The Bi-Level Particle Swarm Optimization for Joint Pricing in a Supply Chain

Umar Mansyuri[1], Andreas Tri Panudju[2], Helena Sitorus[3], Widya