of the new generation, the development of online retailing has become more and more prevalent.

In addition to traditional sales channels, upstream manufacturers in the supply chain are gradually developing channels to sell their products directly online. In this way, sales can be managed through a third-party platform without expanding your physical store or website and only require the costs associated with the platform, such as profit sharing and rent; the structure of the dual-supply chain is shown in Figure 1. Various studies on the supply chain phenomenon are also available in the market with service competition [19], channel selection [20], pricing strategies [21], and dual-channel supply chains [22].

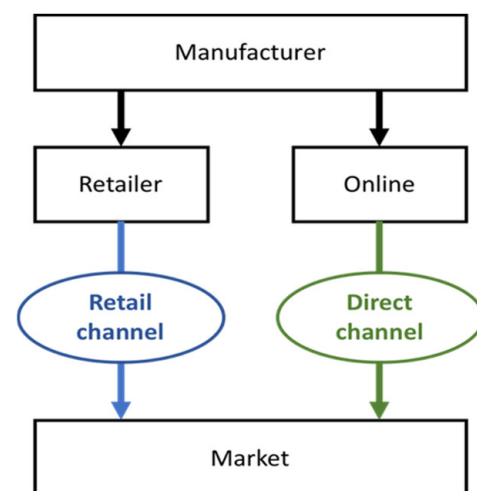


Figure 1. Structure of the dual-channel supply chain.

Channel competition holds an important role in dual-channel supply chain management. For example, Bernstein et al. (2009) address how competition between both retail and direct channels affects decisions made by manufacturers on supply chain structure [23]. Ryan et al. (2012) discussed the price competition and coordination in a dual-channel model [24]. Saha (2016) compared the performance of the manufacturer, the distributor, the retailer, and the entire supply chain in three different supply chain structures to prove that under some conditions that a dual-channel can outperform a single retail channel [25].

However, the studies on dual-channel supply chains mostly do not assume that firms are capital constrained; therefore, this study uses the financing strategy preferences of capital-constrained firms in the dual-channel supply chain proposed by Zhen in 2020 [18] while considering the financing strategies of third-party platforms in SCM. As a result, this study considers the operational management and financing strategy preferences of supply chain systems in the above-mentioned points.

Therefore, based on the model proposed by Zhen [18], this study examines the two aforementioned financing approaches for dual-channel competition and consumer considerations and presents the decision relationships between manufacturers and retailers with three different financing strategies. In addition, we compare the impact of cost and revenue on manufacturers, retailers, and lenders in the supply chain, maximizing profit and minimizing each cost to obtain the best pricing decision for the entire supply chain.

## 2.2. Supply Chain Finance

As the members of the supply chain gain benefits by selling their products while the market demand is sensitive to the selling price of the products, therefore, the pricing decision plays an important role in the profit optimization of the supply chain [6]. In considering changes in the correlation between product prices and market demand, companies can make profit analysis and pricing strategies efforts [26].

Lack of funding may be a hindrance to business development. There are two types of financing discussed in the literature on supply chain financing. One type of financing is external financing, which is defined as loans from institutions outside the supply chain, such as banks, third-party logistics, or other financial institutions. The other is internal financing, defined as loans from companies in the supply chain to their upstream or downstream companies, such as trade credit and buyer's credit [27].

Most research on internal financing has examined trade credit financing, with the majority of studies focusing on contract coordination and operational decisions under credit risk [28,29]. For external financing, the emphasis is on how the financing affects inventory or operations management and supply chain coordination [30,31]. Unlike the previous studies, Zhen (2020) focuses on the capital constraints of upstream firms under channel competition. This study is significant in examining how the capital constraints of upstream manufacturers affect the operation of dual channels [18].

When sales are not limited to traditional retail channels, to maximize the overall revenue is to develop a multi-channel pricing strategy, and it is cooperation and negotiation between each member in the supply chain system which can be considered as a game. For example, Matsui (2017) proposed that it would be appropriate for the manufacturer to release the direct selling price before the wholesale price is set. A sub-game perfect Nash equilibrium with the non-cooperative game of channel members is reached, and the manufacturer's profit is maximized [1]. The subgame perfect Nash equilibrium of the non-cooperative game with channel members is reached, and the manufacturer's profit is maximized.

## 2.3. Game Theory

Game theory is considered to be one of the most effective tools for dealing with these management problems. The well-known Prisoner's Dilemma and the Nash Equilibrium of modern noncooperation have become important concepts in game theory.

The strategic interactions between players are what game theory studies as the real-life dilemma that we often encounter. A strategic interaction means that the optimal choices of one player depend on other players' optimal choices and vice versa. Assume that each player is aware of the equilibrium strategies of the other players. In addition, none of the players gains any benefit by unilaterally changing its own strategy.

Increasingly, research papers are applying game theory to supply chain management [2,32,33]. Cachon and Zipkin [34] addressed the Nash equilibrium in a non-cooperative supply chain with one supplier and multiple retailers. Hennes and Arda [35] evaluated the efficiency of different types of contracts among industrial parties in a supply chain. Tian et al. [36] proposed a dynamic system model for green supply chain management based on evolutionary game theory, which applied game-theoretic methods to decision-making purposes.

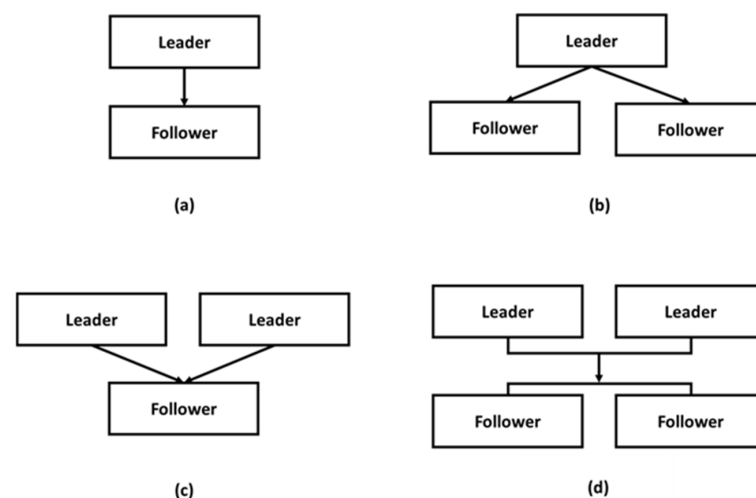
#### 2.4. Stackelberg Game

Several researchers have studied through game theory about coordination between manufacturers and retailers [37,38]. Each member attempts to maximize their own profit, a situation known as a non-cooperative game.

Since the market position asymmetry leads to the asymmetry of the decision sequence, there may be some conflict of interest. For this reason, the German economist Heinrich Freiherr von Stackelberg proposed the Stackelberg model in 1934 [3]. The Stackelberg model emphasizes the sequential relationship of decisions. In a game, the player who decides on a decision firstly is called the first player, while the other player is called the follower. When the first player decides his own strategy, he has already taken into account the possible decisions made by the followers in response to the first player's decision. After the first player's decision, the follower observes the first player's decision and thinks about the effect of the strategy on itself, and then makes the best response decision. The whole process means that both sides in the game make decisions based on the pursuit of their own best goals while considering the possible best response of the other side.

In the Stackelberg non-cooperative game, the dominant (leader) member controls the other members who act after the leader (followers). After estimating the reactions of other members, the leader will take the first decision [39]. The aforementioned hierarchical structure and the sequential nature of decision-making are consistent with the context set by Stackelberg's theory. Therefore, the main modeling framework in this paper applies the Stackelberg model in the tournament.

According to the number of participants, the Stackelberg game can be divided into four main structures, as shown in Figure 2.



**Figure 2.** Relationship between participants in the Stackelberg game. (a) represents a single leader and follower, (b) represents a single leader and multiple followers, (c) represents multiple leaders and single followers, and (d) represents multiple leaders with multiple followers.



### 2.5. Multi-Level Programming Problem

In this section, we first review the development of techniques for solving the Stackelberg game problem, then addresses the general formulation of the bi-level programming problem model and multi-level programming problem model.

The multilevel programming problem (MLPP) is an extension of the Stackelberg game [39]. It aims to solve decentralized planning involving multiple decision-makers, where each member seeks to maximize its own interests in a hierarchical organization. This mathematical model has been widely used in practical problems such as resource allocation [4], transportation network design [5], and pricing and lot-sizing [6].

When decision-makers conflict with each other, a decentralized decision-making problem arises. The decentralized decision-making should be by organization departments and form a kind of hierarchical structure. The decision makers' objectives are independent and may have some conflict of profit. Every decision-maker always wants to achieve a win-win situation called "dominant strategic equilibrium." However, in reality, the situation is actually not that simple. Nash equilibrium does not guarantee the best solution for every decision-maker, but it can get the best solution under the consideration of the entire group; therefore, multilevel programming (MLP) would be needed to find a solution. Zhou (2012) used game theory to determine the optimal pricing strategy to maximize the multilevel remanufacturing reverse supply chain [40]. Sadigh et al. (2012) found the optimal equilibrium of price, advertising spending, and production strategy in a bi-level programming approach [41].

The multilevel programming model has more advantages compared to the traditional single-level programming model. Its main benefits are (1) multilevel planning can be applied to analyze both different or even conflicting objectives in the decision process; (2) The multi-criteria approach of bi-level planning for decision-making can better reflect the actual problem; (3) The multi-level planning approach can denote the interactions between decision-makers.

In the current development of multi-level programming, several challenges emerge (1) Large scale—due to high dimensional decision variables for multi-level decision problems which become complex; (2) Uncertainty—with the uncertain information causing imprecise or unclear decision parameters and conditions for the decision subjects concerned; (3) Variety—with the possibility of the existence of multiple decision subjects with various relationships among them in each decision level. Yet, existing decision models or solution methods cannot fully and effectively handle these large-scale, uncertain and diverse multilevel decision problems [42].

There are two fundamental problems in solving MLPPs from a practical point of view. The first is the way to construct a multilevel decision model that describes the hierarchical decision process. Depending on the number of objectives involved, including dual objectives or multiple objectives; the number of members involved, including single leaders and followers or multiple leaders and followers; and the number of layers in the structure, including the bi-level programming problem (BLPP) or the MLPP. The BLPP is a special type of MLPP, and most of the research has been devoted to the BLPP study [9–11,41,43]. In addition, MLPP making that it is more complex than BLPP has been studied in depth in model building.

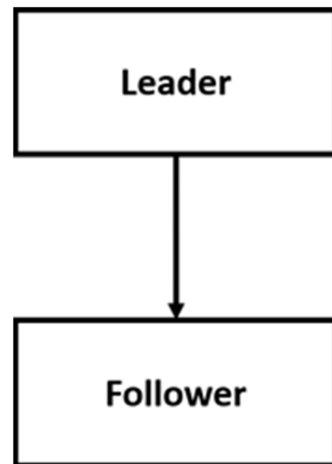
A further problem is how to identify methods for optimizing decisions. Several solving methods have been developed to solve these problems, broadly classified as exact algorithms and intelligent optimization algorithms. On the basis of the complexity of solving MLPP solutions, Ben & Blair (1990) proved through the well-known knapsack problem that BLPP is an NP-hard problem [7], and Bard (1991) even proved that BLPP is also an NP-hard problem through the search for locally optimal solutions [8]. This leads to exact algorithms that are time-consuming in solving nonlinear, discrete, and multi-optimal versions of large-scale problems that rely heavily on target function differentiability, which is not universally applicable [42].

At present, to obtain the optimal solution of MLPP, metaheuristic algorithms or innovative computations have been designed and widely used to solve BLPP and MLPP, i.e., Liu (1998) proposed a genetic algorithm for solving the Stackelberg-Nash equilibrium problem for generic MLPP with multiple followers [12], and Ma et al. (2013) using Particle Swarm Optimization (PSO) to solve BLPP on supply chain model [6]. Moreover, extending these algorithms to solve MLPPs is difficult and sometimes almost impossible. The main reason why solving MLPPs remains difficult is the lack of efficient algorithms; this is the biggest obstacle to the MLP problem [35,37].

Consequently, a more efficient algorithm has to be developed to solve large-scale BLPP and these algorithms can also be extended to solve MLPP. Thus, in this paper, we propose a multi-level improved, simplified swarm optimization (MLiSSO) method to solve the complex pricing strategy problem of a dual-channel supply chain involving multi-decision-makers, which are applied with a multi-level structure.

### 2.5.1. Bi-Level Programming Problem

A special case of a multi-level programming problem (MLPP) with a two levels structure is the bi-level programming problem (BLPP) [44]. The general form of the BLPP structure is shown in Figure 3.



**Figure 3.** The general form of BLPP structure.

Assume that upper-level decision-makers are given control over  $X$ , and lower-level decision-makers are given control over  $Y$ . Thus, we have  $x \in X \subset R^P$ ,  $y \in Y \subset R^q$ , and  $F, f : R^P \times R^q \rightarrow R^1$ . The general BLPP can be formulated as follows:

$$\underset{x \in X}{\text{Min}} F(x, y) \text{ (Leader)} \quad (1)$$

$$\text{s.t. } G(x, y) \leq 0 \quad (2)$$

where  $y$ , for each  $x$  fixed, solves the problems Equations (3) and (4).

$$\underset{y \in Y}{\text{Min}} f(x, y) \text{ (Follower)} \quad (3)$$

$$\text{s.t. } g(x, y) \leq 0 \quad (4)$$

The leader is the upper-level decision-maker Equation (1), and the follower is the lower-level decision-maker Equation (3). Depending on the demands of the model,  $x$  and  $y$  may have some additional restrictions, such as integer restrictions or limits on upper and lower bounds.

Based on these, we have the following definitions [45]:

**Definition 1.1.**

1. The problem constraint region,

$$S = \{(x, y) \in X \times Y : G(x, y) \leq 0, g(x, y) \leq 0\} \quad (5)$$

2. The follower feasible set for each fixed  $x$ ,

$$S(x) = \{y \in Y : g(x, y) \leq 0\} \quad (6)$$

3. The follower rational reaction set,

$$P(x) = \{y \in Y : y \in \arg \min[f(x, y) : y \in S(x)]\}. \quad (7)$$

4. The problem inducible region (IR),

$$IR = \{(x, y) : (x, y) \in S, y \in P(x)\}. \quad (8)$$

5. The problem optimal solution set,

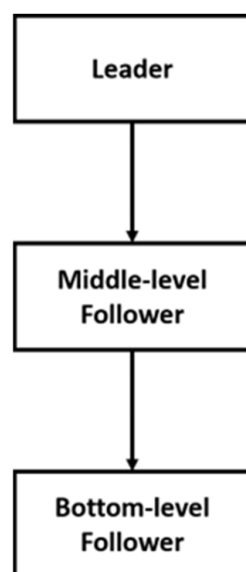
$$OS = \{(x, y) : (x, y) \in \operatorname{argmin}[F(x, y) : (x, y) \in IR]\} \quad (9)$$

**Definition 1.2.** This section may be divided into subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

**Definition 1.3.** For  $\forall (x, y) \in IR$ , if  $\exists_{(x^*, y^*)} \in IR, F(x^*, y^*) \leq F(x, y)$ , then  $(x^*, y^*)$  is an optimal solution of problem.

### 2.5.2. Multi-Level Programming Problem

In many applications, the problem of decentralized decision-making within a hierarchical system tends to include more than two levels, which are known as multi-level programming problems (MLPP). The general form of MLPP—tri-level structure is shown in Figure 4.



**Figure 4.** The general form of MLPP—tri-level structure.



For  $x \in X \subset R^P$ ,  $y \in Y \subset R^q$ ,  $z \in Z \subset R^r$ , The general tri-level decision problem presented by Faisca [41] is defined as follows:

$$\text{Min}_{x \in X} f_1(x, y, z) \text{ (Leader)} \quad (10)$$

$$\text{s.t. } g_1(x, y, z) \leq 0 \quad (11)$$

where  $(y, z)$ , for each  $x$  fixed, solves the problems Equations (12)–(15)

$$\text{Min}_{y \in Y} f_2(x, y, z) \text{ (Middle – level follower)} \quad (12)$$

$$\text{s.t. } g_2(x, y, z) \leq 0 \quad (13)$$

where  $z$ , for each  $(x, y)$  fixed, solves the problems Equations (14) and (15)

$$\text{Min}_{z \in Z} f_3(x, y, z) \text{ (Bottom – level follower)} \quad (14)$$

$$\text{s.t. } g_3(x, y, z) \leq 0 \quad (15)$$

where  $x, y, z$  are the decision variables of the leader, the middle-level follower, and the bottom-level follower, respectively;  $f_1, f_2, f_3 : R^P \times R^q \times R^r \rightarrow R$  are the objective functions of the three decision entities, respectively;  $g_i : R^P \times R^q \times R^r \rightarrow R^{k_i}$ ,  $i = 1, 2, 3$  are the constraint conditions of the three decision entities respectively.

Based on these, we have the following definitions [46]:

**Definition 2.1.**

6. The problem constraint region,

$$S = \{(x, y, z) \in X \times Y \times Z : g_i(x, y, z) \leq 0, i = 1, 2, 3\} \quad (16)$$

7. The middle-level follower feasible set for each fixed  $x$ ,

$$S(x) = \{(y, z) \in Y \times Z : z \in g_2(x, y, z) \leq 0, g_3(x, y, z) \leq 0\} \quad (17)$$

8. The bottom-level follower feasible set for each fixed  $(x, y)$ ,

$$S(x, y) = \{z \in Z : g_3(x, y, z) \leq 0\} \quad (18)$$

9. The bottom-level follower rational reaction set,

$$P(x, y) = \{z \in Z : z \in \text{argmin}[f_3(x, y, z) : z \in S(x, y)]\} \quad (19)$$

10. The middle-level follower rational reaction set,

$$P(x) = \{y, z \in Y \times Z : (y, z) \in \text{argmin}[f_2(x, y, z) : (y, z) \in S(x), z \in P(x, y)]\} \quad (20)$$

11. The problem inducible region,

$$IR = \{(x, y, z) : (x, y, z) \in S, (y, z) \in P(x)\} \quad (21)$$

12. The problem optimal solution set,

$$OS = \{(x, y, z) : (x, y, z) \in \text{argmin}[f_1(x, y, z) : (x, y, z) \in IR]\} \quad (22)$$

To develop an efficient algorithm to solve a 3 levels decision problem, it is necessary to explore the geometry of the solution space and the associated theoretical properties. The following assumptions are usually made to ensure that the problem is well formulated in terms of the existence of a solution.

**Assumption 2.1.**  $f_1, f_2, f_3, g_1, g_2, g_3$  are continuous functions, whereas  $f_2, f_3, g_1, g_2, g_3$  are continuously differentiable.

**Assumption 2.2.**  $f_3$  is strictly convex in  $z$  for  $z \in S(x, y)$  where  $S(x, y)$  is a compact convex set, while  $f_2$  is strictly convex in  $(y, z)$  for  $(y, z) \in S(x)$  where  $S(x)$  is a compact convex set.

**Assumption 2.3.**  $f_1$  is continuous convex in  $x, y$ , and  $z$ .

Under Assumptions 2.1 and 2.2, the rational reaction sets of the bottom-level follower and the middle-level follower  $P(x, y)$  and  $P(x)$  are point-to-point maps and closed, which implies that  $IR$  is compact. Thus, under Assumption 2.3, solving the tri-level decision problem is equivalent to optimizing the leader's continuous function  $f_1$  over the compact set  $IR$ . It is well known that the solution to such a problem is guaranteed to exist.

It is noticeable that if the bottom-level follower's problem is a convex parametric programming problem that satisfies the Karush–Kuhn–Tucker Conditions (KKT) for each fixed  $(x, y)$  [45,47], the bottom-level follower's problem is equivalent to the following KKT Equations (23)–(26):

$$\nabla_z L(x, y, z, u) = \nabla_z f_3(x, y, z) + u \nabla_z g_3(x, y, z), \quad (23)$$

$$u g_3(x, y, z) = 0, \quad (24)$$

$$g_3(x, y, z) \leq 0, \quad (25)$$

$$u \geq 0 \quad (26)$$

where  $\nabla_z f_3(x, y, z) + u \nabla_z g_3(x, y, z)$  is the Lagrangian function of the bottom-level follower,  $\nabla_z L(x, y, z, u)$  denotes the gradient of the function, for  $z$  and  $u$  is the vector of Lagrangian multipliers. A necessary and sufficient condition that  $(y, z) \in P(x)$  is that the row vector  $u$  exists such that  $(x, y, z, u)$  satisfies the KKT Equations (23)–(26).

On this basis, by replacing the bottom-level follower problem with the KKT Equations (23)–(26), the tri-level programming problem can be transformed into a bi-level programming problem. The converted equation is shown below:

$$\underset{x}{\text{Min}} f_1(x, y, z) \text{ (Leader)} \quad (11) \quad (27)$$

where  $(y, z)$ , for each  $x$  fixed, solves the problems Equations (22)–(25)

$$\underset{y, z, u}{\text{Min}} f_2(x, y, z) \text{ (Follower)} \quad (12) \quad (28)$$

$$\nabla_z f_3(x, y, z) + u \nabla_z g_3(x, y, z) = 0 \text{ (24)–(26)} \quad (29)$$

In this research, the proposed MLiSSO algorithm is extended to solve a multi-level supply chain pricing problem to find a solution  $(x, y, z)$  based on Equations (11), (12) and (27)–(29).

### 2.6. Improved Simplified Swarm Optimization (iSSO)

In this study, because of the NP-hard nature of the multi-level model, we propose a solution procedure based on a novel, convenient and efficient heuristic algorithm called improved Simplified Swarm Optimization (iSSO) [17], which is based on the Simplified Swarm Optimization (SSO) [48] that can perform a full domain search over a large feasible solution space and enhance the solution quality of the algorithm during the search process.

In 2009, Yeh designed the Simplified Swarm Optimization (SSO) [43] to overcome the shortcomings of PSO proposed by Kennedy and Eberhart [49], which was developed based on human observation of birds foraging behavior and a little weak for discrete problems. The targeting principle was used to update variables quickly, which only uses one random number, two multiplications, and one comparison after  $c_w$ ,  $c_p$ , and  $c_g$  are given in SSO. According to the results of Yeh [50,51], SSO is more efficient in converging to high-quality solution spaces in some problems.

The update mechanism of SSO is very simple, efficient and flexible [48,50–56], and can be presented as a stepwise-function update:

$$x_{ij}^{t+1} = \begin{cases} g_j, & \text{if } \rho_{[0,1]} \in [0, C_g) \\ p_{ij}, & \text{if } \rho_{[0,1]} \in [C_g, C_p) \\ x_{ij}^t, & \text{if } \rho_{[0,1]} \in [C_p, C_w) \\ x, & \text{if } \rho_{[0,1]} \in [C_w, 1) \end{cases} \quad (30)$$

All variables need to be updated in traditional SSO (called all-variable update),  $i = 1, 2, \dots, Nsol$ ,  $j = 1, 2, \dots, Nvar$ ,  $t = 0, 1, 2, \dots, Ngen - 1$ . Let  $X_i^t = \{x_{i1}^t, x_{i2}^t, \dots, x_{iNvar}^t\}$  represent the  $i$ th solution in the  $t$  generation, and in the formula of Equation (30),  $x_{ij}^t$  is expressed as the  $j$ th variable in  $X_i^t$ ;  $Nvar$  represents the number of variables;  $c_w$ ,  $c_p$ , and  $c_g$  are a preset constant;  $p_{ij}^t$  is the best solution in its evolutionary history;  $g_j$  is the  $j$ th variable of the best solution ever, and  $x$  is a random number between the lower bound and the upper bound of the  $j$ th variable.

Then to further improve the ability of SSO to solve continuous type problems, Yeh introduced the improved Simplified Swarm Optimization (iSSO) in 2015 [17]. A continuous version of SSO with a new update mechanism is proposed in this work to enhance the ability to solve continuous problems with traditional SSO. To date, iSSO has been successfully applied to many sequential problems, as shown in Yeh [57,58], with experimental results demonstrating its effectiveness in solving sequential problems and its ability to produce high-quality solutions. The update mechanism of iSSO is much simpler than the major soft computing technique-PSO (which must calculate both the velocity and position functions) [18,48,54–56].

The update mechanism of iSSO can be presented as follows:

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + r_{ij[-0.5,0.5]}^t \cdot u_j & \text{if } x_{ij}^t = g_i \text{ or } \rho_{ij[0,1]}^t \in [0, C_r = c_r) \\ g_j + r_{ij[-0.5,0.5]}^t \cdot u_j & \text{if } x_{ij}^t \neq g_i \text{ and } \rho_{ij[0,1]}^t \in [C_r, C_g = c_r + c_g) \\ x_{ij}^t + r_{ij[-0.5,0.5]}^t \cdot (x_{ij}^t - g_j) & \text{if } x_{ij}^t \neq g_i \text{ and } \rho_{ij[0,1]}^t \in [C_g, 1 = c_r + c_g + c_w) \end{cases} \quad (31)$$

$$u_j = \frac{x_j^{min} - x_j^{max}}{2 \cdot Nvar} \quad (32)$$

As defined in Equation (36),  $C_r = c_r$ ,  $C_g = c_r + c_g$ . In addition, in Equation (32),  $u_j$  is calculated with the variable's lower-bound  $x_j^{min}$ , the upper-bound  $x_j^{max}$ , and the numbers of variables. For each update, a random number  $\rho_{ij}^t$  that is uniformly distributed between  $[0, 1]$  is randomly generated first, and  $r_{ij}^t$  is a random number that is uniformly distributed between  $[-0.5, 0.5]$ . To compare  $\rho_{ij}^t$  with the three constants  $C_r$ ,  $C_g$ , and  $C_w$ , if  $0 < \rho_{ij}^t < C_r$ , the variable is updated according to the first term of Equation (31); if  $C_r < \rho_{ij}^t < C_g$ , the variable is updated according to the second term of Equation (31) to find the adjacent values of  $g$ . If  $C_g < \rho_{ij}^t < C_w$ , the variable will be updated according to the third term of Equation (36) to find a value between the interval from itself to  $g_j$ .

If the variable does not meet the upper and lower bound restrictions, the variable will be set to the nearest boundary value. If  $X_i^{t+1}$  does not outperform  $X_i^t$  in the target function, then  $X_i^{t+1} = X_i^t$  and will not be updated.

So far, only a few papers have studied dual-channel supply chains under capital constraints, which can be regarded as an MLPP. To solve these problems, we apply a continuous-type algorithm iSSO on MLPP to deal with these pricing strategy problems. The detailed algorithmic procedure will be presented and explained in Section 4.

### 3. Statement

#### 3.1. Model Description

To solve the optimal pricing strategy for the overall supply chain and to further illustrate the hierarchical and interactive relationships among the supply chain decisions, we use a multi-level programming problem to describe the master-slave decision structure of the proposed capital constraint dual-channel supply chain model by Zhen [18]. The Stackelberg game is applied to the model due to the aforementioned level structure of the supply chain system and the sequential relation of decision-making is consistent with the context set. As a result and according to the different financing strategies, the supply chain structure can be divided into two types: bi-level and tri-level planning models to present the decisions made by all members of the supply chain in pursuit of their own optimal goals while considering the optimal responses of each other. This chapter introduces the assumptions, notations, and mathematical models of the problem.

#### 3.2. Assumptions

All assumptions regarding the study are described below.

1. This study constructs a dual-channel supply chain model with three levels of the supply chain (manufacturer  $\rightarrow$  retailer  $\rightarrow$  customer) to profit maximization.
2. The manufacturer's initial capital is zero and must repay the entire capital liability.
3. The basic principle of profitability is that the price must be designed to meet the conditions of profitability for all parties.
4. In the model, neither the upstream manufacturer nor the downstream manufacturer considers the inventory problem. The upstream manufacturer ships as much product as it makes to the downstream retailer. The downstream manufacturer buys as much as it can and sells it all to the market.

#### 3.3. Notations

According to the article published by Zhen [18], the notation in Table 1 is used in the capital-constrained dual-channel supply chain model.

**Table 1.** Notations of the dual-channel supply chain model.

Type	Symbol	Description
parameter	$a$	The total potential market size.
	$\lambda$	The underlying market share of the retailer for the manufacturer is $(1 - \lambda)$ . $0 \leq \lambda \leq 1$ .
	$b$	Demand sensitivity to its selling/retail price. $0 < b \leq 1$ .
	$d$	The coefficient of cross – price sensitivity. $0 < d \leq 1$ .
	$c$	Product production cost.
	$\eta$	Revenue sharing of 3rd party platform. $0 < \eta \leq 1$ .
	$i$	Finance strategy for: $i = \begin{cases} B, & \text{bank finance strategy.} \\ T, & \text{3rd platform finance strategy.} \\ R, & \text{retailer finance strategy.} \end{cases}$

Table 1. Cont.

Type	Symbol	Description
variables	$w^i$	Wholesale price, for $i = \begin{cases} B \\ T \\ R \\ None \end{cases}$ , $w^i \geq 0$ .
	$P_R^i$	Retailer's retail channel retail price, with finance strategy for $i = B, T$ or $R$ . $P_R^i \geq 0$ .
	$P_M^i$	Manufacturer's direct channel selling price through 3rd party platform, with finance strategy for $i = B, T$ or $R$ . $P_M^i \geq 0$ .
	$q_R^i$	Retail channel demand, with finance strategy for $i = B, T$ or $R$ .
	$q_M^i$	Direct channel demand, with finance strategy for $i = B, T$ or $R$ .
	$r^i$	Revenue sharing rate, with finance strategy for $i = B, T$ or $R$ . $0 < r^i \leq 1$ .

In general, in each channel, the demand is mainly influenced by its own price; therefore, it is assumed that  $b > d$ . In addition, we also assume  $a > (b - d)(P_M + P_R)$  as the demand is not negative.

### 3.4. The Mathematical Model Description

A dual-channel supply chain with the aim of profit maximization is considered in this article, with a manufacturer to produce a unit product at cost  $c$ . The manufacturer has two sales channels in this market. One is the retail channel, where the manufacturer sells the product at wholesale price  $w$  to the retailer, which sells it at retail price  $P_R$  to the consumer, and this channel is also known as the traditional channel. The other channel is the direct sales channel. The manufacturer sells the products directly to the consumers at the selling price  $P_M$  through a third-party platform, also called the third-party platform channel. The structure of the supply chain model is shown in Figure 5; the solid black line indicates that the products are sold to retailers through wholesale; the dotted black line indicates that the products are sold directly through a third-party online platform, and the platform fee  $\eta$  is paid for the cooperation.

In addition, the demands in this market are variables, defined as manufacturer demand  $q_M$  and retailer demand  $q_R$ , respectively. It is assumed that the demand structure of this supply chain is a linear price dependence, which is widely used in the literature [59]. The demand functions are as follows,

$$q_R = \lambda a - bP_R + dP_M \quad (33)$$

$$q_M = (1 - \lambda)a - bP_M + dP_R \quad (34)$$

where  $b$  means that a unit of price reduction increases the demand by  $b$ , corresponding to marginal and switching customers. A large value of  $d$  corresponds to switching customers who are sensitive to differences between the selling price and the retail price. In other words, the degree of differentiation between direct and retail channels decreases as  $d$  increases. Thus,  $d$  captures the degree of competition between the two channels [60,61].

To find the optimal financing decision and related pricing outcome for this model, we try to maximize the profits for manufacturers, retailers, third-party platforms, and the bank. The objective function and constraints of the model are described in the next part.

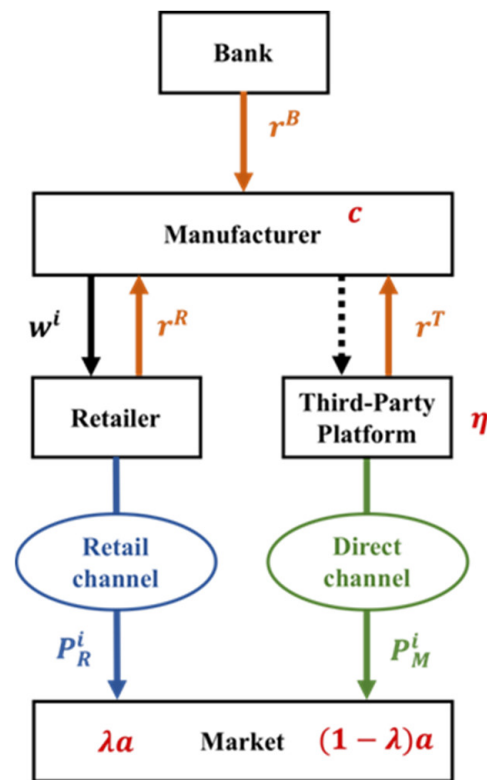


Figure 5. Supply chain model.

### 3.5. Model Construction

Based on the proposed dual-channel supply chain model proposed in [18] and due to the hierarchical decision relationship of the model, we formulate the supply chain model as an MLPP according to the literature review as follows:

(a) Retailer

$$\text{Max} = w^R q_R^R + (1 - \eta) P_M^R q_M^R - c(1 + r^R)(q_M^R + q_R^R) \quad (35)$$

$$\text{Max} = (P_R^i - w^R) q_R^R + r^R c(q_M^R + q_R^R) \quad (36)$$

(b) Bank

$$\text{Max } f_1 = w^B q_R^B + (1 - \eta) P_M^B q_M^B - c(1 + r^B)(q_M^B + q_R^B) \quad (37)$$

$$\text{Max } f_2 = (P_R^B - w^B) q_R^B \quad (38)$$

$$\text{Max } f_3 = r^B c(q_M^B + q_R^B) \quad (39)$$

(c) 3rd Party Platform

$$\text{Max } f_1 = w^T q_R^T + (1 - \eta) P_M^T q_M^T - c(1 + r^T)(q_M^T + q_R^T) \quad (40)$$

$$\text{Max } f_2 = (P_R^T - w^T) q_R^T \quad (41)$$

$$\text{Max } f_3 = \eta P_M^T q_M^T + r^T c(q_M^T + q_R^T). \quad (42)$$



(d) Constraints of all

$$\text{s.t. } a > (b - d)(P_M^i + P_R^i) \quad (43)$$

$$w^i q_R + (1 - \eta)P_M^i q_M^i > c(q_R^i + q_M^i)(1 + r^i) \quad (44)$$

$$P_R^i \geq w^i \geq c(1 + r^i) \quad (45)$$

$$P_M^i \geq c(1 + r^i) \quad (46)$$

$$P_M^i \geq w^i \quad (47)$$

Equation (35) means the manufacturer's maximum profit with retailer finance strategy, and Equation (36) means the retailer's maximum profit with retailer finance strategy.

Equation (37) means the manufacturer's maximum profit with bank finance strategy, Equation (43) means the retailer's maximum profit with bank finance strategy, and Equation (38) means the bank's maximum profit with bank finance strategy.

Equation (39) means the manufacturer's maximum profit with a third-party platform finance strategy, Equation (40) means the retailer's maximum profit with a third-party platform finance strategy, and Equation (41) means the third-party platform's maximum profit with a third-party platform finance strategy.

These three problems all share the same constraints for each layer of Equations (43)–(47). For Equation (43) means demand should not be negative. To fulfill Equation (44), manufacturers must set a selling price that ensures their revenue is greater than its cost. For Equations (45) and (46), we assume that channel prices must exceed marginal costs. For Equation (47), the retailer cannot purchase from the direct channel, so the selling price of the direct channel must not be lower than the wholesale price.

## 4. Methodology

### 4.1. Multi-Level Improved Simplified Swarm Optimization (MLiSSO)

In this study, we propose an MLISSO approach to apply iSSO to MLPP, including the following functional concepts, which are described in detail in the subsequent subsections.

The basic idea of MLPP can be explained as follows, for a strategy given by the leader, followers are assumed to react rationally. The resulting decisions of the leader and the followers can be considered as the "outcome" of the problem. If the leader chooses a different strategy, the outcome will change accordingly. The inducible region is then defined as the set of these outcomes for all the leader's strategies. Thus, the best outcomes that the leader can induce are the best results in the inducible region.

Therefore, in this paper, based on the main constraints of MLPP solving, we use the following methods to ensure the implementation of the above ideas and to avoid the problem of the solving process falling into the best solution of the region, as well as to ensure that it presents the characteristics of the MLPP model and the generation of feasible solutions.

#### 4.1.1. Improved Simplified Swarm Optimization (iSSO)

For the update mechanism iSSO we used in this paper, which is mentioned in the literature review [17], to maintain the diversity of the solution, we add the condition of partial best  $p_{ij}$  back into the formula, set the random range for turbulence to  $[-0.001, 0.001]$ , and make the value of  $u_j$  decrease as the number of generations increases. Equations (48) and (49) are the core formulas when iSSO evolves. Thus, there are four evolution scenarios for  $x_{ij}^{t+1}$ : oscillating near  $g_j$ , oscillating near  $P_{ij}$ , oscillating near the

original value, or evolving to the point between the original value and  $g_j$ . The evolution decision is determined by  $\rho_{ij}^t$ .

The modified formula is as follows:

$$x_{ij}^{t+1} = \begin{cases} g_j + \rho_{[-0.001,0.001]} \cdot u_j & \text{if } x_{ij}^t \neq g_j \text{ and } \rho_{[0,1]} \in [0, C_g) \\ p_{ij} + \rho_{[-0.001,0.001]} \cdot u_j & \text{if } x_{ij}^t \neq g_j \text{ and } \rho_{[0,1]} \in [C_g, C_p) \\ x_{ij}^t + \rho_{[-0.001,0.001]} \cdot u_j & \text{if } x_{ij}^t = g_j \text{ or } \rho_{[0,1]} \in [C_p, C_w) \\ x_{ij}^t + \rho_{[-0.001,0.001]} \cdot (x_{ij}^t - g_j) & \text{if } x_{ij}^t \neq g_j \text{ and } \rho_{[0,1]} \in [C_w, 1) \end{cases} \quad (48)$$

$$u_j = \frac{x_j^{\min} - x_j^{\max}}{2 \cdot N_{gen} N_{var}} \quad (49)$$

If any variable violates the boundary condition, it is set to the nearest boundary after using Equation (53). The steps are shown in detail in the following Section 4.1.7

#### 4.1.2. Fixed-Variables Local Search

Based on the hierarchical properties of the MLP problem, we introduce a local search method with fixed variables in MLISSO. In traditional SSO, the initial solution is generated randomly between the lower and upper bounds at the same time. When SSO updates the position, the solutions of all dimensions are changed simultaneously. In this study, only the solution of the decision variable of that level is changed when it is updated and then the local search is executed. The solutions of the remaining levels keep the original results. The calculation process is explained in detail in the following Section 4.1.7.

#### 4.1.3. Fitness Function

For the BLPP structure mentioned in the literature review Equations (1)–(4), the upper and lower levels of the programming problem are both standard constraints optimization problems that do not consider the information interaction between the leader and the follower. We treat the lower-level programming problem as a separate constraint optimization problem without losing the general approach to describe constraint processing techniques. In this case, the fitness of all particle updates can be calculated according to Equation (50); the fitness of the best solution is calculated and evaluated according to Equation (51):

$$fitness(x, y) = \begin{cases} f(x, y), & \text{if } y \in S(x) \\ F(x, y), & \text{if } y \in S(x)/S \end{cases} \quad (50)$$

where  $S(x)$  denotes the lower-level programming problem feasible set and  $S$  denotes the constraint region. The fitness value is calculated differently according to the updated level. For the upper-level update, we generate the value of  $F(x, y)$ , and for the lower-level update, we generate the value of  $f(x, y)$ , due to the level having different objective functions to obtain their optimal value.

$$fitness(x, y) = F(x, y), \text{ if } y \in S(x)/S \quad (51)$$

The MLISSO targets the leader's priority first. Therefore we use the higher-level objective functions for the best solution evaluation to ensure that we always put the leader's interest first in a multilevel programming situation. The calculation of all processes is explained in detail in the following Section 4.1.7.

#### 4.1.4. Constraint Handling

In this study, to make the solutions obtained by MLISSO conform to the problem constraints, we propose a simple but effective constraint method that ensures that the solutions generated during its operation are all conforming to the various constraints of the problem.

We use conditional constraints to enforce domain integrity by restricting the solutions generated after iterative updates to acceptable values that match the domain restrictions. A Boolean operator is used to establish the constraints, and when a solution is generated, it is determined whether it satisfies the constraints, and the result is returned. If the result meets the constraint, it is accepted as True and proceeds to the next step of the process; if it violates any of the constraints, it is rejected as False and generates a random set of solutions that meet the variable limitations (upper bound and lower bound), then redo the Boolean evaluation, repeating this step until it is accepted.

#### 4.1.5. Stopping Criteria

There are two major stopping criteria used:

1. The generation number.
2. The maximum iteration.

It will terminate the MLISSO algorithm after it has reached the maximum number of iterations or generations.

#### 4.1.6. Level Conversion

Based on the literature review that we referred to above, the problem of tri-level supply chains required to be solved in this study can be transformed into a bi-level programming problem through the use of Kuhn–Tucker conditions Equations (23)–(26) to convert the problem to the term as Equations (27)–(29) and Equations (11) and (12). The transformed supply chain equation is shown in Equations (52)–(57), Equations (11), Equations (13) and Equation (13) below:

$$\text{Max}_x F = w^i q_R^i + (1 - \eta) P_M^i q_M^i - c(1 + r^i)(q_M^i + q_R^i) \quad (11) \quad (52)$$

where  $(y, z, u)$ , for each  $x$  fixed, solves the problems Equations (59)–(66)

$$\text{Max}_{y,z,u} f = (P_R^i - w^i) q_R^i \quad (13) \quad (53)$$

$$0.4(q_M^i + q_R^i) - u_1 [0.4(q_M^i + q_R^i)] - 0.4u_2 - 0.4u_3 = 0 \quad (54)$$

$$u_1 [-w^i q_R^i - (1 - \eta) P_M^i q_M^i + 0.4(1 + r^i)(q_M^i + q_R^i)] = 0, \quad (55)$$

$$u_2 [-w^i + 0.4(1 + r^i)] = 0, \quad (56)$$

$$u_3 [-P_M^i + 0.4(1 + r^i)] = 0, \quad (15), (26) \quad (57)$$

#### 4.1.7. Steps of MLISSO for Solving MLPP

The steps of MLISSO to solve MLPP are described in this section. With one main program and two subprograms are included, which are based on iSSO algorithms. The details are explained as follows.

**Main Program: The best solution to solving**

- STEP 1-1** Maximum iteration  $T_{max}$ .
- STEP 1-2** Set  $T_{max} T = 0$ .
- STEP 1-3** Call Subprogram1 and generate the initial solution  $(X_i^T, Y_i^T)$ .
- STEP 1-4** Evaluate  $F(X_i^T, Y_i^T)$  and let  $(X^*, Y^*) = (X_i^T, Y_i^T)$ .
- STEP 1-5** Fixed  $Y_i^T$  to the upper-level programming model.
- STEP 1-6** Let  $T = T + 1$ .
- STEP 1-7** Call **Subprogram2** to generate  $X_i^T$ .
- STEP 1-8** Fixed the solution  $X_i^T$ .
- STEP 1-9** Call **Subprogram2** to generate  $Y_i^T$ .  
Fixed  $(X_i^T, Y_i^T)$  into the objective
- STEP 1-10** function to evaluate the value of the objective function.  
If  $F(X_i^T, Y_i^T) > F(X^*, Y^*)$ ,  $(X_i^T, Y_i^T)$  it is recorded as  $(X^*, Y^*)$ .
- STEP 1-11** Stopping criterion : if  $T \geq T_{max}$  go to **STEP 1-12**; otherwise, go to **STEP 1-6**.
- STEP 1-12** Output  $(X^*, Y^*)$  and the objective function value of the upper – level  $F(X^*, Y^*)$  and the lower – level  $f(X^*, Y^*)$ .

**Subprogram 1: Solution initialization**

- STEP 2-1** Initiate  $N_{sol}, N_{gen}, N_{var}, C_g, C_p$ , and  $C_w$ , and the upper and lower bounds of each variable.
- STEP 2-2** Set  $N_{gen} t = 0$  and  $i = 1$ , where  $i = 1, 2, \dots, N_{sol}$ .  
Generate  $(X_i^{Tt}, Y_i^{Tt})$ . Let  $P_{fi}^T = (X_i^{Tt}, Y_i^{Tt})$ , and calculate  $f(P_{fi}^T) = f(X_i^{Tt}, Y_i^{Tt})$  for
- STEP 2-3**  $i = 1, 2, \dots, N_{sol}$ . And find Gbest such that  $f(P_{fG}^T)$  is the best, and then let  $t = 1$  and  $i = 1$ .
- STEP 2-4** Generate  $\rho$  and calculate  $u_j$ .
- STEP 2-5** Generate  $r$  to update the  $X_i^{Tt}$  and  $Y_i^{Tt}$ , and calculate  $f(X_i^{Tt}, Y_i^{Tt})$ .
- STEP 2-6** If  $f(X_i^{Tt}, Y_i^{Tt}) > f(P_{fi}^T)$ , then  $P_{fi}^T = (X_i^{Tt}, Y_i^{Tt})$ ; Otherwise, go to **STEP 2-8**.
- STEP 2-7** If  $f(P_{fi}^T) > f(P_{fG}^T)$ , then  $P_{fG}^T = P_{fi}^T$ .
- STEP 2-8** If  $i \leq N_{sol}$  then  $i = i + 1$  and return to **STEP 2-4**.
- STEP 2-9** If  $t < N_{gen}$  then  $t = t + 1$  and  $i = 1$ , and return to **STEP 2-4**. Otherwise, go to **STEP 2-10**.
- STEP 2-10** Output  $P_{fG}^T = (X_i^T, Y_i^T)$ .

**Subprogram 2: Level updating solving**

- STEP 3-1** Initiate  $N_{sol}$  for both levels,  $N_{genl}$  (if updating with upper-level  $l = 1$ ; otherwise,  $l = 2$ ),  $N_{var}, C_g, C_p$ , and  $C_w$ , and the upper and lower bounds of each variable.
- STEP 3-2** Set  $N_{genl} t = 0$  and  $i = 1$ , where  $i = 1, 2, \dots, N_{sol}$ .  
Generate  $X_i^{Tt}$  or  $Y_i^{Tt}$ . Let  $P_{Fi}^T = (X_i^{Tt}, Y_i^T)$ ,  $P_{fi}^T = (X_i^T, Y_i^{Tt})$ ,  
and calculate  $F(P_{Fi}^T) = F(X_i^{Tt}, Y_i^T)$ ,  $f(P_{fi}^T) = f(X_i^T, Y_i^{Tt})$
- STEP 3-3** for  $i = 1, 2, \dots, N_{sol}$ . And find Gbest such that  $F(P_{FG}^T)$  or  $f(P_{fG}^T)$  is the best, and then let  $t = 1$  and  $i = 1$ .
- STEP 3-4** Generate  $\rho$  and calculate  $u_j$ .
- STEP 3-5** Generate  $r$  to update  $X_i^{Tt}$  and  $Y_i^{Tt}$  and calculate  $F(X_i^{Tt})$  and  $f(Y_i^{Tt})$ .  
For upper – level update, If  $F(X_i^{Tt}, Y_i^T) > F(P_{Fi}^T)$ , then  $P_{Fi}^T = (X_i^{Tt}, Y_i^T)$ ;
- STEP 3-6** for lower – level update, if  $f(X_i^T, Y_i^{Tt}) > f(P_{fi}^T)$ , then  $P_{fi}^T = (X_i^T, Y_i^{Tt})$ ;  
Otherwise, go to **STEP 3-8**.

- STEP 3-7** For upper – level update, if  $F(P_{Fi}^T) > F(P_{FG}^T)$ , then  $P_{FG}^T = P_{Fi}^T$ ;  
for lower – level update, if  $f(P_{fi}^T) > f(P_{fG}^T)$ , then  $P_{fG}^T = P_{fi}^T$ .
- STEP 3-8** If  $i \leq N_{sol}$ , then  $i = i + 1$  and return to **STEP 3-4**.
- STEP 3-9** If  $t < N_{genl}$  then  $t = t + 1$  and  $i = 1$ , and return to **STEP 3-4**. Otherwise, stop.
- STEP 3-10** Output  $P_{FG}^T$  or  $P_{fG}^T$ .

## 5. Data Analysis and Results

Section 5 is divided into two subsections. The first subsection presents a comparative analysis of the differences between the performance of the proposed algorithms in this thesis and other algorithms based on other papers. In the second section, the proposed methodology is applied to the actual supply chain problem, and the pricing decision results are analyzed.

### 5.1. Numerical Experiments

To test and demonstrate the above concept, three different types of numerical examples taken from the literature are presented. For comparison, in this study, 20 runs were performed (for problem 1 is 30 runs according to the compared algorithm results) for each problem. The standard deviation was calculated with the formula listed below in Table 2, where the standard deviation is based on the upper-level objective function.

**Table 2.** Comparison formula.

	Formula	Description
<b>Standard deviation (SD)</b>	$\sqrt{\frac{\sum_{i=1}^R (F_{Mi}^* - F_{MA}^*)^2}{R}}$ where $i = 1, 2, \dots, R$ . $R = 30$ in this paper.	$F_{Mi}^*$ = The optimal solution for MLiSSO in $i$ th run. $F_{MA}^*$ = The average of $R$ optimal solutions for MLiSSO.

#### 5.1.1. Experimental Datasets

In this study, we used four questions used in previous literature as the datasets for the validation tests; the functions are as shown in Tables 3–5. The dataset parameters were set according to the parameters used in the reference source data.

**Table 3.** Functions for problem 1.

No.	Problem Functions
<b>Problem 1 [62]</b>	$\text{Max } F = 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3,$ where $(y_1, y_2, y_3)$ solves, $\text{Max } f = -x_1 - 2x_2 - y_1 - y_2 - 2y_3$ s.t. $y_1 - y_2 - y_3 \geq -1$ $-2x_1 + y_1 - 2y_2 + 0.5y_3 \geq -1$ $-2x_2 - 2y_1 + y_2 + 0.5y_3 \geq -1$ $x_1, x_2, y_1, y_2, y_3 \geq 0$

**Table 4.** Functions for problem 2.

No.	Problem Functions
<b>Problem 2 [63]</b>	$\text{Min } F = -x_1^2 - 3x_2^2 - 4y_1 + y_2^2, \text{ where } (y_1, y_2) \text{ solves,}$ s.t. $x_1^2 + 2x_2 \leq 4$ $x_1, x_2 \geq 0$ $\text{Min } f = 2x_1^2 + y_1^2 - 5y_2$ s.t. $x_1^2 - 2x_1 + x_2^2 - 2y_1 + y_2 \geq -3$ $4x_2 + 3y_1 - 4y_2 \geq 4$ $y_1, y_2 \geq 0$

**Table 5.** Functions for problem 3.

No.	Problem Functions
<b>Problem 3 [6]</b>	$\text{Min } F = x^2 + (y - 10)^2, \text{ where } y \text{ solves,}$ s.t. $x + 2y - 6 \leq 0,$ $-x \leq 0$ $\text{Min } f = x^3 - 2y^3 + x - 2y - x^2$ s.t. $-x + 2y - 3 \leq 0,$ $-y \leq 0$

### 5.1.2. Experiments with Orthogonal Arrays

The experimental design of the MLiSSO setup was carried out using a two-factor, two-level full factorial design with four experimental combinations. Including the parameter pbest, and the modification of the u value according to the above mentioned in Section 4.

Each of the above-mentioned three experimental datasets was used to perform independent configuration experiments to identify the most suitable configurations, and Tables 6 and 7 show the configuration combinations.

**Table 6.** Factor level table.

Level\Factor	Parameter Cp	u Value Setting
1	Without	Constant
2	Add-in	Dynamic

**Table 7.** Full factorial design table.

Setting\Factor	Parameter Cp	u Value Setting
1	Without	Constant
2	Without	Dynamic
3	Add-in	Constant
4	Add-in	Dynamic

The following experiments were compiled using python 3.8 with the same basic parameters,  $Cg = 0.2$ ,  $Cp = 0.3$ ,  $Cw = 0.5$ , number of particles = 20, number of generations = 200 (for subprogram), and iterations= 500 (for main program).



Each experiment was run 20 times, and the results were evaluated and analyzed by using the leader's target function value results. Assuming that the samples conformed to the norm, an analysis of variance (ANOVA) with  $\alpha = 0.05$  was conducted to select the most suitable configuration.

#### Dataset: Problem 1

Table 8 shows that the  $p$ -values of factors A is smaller than  $\alpha = 0.05$ , so the factors did cause significant differences, and the  $p$ -value of factor B were greater than  $\alpha = 0.05$ , so the factors did not cause significant differences. However, factor B was more likely to cause differences than factor A, as shown in Table 9, which shows the mean value of 20 experiments for each of the four groups of experiments

Table 8. ANOVA table of Dataset problem 1.

Source	DF	SS	MS	F-Value	p-Value
A	1	23.401	23.4015	4.53	0.036
B	1	0.413	0.4126	0.08	0.778
Error	76	392.193	5.1604		
Total	79	427.444	0.000		
S = 2.27166		R – Sq = 8.25%		R – Sq (adj) = 4.63%	

Table 9. Response table of Dataset problem 1.

Level	A	B
1	27.5	27.0
2	28.0	26.0
Delta	0.5	1.0
Rank	2	1

From Figure 6, it can be concluded that the A factor has better performance at level 2 than level 1, and the B factors have better performance at level 1 than at level 2. But, according to the interaction plot, as shown in Figure 7, it indicates the existence of interaction, and we cannot tell if the configuration settings will have better performance by all set to level 2.

#### Dataset: Problem 2

Table 10 shows that the  $p$ -values of both factors A and B are smaller than  $\alpha = 0.05$ , so the factors did cause significant differences; furthermore, the factor B was more likely to cause differences than the factor A, as shown in Table 11, which is the mean values of 20 experiments for each of the four groups of experiments.

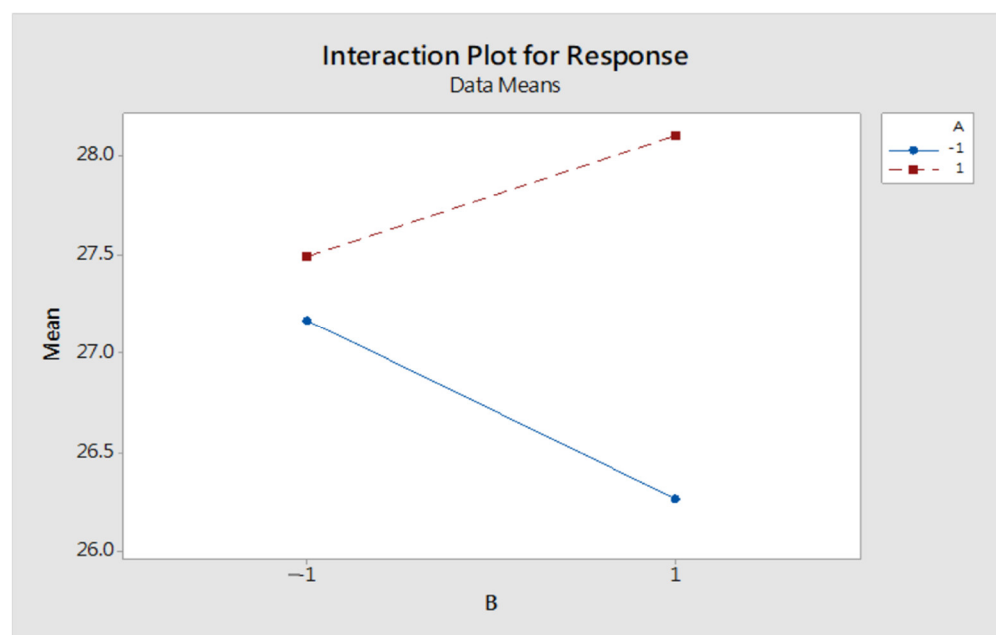
Table 10. ANOVA table of Dataset problem 2.

Source	DF	SS	MS	F-Value	p-Value
A	1	0.02050	0.020505	5.13	0.026
B	1	0.04019	0.040187	10.06	0.002
Error	76	0.30370	0.003996		
Total	79	0.45029	0.000000		
S = 0.0632139		R – Sq = 32.56%		R – Sq (adj) = 29.89%	

From Figure 8, it can be concluded that both A and B factors have better performance at level 2 than at level 1, which also has a significant difference in the performance. However, the interaction plot, as shown in Figure 9, it indicates the existence of interaction, and we cannot tell if they could have better performance by all set to level 2.

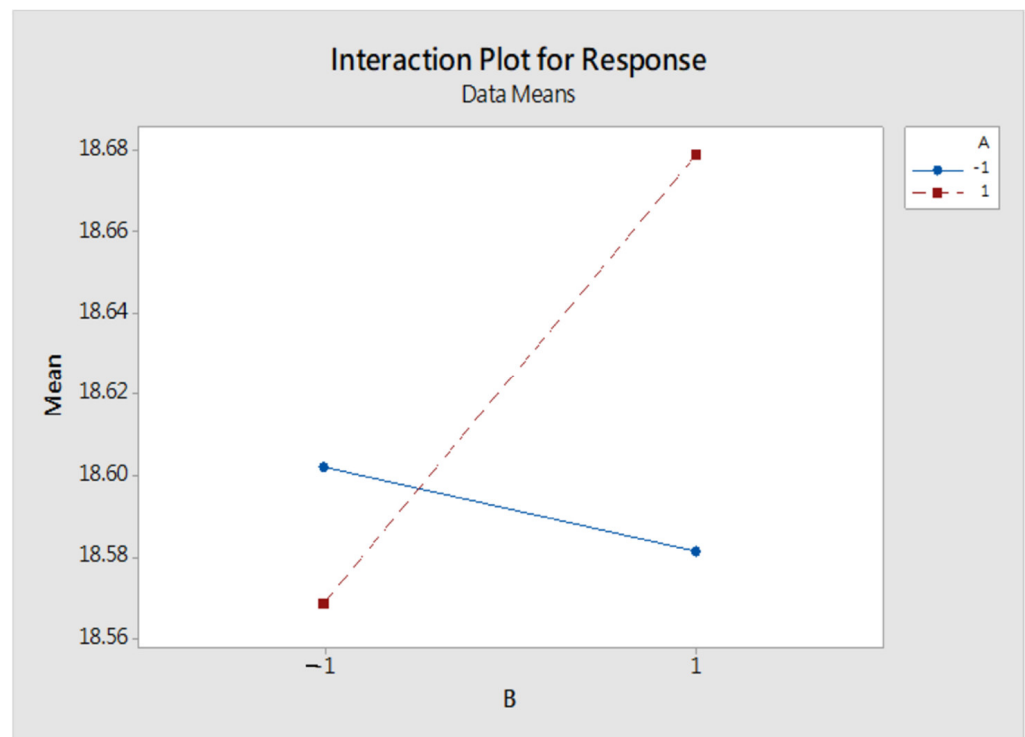
**Table 11.** Response table of Dataset problem 2.

Level	A	B
1	18.59	18.63
2	18.625	18.585
Delta	0.035	0.045
Rank	2	1

**Figure 6.** Main effects plot of problem 1. A represents factor A in Table 8, B represents factor B in Table 8.**Figure 7.** Interaction plot of problem 1. A represents factor A in Table 8, B represents factor B in Table 8.



**Figure 8.** Main effects plot of problem 2. A represents factor A in Table 8, B represents factor B in Table 8.



**Figure 9.** Interaction plot of problem 2. A represents factor A in Table 8, B represents factor B in Table 9.

### Dataset: Problem 3

Table 12 shows that the  $p$ -values of factors A and B were greater than  $\alpha = 0.05$ , so the factors did not cause significant differences; however, factor A was more likely to cause differences than factor B, as shown in Table 13, which is the mean values of 20 experiments for each of the four groups of experiments.

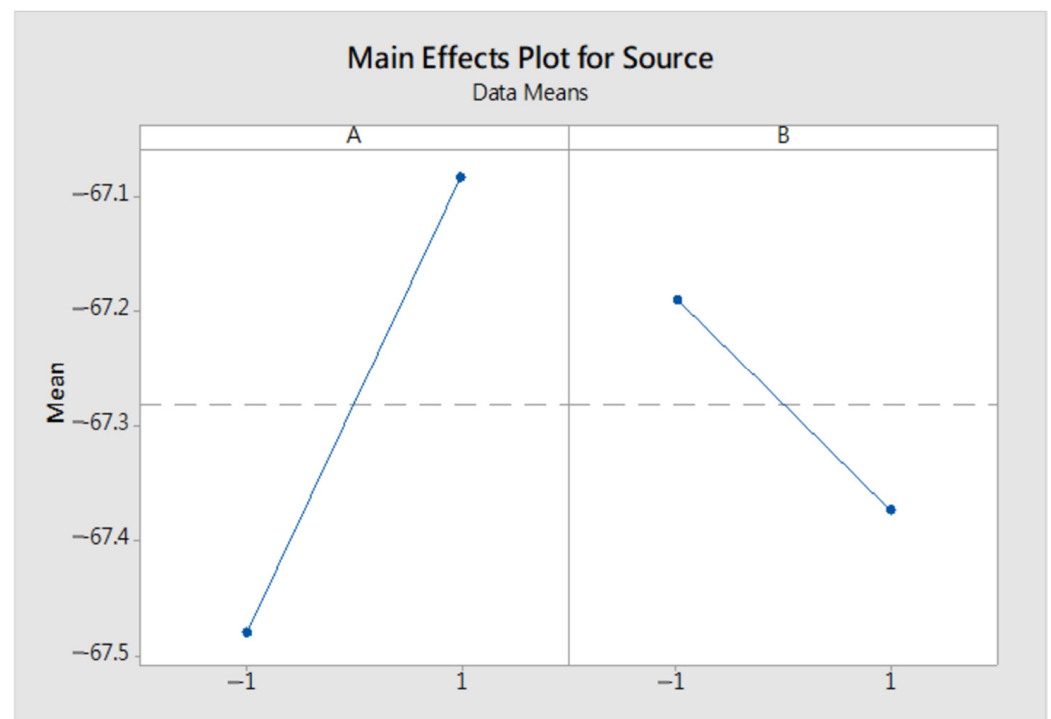
**Table 12.** ANOVA table of Dataset problem 3.

Source	DF	SS	MS	F-Value	p-Value
A	1	3.1344	3.1344	3.24	0.076
B	1	0.6697	0.6697	0.69	0.408
Error	76	73.4531	0.9665		
Total	79	78.8582	0.0000		
S = 0.983102		R-Sq = 6.85%		R-Sq (adj) = 3.18%	

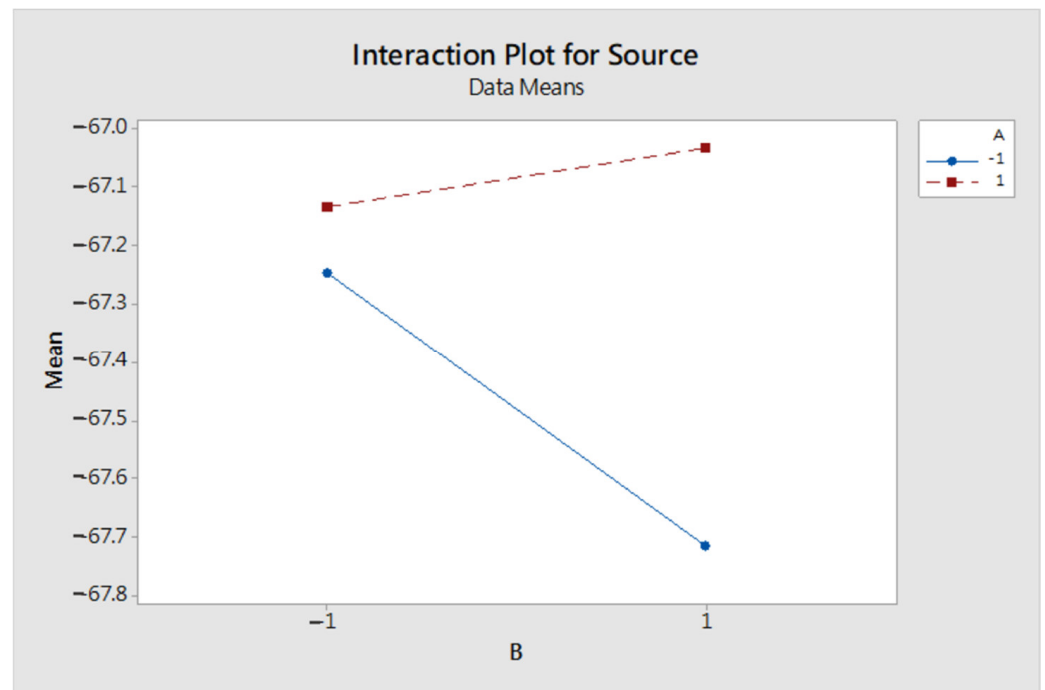
**Table 13.** Response table of Dataset problem 3.

Level	A	B
1	−67.5	−67.2
2	−67.1	−67.4
Delta	0.4	0.2
Rank	1	2

From Figure 10, it can be concluded that the A factor has better performance at level 2 than level 1, and the B factors have better performance at level 1 than at level 2. However, the interaction plot, as shown in Figure 11, it indicates the existence of interaction, and we cannot tell if the configuration settings will have better performance by all set to level 2.



**Figure 10.** Main effects plot of problem 3. A represents factor A in Table 8, B represents factor B in Table 8.



**Figure 11.** Interaction plot of problem 3. A represents factor A in Table 8, B represents factor B in Table 8.

#### Result Summary

The results of the experiments for the above four configurations are listed and discussed. As shown in Tables 14–16, it can be concluded that when both factors A and B are set to level 2 (setting 4), the results obtained for this configuration are superior to those in the other configurations in all three experiments. This setting 4 is also the MLiSSO configuration proposed in this study. Therefore, based on this result, the proposed MLiSSO will be used for other experiments and analyses in the following.

**Table 14.** Results of dataset problem 1.

Setting	$F_{avg}$	$F_{stdev}$	$F_{stdev}$	$F_{min}$
1	27.1654863	2.7327978	2.7327978	17.3713556
2	26.2656556	3.0503393	3.0503393	20.0050241
3	27.4909892	1.7662205	1.7662205	21.7917118
4	28.1035510	0.8657080	0.8657080	26.4103750

**Table 15.** Results of dataset problem 2.

Setting	$F_{avg}$	$F_{stdev}$	$F_{stdev}$	$F_{min}$
1	18.6020324	0.0710975	18.7231808	18.4959183
2	18.5813208	0.0616991	18.7136139	18.4883674
3	18.5685148	0.0636811	18.7202904	18.4420441
4	18.6788774	0.0553813	18.8334935	18.6205643

**Table 16.** Results of dataset problem 3.

Setting	$F_{avg}$	$F_{stdev}$	$F_{stdev}$	$F_{min}$
1	67.24701323	0.977282509	68.93931587	65.6916659
2	67.71292941	1.031981155	69.28061716	65.54469136
3	67.13405382	0.969031058	69.17334034	65.33748003
4	67.03412608	0.952295648	67.03412608	65.23260349

### 5.1.3. Comparison Experiment Results

In this study, we solved three sets of MLPP problems with different levels of complexity by using MLISSO and compared the results with those of algorithms proposed in other related literature.

#### Dataset: Problem 1

We constructed a linear BLPP with multiple leaders and followers from [62] as a numerical example to analyze more complex problems; the functions of problem 1 are listed in Table 3.

In this problem, the parameters setting of two different algorithms, GA [64] and PSO, are given in [11], then we used the trial-and-error method for the setting of MLISSO, and summarized in Table 17. In addition, as mentioned above, the number of iterations of MLISSO is indicated by generation(update) and iteration(main), while other algorithms used for comparison are indicated by iteration if not specified. The best optimal solution, mean, and standard deviation values of the solutions for 30 runs are presented in Tables 18 and 19. Figure 12 shows the convergence of the optimal solution target function value  $F(X, Y)$  in MLISSO.

**Table 17.** Parameters setup for problem 1.

GA [64]	PSO [11]	MLiSSO
opulation: 20, Crossover rate: 0.9, Mutation rate: 0.1, Iterations: N/A	Population: 20, Vmax: 10, Inertial weight: 1.2–0.2, Iterations: 150	Population: 20, Cg: 0.3, Cp: 0.6, Cw: 0.8, Generations:100/150, Iterations: 500/150

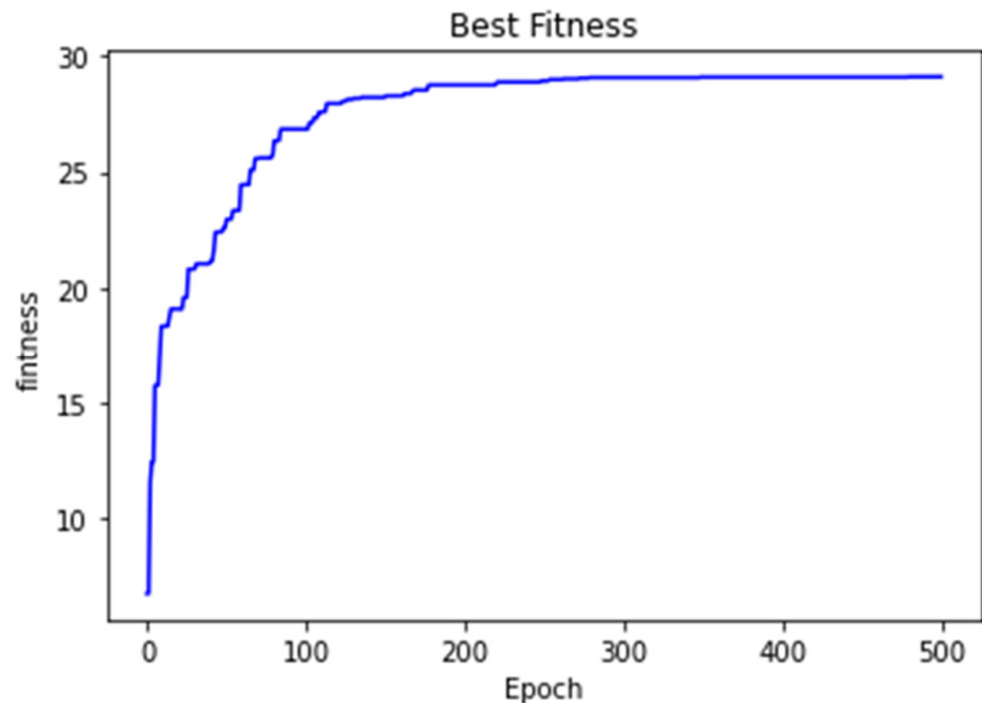
**Table 18.** Best results of problem 1.

	GA	PSO	MLiSSO (500)	MLiSSO (As Literature/150)
$x_1$	0.000	0.0004	0.0002	0.0266
$x_2$	0.898	0.8996	0.8991	0.0205
$y_1$	0.000	0.0000	0.0000	0.7969
$y_2$	0.599	0.5995	0.5993	0.7944
$y_3$	0.399	0.3993	0.3986	0.1503
$F$	29.1480	29.1788	29.6631	29.4853
$f$	−3.193	−3.1977	−3.1948	−1.9594
Runtime(s)	N/A	N/A	35	32



**Table 19.** Average results & SD of problem 1.

	GA	PSO	MLiSSO (500)	MLiSSO (As Literature/150)
$x_1$	0.15705	0.02192	0.00078	0.01579
$x_2$	0.86495	0.86693	0.89607	0.18669
$y_1$	0.00000	0.00000	0.00000	0.41225
$y_2$	0.47192	0.56335	0.59701	0.66371
$y_3$	0.51592	0.34108	0.39351	0.19149
$F$	21.52948	24.81256	29.04494	26.53842
$f$	-3.39072	-3.1977	-3.17696	-1.84811
$F$ stdev	3.14432	1.55374	0.10689	2.23245
Runtime(s)	N/A	N/A	45	35

**Figure 12.** Convergence curve when iteration = 500 of problem 1.

Tables 18 and 19 indicate that MLISSO has the smallest standard deviation according to the objective value priority of the leader in the case of linear bi-level decision-making with multiple leaders and multiple followers. It returns better results than the GA, just after the results of PSO with a difference of 0.0157 in terms of the best result. In addition, the average solutions of MLISSO return significantly better than the solutions of other algorithms in the average result, and the standard deviation of the MLISSO method is lower than that of other algorithms. This indicates that MLISSO has higher stability and provides better solution quality for solving complex problems.

In the study of Kuo & Huang [11], for the initial solution, they adopt the float coding method to generate the random numbers for the upper-level variables and program for variables in the lower level. Then, update the velocity and position for every particle at once. To compare with the results, we use the same structure and iteration = 150 to generate the results, which are listed in the right column (as literature) in Tables 18 and 19.

The results show that if we only use the proposed modified iSSO with the same kind of structure, the best result and the average result are both superior to the other two

methods, and the average result is just after the original MLISSO. However, the purpose of MLISSO is for general use on other types of MLPP when this method is only used on linear programming problems mentioned in the literature, so we use the following non-linear problems to emphasize the commonality of MLISSO.

#### Dataset: Problem 2

For the example of nonlinear BLPP, which was constructed from [63], the functions of problem 2 are listed in Table 4. In this problem, the two different algorithms, evolutionary algorithm (EA) and PSO-CST are given in [65,66], and the setup of the parameters is listed in Table 20. In addition, as mentioned above, the number of iterations of MLISSO is indicated by generation (update) and iteration (main), while other algorithms used for comparison are indicated by iteration if not specified. The best optimal solution, mean, and standard deviation values of the solutions for 20 runs are shown in Table 21, and because the average results are not given in the literature [65,66], we only list the result of MLISSO. Figure 13 shows the convergence of the optimal solution target function value  $F(X, Y)$  in MLISSO.

**Table 20.** Parameters setup for problem 2.

EA [65]	PSO-CST [66]	MLISSO
Population: 30, Crossover rate: 0.8, Mutation rate: 0.2, Iterations: 100	Population : 45, Numbers of particles : $m = 40$ (first update), $n = 5$ (CST particles), $V_{max} = 2$ , $c_1 = c_2 = 2$ , Iterations: 8	Population: 20, Cg: 0.2, Cp: 0.3, Cw: 0.5, Generations: 100, Iterations: 100

**Table 21.** Best results of problem 2.

	EA [65]	PSO-CST [66]	MLISSO
$x_1$	0.00000044	0.3844	0.0115
$x_2$	2	1.6124	1.9765
$y_1$	1.875	1.8690	1.8466
$y_2$	0.9063	0.8041	0.7988
$F$	-12.68	-14.7772	-18.4633
$f$	-1.016	-0.2316	-6.1174
$F$ stdev	N/A	N/A	0.1396
$F$ avg	N/A	N/A	-18.4566
Runtime(s)	N/A	N/A	5

In the case of this problem, we conclude from Table 21 that MLISSO outperforms the other algorithms in terms of the objective priority of the leaders and the standard deviation of the solutions obtained is only 0.1396, which means that its solutions remain fairly stable. Thus, MLISSO shows better performance than these two algorithms for the nonlinear BLPP. This result also implies that the proposed MLISSO is suitable for solving nonlinear multi-player BLPP.

#### Dataset: Problem 3

We constructed a nonlinear BLPP with a single leader and follower from [6] as a numerical example to analyze more complex problems(cube); the functions of problem 1 are listed in Table 5.

In this problem, the two different algorithms, HPSOBLP and IBPSO, are given in [6,67], with parameter settings summarized in Table 22. In addition, as mentioned above, the number of iterations of MLISSO is indicated by generation (update) and iteration (main), while other algorithms used for comparison are indicated by iteration if not specified. The best optimal solution, mean, and standard deviation values of the solutions for 20 runs are

presented in Tables 23 and 24. Figure 14 shows the convergence of the optimal solution target function value  $F(X,Y)$  in MLISSO.

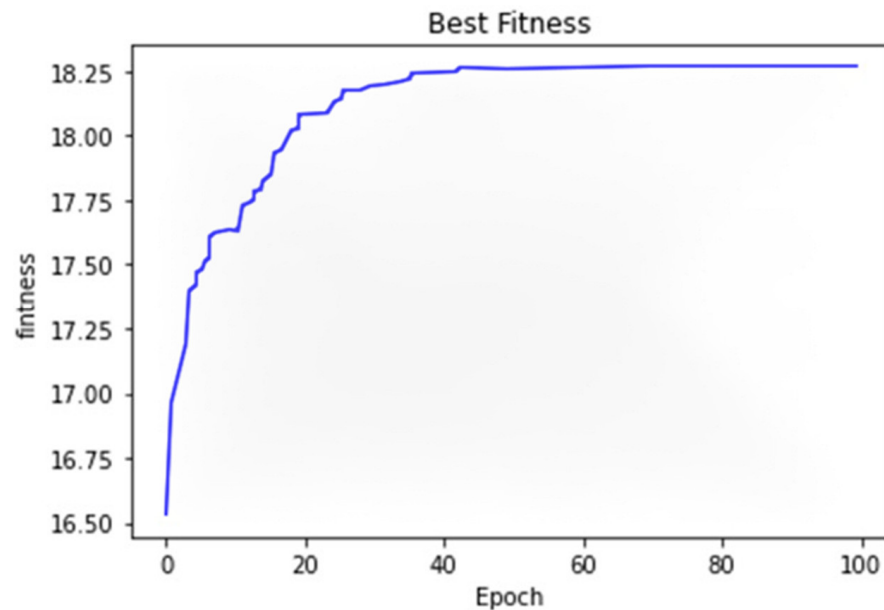


Figure 13. Convergence curve when iteration = 100 of problem 2.

Table 22. Parameters setup for problem 3.

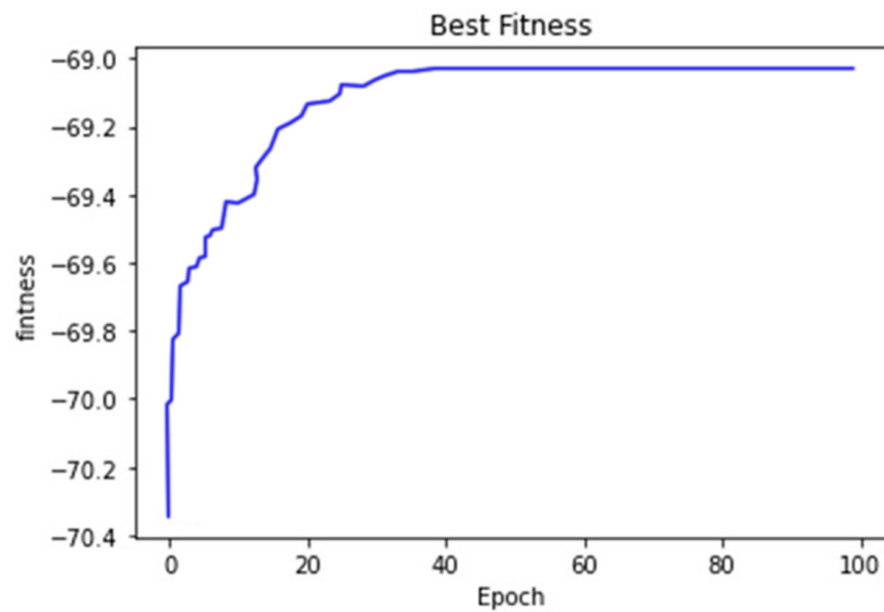
HPSOBLP [67]	IBPSO [6]	MLiSSO
Population : $N_{max} = 20, 40$ , $c_1 = c_2 = 2$ , $V_{max} = \text{bounds}$ , $w = \text{decrease}$ linearly from 1.2 to 0.1, Iterations: 120, 30	Population : $N_1 = N_2 = 20$ , $V_{max} = 10$ , $c_1 = c_2 = 2.5$ , Iteration : $T_1 = T_2 = 100$	Population: 20, Cg: 0.2, Cp: 0.3, Cw: 0.5, Generations: 100, Iterations: 100

Table 23. Best results of problem 3.

	HPSOBLP	IBPSO	MLiSSO
$x$	N/A	0.4960	1.0186
$y$	N/A	1.7356	1.9753
$F$	88.77571	68.5459	65.8663
$f$	-0.7698	-13.5561	-18.9133
Runtime(s)	N/A	N/A	5

Table 24. Average results & SD of problem 3.

	HPSOBLP	IBPSO	MLiSSO
$x$	N/A	1.1985	1.1036
$y$	N/A	1.7791	1.8756
$F$	88.7835	69.0192	67.4949
$f$	N/A	-13.3375	-15.8649
$F \text{ stdev}$	0.0016	N/A	1.0366
Runtime(s)	N/A	N/A	5



**Figure 14.** Convergence curve when iteration = 100 for problem 3.

Tables 23 and 24 indicate that MLiSSO outperforms the other algorithms in terms of the objective priority of the leaders, which means that MLiSSO has a better performance than these two algorithms. In addition, also the average results show that the solution can be obtained with better quality in several independent experiments.

### 5.2. Model Evaluation

Based on the aforementioned experimental results, it can be stated that MLiSSO can be used to solve the MLPP with a higher stability quality of optimal solution results. This section further verifies the practicality of the MLiSSO on supply chain problems by using the supply chain model with three different financing strategies Equations (40)–(47) and constraints of them Equations (48)–(52). The parameters setup of the supply chain model is, according to Zhen [18], listed in Table 25, and the parameter setup of MLiSSO is listed in Table 26. The corresponding solutions of the three financing strategy models are shown in Tables 27 and 28. Figures 15 and 16 show the convergence of the optimal solution target function value  $F(X, Y)$  in MLiSSO

**Table 25.** Model parameters setup.

Parameter	a	$\lambda$	b	d	c	$\eta$
Setup	1	0.4	1	0.5	0.4	0.15

**Table 26.** Parameters setup for supply chain model.

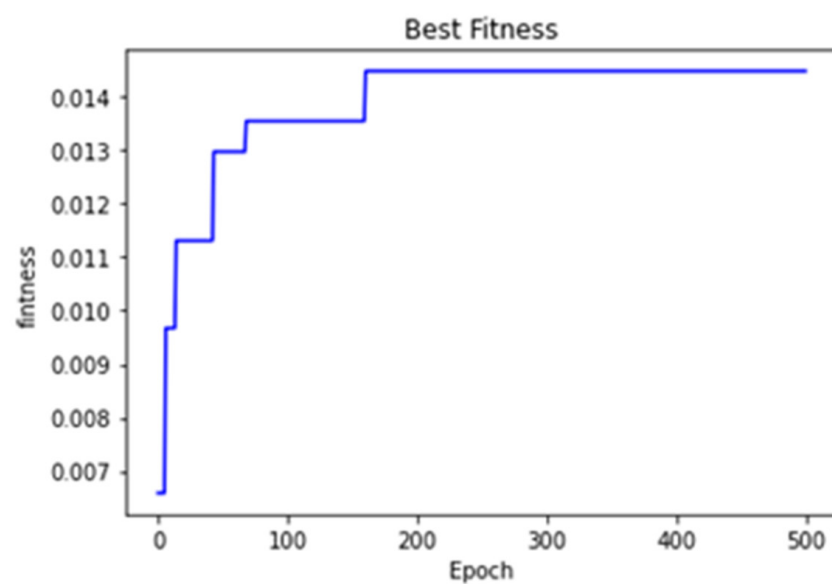
MLiSSO
Population: 20
Cg: 0.2
Cp: 0.3
Cw: 0.5
Generations: 100
Iterations: 500

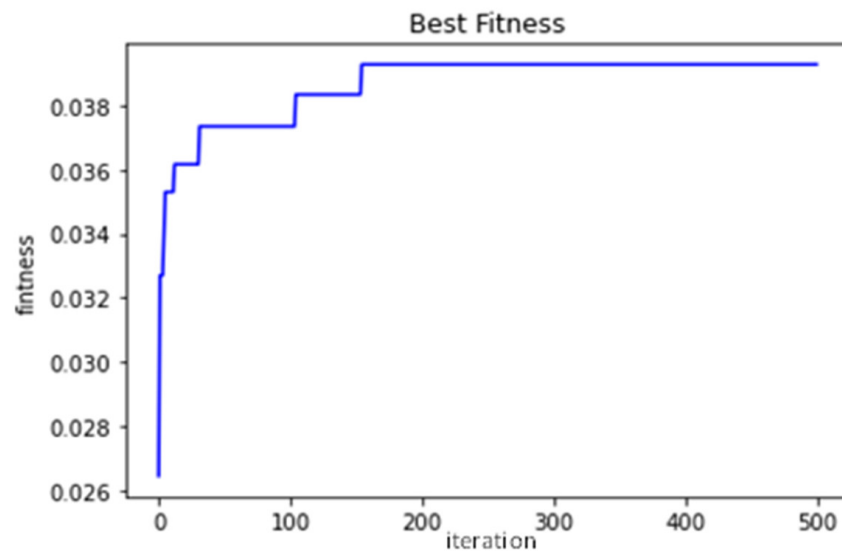
**Table 27.** The best result of the supply chain model.

	RF	BF	3PF
$w$	0.64952	0.40506	0.40506
$P_M$	0.83284	0.70765	0.70765
$P_R$	0.71113	0.62951	0.62951
$q_M$	0.10529	0.12432	0.12432
$q_R$	0.12272	0.20710	0.20710
$r$	0.34810	0.01265	0.01265
$f_1$	0.03231	0.04068	0.04068
$f_2$	0.00756	0.00284	0.00284
$f_3$	N/A	0.00168	0.01487

**Table 28.** Average results & SD of supply chain model.

	RF	BF	3PF
$w$	0.73053	0.40495	0.40495
$P_M$	0.88513	0.71293	0.71293
$P_R$	0.76135	0.61588	0.61588
$q_M$	0.08122	0.14058	0.14058
$q_R$	0.09554	0.19501	0.19501
$r$	0.53832	0.01237	0.01237
$f_1$	0.02197	0.03898	0.03898
$f_2$	0.00355	0.00271	0.00271
$f_3$	N/A	0.00161	0.01664
$f_1$ stdev	0.00524	0.00068	0.00068
$f_2$ stdev	0.00434	0.00265	0.00265
$f_3$ stdev	N/A	0.00153	0.00106

**Figure 15.** Convergence curve of RF.



**Figure 16.** Convergence curve of 3PF & BF.

As shown in Table 27 (1) for the retailer-financed case, the objective value of the leader has converged to  $f_1(x, y, z) = 0.03231$ , while the objective value of layers 2 have converged to  $f_2(x, y, z) = 0.00756$ ; (2) for the bank-financed case, the objective value of the leader has converged to  $f_1(x, y, z) = 0.04068$  while the objective values for layers 2 and 3 converge to  $f_2(x, y, z) = 0.00284$  and  $f_3(x, y, z) = 0.00168$ ; (3) the third-party platform-financed case, the objective value of the leader has converged to  $f_1(x, y, z) = 0.04068$ , while the objective values for layers 2 and 3 converge to  $f_2(x, y, z) = 0.00284$  and  $f_3(x, y, z) = 0.01487$ . We also list the average and standard deviation of the solutions we obtained in 20 runs in Table 28.

It can be noted that under this group of market conditions of parameters and after conversion calculation, as shown in Table 27, we learn that among all financing options, the financing strategy with third-party platforms and banks has absolutely favorable conditions for manufacturers. Thus, it can be concluded that this approach can be applied to complex and practical decision problems to solve MLPP.

## 6. Conclusions

First, we review this paper; our proposed method uses hierarchical updates of fixed variables, trivial problem transformations, computation of objective functions, and iSSO algorithms. Although it does not outperform the best current algorithms for the related small-scale problems, it surpasses the performance of other algorithms for large-scale problems. In conclusion, due to the average and standard deviation of the results, it provides a relatively stable, feasible, and effective solution to the MLPP problem and can be applied to the relevant decision-making process. On the other hand, this paper uses a fixed-variable approach to search, which can express the concept of hierarchical decision-making more effectively and can be implemented on higher-level MLPs that introduce multiple leaders and multiple followers to achieve a more realistic large-scale goal problem. It is also easier to extend to solve a complex problem. If further exploration and experimentation can be done, it may further enhance the ability of this solution to solve problems.

In recent years, many researchers have been studying hybrid algorithms for solving MLPP problems, and as the complexity of the problems increases, mathematical research will become more practical. Therefore, it is expected that more ways and improvements will be developed to solve related problems to meet the industry's current needs.

With the results of this study, the necessity of investigating many of these issues is highlighted, especially to improve the methodology of MLPP. Among the many topics to be explored in future research, there are several major extensions that we intend to focus on.

- (1) Hybridization of other heuristic mechanisms to improve MLISSO solving
- (2) Consider the dynamical mechanism for adjusting the upper and lower terms in terms of the turbulence of the update mechanism to improve the generated solutions towards the desired optimal solution to improve the efficiency and quality of the solutions.

**Author Contributions:** Conceptualization, Z.L. and W.-C.Y.; methodology, Z.L., Y.-C.Y. and W.-C.Y.; writing—original draft preparation, Z.L.; writing—review and editing, Z.L. and S.-Y.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data included in the main text.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Matsui, K. When should a manufacturer set its direct price and wholesale price in dual-channel supply chains? *Eur. J. Oper. Res.* **2017**, *258*, 501–511. [\[CrossRef\]](#)
2. Ma, J.; Wang, H. Complexity analysis of dynamic noncooperative game models for closed-loop supply chain with product recovery. *Appl. Math. Model.* **2014**, *38*, 5562–5572. [\[CrossRef\]](#)
3. Von Stackelberg, H. *Marktform Und Gleichgewicht*; Springer: Berlin/Heidelberg, Germany, 1934.
4. Grand, S.; Von Krogh, G.; Leonard, D.; Swap, W. Resource allocation beyond firm boundaries: A multi-level model for open source innovation. *Long Range Plan.* **2004**, *37*, 591–610. [\[CrossRef\]](#)
5. Dotoli, M.; Fanti, M.P.; Meloni, C.; Zhou, M.C. A multi-level approach for network design of integrated supply chains. *Int. J. Prod. Res.* **2005**, *43*, 4267–4287. [\[CrossRef\]](#)
6. Ma, W.; Wang, M.; Zhu, X. Improved particle swarm optimization based approach for bilevel programming problem—an application on supply chain model. *Int. J. Mach. Learn. Cybern.* **2014**, *5*, 281–292. [\[CrossRef\]](#)
7. Ben-Ayed, O.; Blair, C.E. Computational difficulties of bilevel linear programming. *Oper. Res.* **1990**, *38*, 556–560. [\[CrossRef\]](#)
8. Bard, J.F. Some properties of the bilevel programming problem. *J. Optim. Theory Appl.* **1991**, *68*, 371–378. [\[CrossRef\]](#)
9. Segall, R. Using branch-and-bound to solve bi-level geometric programming problems: A new optimization model. *Appl. Math. Model.* **1990**, *14*, 271–274. [\[CrossRef\]](#)
10. Kasemset, C.; Kachitvichyanukul, V. A PSO-based procedure for a bi-level multi-objective TOC-based job-shop scheduling problem. *Int. J. Oper. Res.* **2012**, *14*, 50–69. [\[CrossRef\]](#)
11. Kuo, R.; Huang, C. Application of particle swarm optimization algorithm for solving bi-level linear programming problem. *Comput. Math. Appl.* **2009**, *58*, 678–685. [\[CrossRef\]](#)
12. Liu, B. Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Comput. Math. Appl.* **1998**, *36*, 79–89. [\[CrossRef\]](#)
13. Kliestik, T.; Zvarikova, K.; Lăzăroiu, G. Data-driven machine learning and neural network algorithms in the retailing environment: Consumer engagement, experience, and purchase behaviors. *Econ. Manag. Financ. Mark.* **2022**, *17*, 57–69.
14. Hopkins, E. Machine Learning Tools, Algorithms, and Techniques. *J. Self-Gov. Manag. Econ.* **2022**, *10*, 43–55.
15. Nica, E.; Sabie, O.M.; Mascu, S.; Luțan, A.G. Artificial Intelligence Decision-Making in Shopping Patterns: Consumer Values, Cognition, and Attitudes. *Econ. Manag. Financ. Mark.* **2022**, *17*, 31–43.
16. Kliestik, T.; Kovalova, E.; Lăzăroiu, G. Cognitive decision-making algorithms in data-driven retail intelligence: Consumer sentiments, choices, and shopping behaviors. *J. Self-Gov. Manag. Econ.* **2022**, *10*, 30–42.
17. Yeh, W.-C. An improved simplified swarm optimization. *Knowledge-Based Syst.* **2015**, *82*, 60–69. [\[CrossRef\]](#)
18. Zhen, X.; Shi, D.; Li, Y.; Zhang, C. Manufacturer’s financing strategy in a dual-channel supply chain: Third-party platform, bank, and retailer credit financing. *Transp. Res. Part E: Logist. Transp. Rev.* **2020**, *133*, 101820. [\[CrossRef\]](#)
19. Tsay, A.A.; Agrawal, N. Channel Conflict and Coordination in the E-Commerce Age. *Prod. Oper. Manag.* **2009**, *13*, 93–110. [\[CrossRef\]](#)
20. Cai, G.G. Channel selection and coordination in dual-channel supply chains. *J. Retail.* **2010**, *86*, 22–36. [\[CrossRef\]](#)
21. Yan, R.; Pei, Z. Information asymmetry, pricing strategy and firm’s performance in the retailer-multi-channel manufacturer supply chain. *J. Bus. Res.* **2011**, *64*, 377–384. [\[CrossRef\]](#)
22. Chiang, W.-Y.K.; Chhajed, D.; Hess, J.D. Direct Marketing, Indirect Profits: A Strategic Analysis of Dual-Channel Supply-Chain Design. *Manag. Sci.* **2003**, *49*, 1–20. [\[CrossRef\]](#)
23. Bernstein, F.; Song, J.-S.; Zheng, X. Free riding in a multi-channel supply chain. *Nav. Res. Logist. (NRL)* **2009**, *56*, 745–765. [\[CrossRef\]](#)



24. Ryan, J.; Sun, D.; Zhao, X. Coordinating a Supply Chain with a Manufacturer-Owned Online Channel: A Dual Channel Model Under Price Competition. *IEEE Trans. Eng. Manag.* **2012**, *60*, 247–259. [\[CrossRef\]](#)
25. Saha, S. Channel characteristics and coordination in three-echelon dual-channel supply chain. *Int. J. Syst. Sci.* **2014**, *47*, 740–754. [\[CrossRef\]](#)
26. Huang, W.; Swaminathan, J.M. Introduction of a second channel: Implications for pricing and profits. *Eur. J. Oper. Res.* **2009**, *194*, 258–279. [\[CrossRef\]](#)
27. Tang, C.S.; Yang, S.A.; Wu, J. Sourcing from Suppliers with Financial Constraints and Performance Risk. *Manuf. Serv. Oper. Manag.* **2018**, *20*, 70–84. [\[CrossRef\]](#)
28. Lee, C.H.; Rhee, B.-D. Trade credit for supply chain coordination. *Eur. J. Oper. Res.* **2011**, *214*, 136–146. [\[CrossRef\]](#)
29. Kouvelis, P.; Zhao, W. Who Should Finance the Supply Chain? Impact of Credit Ratings on Supply Chain Decisions. *Manuf. Serv. Oper. Manag.* **2018**, *20*, 19–35. [\[CrossRef\]](#)
30. Kouvelis, P.; Zhao, W. Supply Chain Contract Design Under Financial Constraints and Bankruptcy Costs. *Manag. Sci.* **2016**, *62*, 2341–2357. [\[CrossRef\]](#)
31. Caldentey, R.; Haugh, M.B. Supply Contracts with Financial Hedging. *Oper. Res.* **2009**, *57*, 47–65. [\[CrossRef\]](#)
32. Aydin, R.; Kwong, C.; Ji, P. Coordination of the closed-loop supply chain for product line design with consideration of remanufactured products. *J. Clean. Prod.* **2016**, *114*, 286–298. [\[CrossRef\]](#)
33. Yang, D.; Jiao, J.; Ji, Y.; Du, G.; Helo, P.; Valente, A. Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game. *Eur. J. Oper. Res.* **2015**, *246*, 263–280. [\[CrossRef\]](#)
34. Cachon, G.P.; Zipkin, P.H. Competitive and cooperative inventory policies in a two-stage supply chain. *Management science* **1999**, *45*, 936–953. [\[CrossRef\]](#)
35. Hennes, J.-C.; Arda, Y. Supply chain coordination: A game-theory approach. *Eng. Appl. Artif. Intell.* **2008**, *21*, 399–405. [\[CrossRef\]](#)
36. Tian, Y.; Govindan, K.; Zhu, Q. A system dynamics model based on evolutionary game theory for green supply chain management diffusion among Chinese manufacturers. *J. Clean. Prod.* **2014**, *80*, 96–105. [\[CrossRef\]](#)
37. Cachon, G.P.; Netessine, S. *Game Theory in Supply Chain Analysis, in Models, Methods, and Applications for Innovative Decision Making*; INFORMS: Catonsville, MD, USA, 2006; pp. 200–233.
38. Leng, M.; Parlar, M. Game Theoretic Applications in Supply Chain Management: A Review. *INFOR Inf. Syst. Oper. Res.* **2005**, *43*, 187–220. [\[CrossRef\]](#)
39. Stackelberg, H.V.; Von, S.H. *The Theory of the Market Economy*; Oxford University Press: Oxford, UK, 1952.
40. Zhou, Z.F. Research on Pricing Decision of Multi-Level Remanufacturing Reverse Supply Chain Based on Stackelberg Game. *Appl. Mech. Mater.* **2012**, *220–223*, 290–293. [\[CrossRef\]](#)
41. Sadigh, A.N.; Mozafari, M.; Karimi, B. Manufacturer–retailer supply chain coordination: A bi-level programming approach. *Adv. Eng. Softw.* **2012**, *45*, 144–152. [\[CrossRef\]](#)
42. Lu, J.; Han, J.; Hu, Y.; Zhang, G. Multilevel decision-making: A survey. *Inf. Sci.* **2016**, *346–347*, 463–487. [\[CrossRef\]](#)
43. Luo, H.; Liu, L.; Yang, X. Bi-level programming problem in the supply chain and its solution algorithm. *Soft Comput.* **2019**, *24*, 2703–2714. [\[CrossRef\]](#)
44. Colson, B.; Marcotte, P.; Savard, G. Bilevel programming: A survey. *4OR* **2005**, *3*, 87–107. [\[CrossRef\]](#)
45. Bard, J. *Practical Bilevel Optimization: Applications and Algorithms, in Series: Nonconvex Optimization and Its Applications*; Springer: Berlin/Heidelberg, Germany, 1998.
46. Faísca, N.P.; Saraiva, P.M.; Rustem, B.; Pistikopoulos, E.N. A multi-parametric programming approach for multilevel hierarchical and decentralised optimisation problems. *Comput. Manag. Sci.* **2007**, *6*, 377–397. [\[CrossRef\]](#)
47. Dempe, S. *Foundations of Bilevel Programming*; Springer Science & Business Media: Berlin, Germany, 2002.
48. Yeh, W.-C. A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems. *Expert Syst. Appl.* **2009**, *36*, 9192–9200. [\[CrossRef\]](#)
49. Eberhart, R.; Kennedy, J. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995.
50. Yeh, W.-C. Orthogonal simplified swarm optimization for the series–parallel redundancy allocation problem with a mix of components. *Knowl.-Based Syst.* **2014**, *64*, 1–12. [\[CrossRef\]](#)
51. Yeh, W.-C. Optimization of the Disassembly Sequencing Problem on the Basis of Self-Adaptive Simplified Swarm Optimization. *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Hum.* **2011**, *42*, 250–261. [\[CrossRef\]](#)
52. Huang, C.-L.; Jiang, Y.-Z.; Yin, Y.; Yeh, W.-C.; Chung, V.Y.Y.; Lai, C.-M. Multi Objective Scheduling in Cloud Computing Using MOSSO. *IEEE Congr. Evol. Comput.* **2018**, *12*, 1–8.
53. Yeh, W.-C. A new exact solution algorithm for a novel generalized redundancy allocation problem. *Inf. Sci.* **2017**, *408*, 182–197. [\[CrossRef\]](#)
54. Yeh, W.-C. New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 661–665.
55. Yeh, W.-C. Novel swarm optimization for mining classification rules on thyroid gland data. *Inf. Sci.* **2012**, *197*, 65–76. [\[CrossRef\]](#)
56. Yeh, W.-C. Simplified swarm optimization in disassembly sequencing problems with learning effects. *Comput. Oper. Res.* **2012**, *39*, 2168–2177. [\[CrossRef\]](#)

57. Yeh, W.-C. A novel boundary swarm optimization method for reliability redundancy allocation problems. *Reliab. Eng. Syst. Saf.* **2019**, *192*, 106060. [[CrossRef](#)]
58. Lin, P.; Cheng, S.; Yeh, W.; Chen, Z.; Wu, L. Parameters extraction of solar cell models using a modified simplified swarm optimization algorithm. *Sol. Energy* **2017**, *144*, 594–603. [[CrossRef](#)]
59. Shen, Y.; Willems, S.P.; Dai, Y. Channel Selection and Contracting in the Presence of a Retail Platform. *Prod. Oper. Manag.* **2018**, *28*, 1173–1185. [[CrossRef](#)]
60. Abhishek, V.; Jerath, K.; Zhang, Z.J. Agency Selling or Reselling? Channel Structures in Electronic Retailing. *Manag. Sci.* **2016**, *62*, 2259–2280. [[CrossRef](#)]
61. Liu, Y.-H.; Hart, S.M. Characterizing an optimal solution to the linear bilevel programming problem. *Eur. J. Oper. Res.* **1994**, *73*, 164–166. [[CrossRef](#)]
62. Bard, J.F.; Falk, J.E. An explicit solution to the multi-level programming problem. *Comput. Oper. Res.* **1982**, *9*, 77–100. [[CrossRef](#)]
63. Amouzegar, M.A.; Cybernetics, P.B. A global optimization method for nonlinear bilevel programming problems. *IEEE Trans. Syst. Man Cybern. Part B (Cybern)* **1999**, *29*, 771–777. [[CrossRef](#)]
64. Guang-Min, W.; Zhong-Ping, W.; Xian-Jia, W.; Ya-Lin, C. Genetic Algorithms for Solving Linear Bilevel Programming. In Proceedings of the IEEE Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05), Dalian, China, 5–8 December 2005; pp. 920–924.
65. Wang, Y.; Jiao, Y.-C.; Li, H. An Evolutionary Algorithm for Solving Nonlinear Bilevel Programming Based on a New Constraint-Handling Scheme. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2005**, *35*, 221–232. [[CrossRef](#)]
66. Wan, Z.; Wang, G.; Sun, B. A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems. *Swarm Evol. Comput.* **2013**, *8*, 26–32. [[CrossRef](#)]
67. Li, X.; Tian, P.; Min, X. A Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems. In *International Conference on Artificial Intelligence and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1169–1178.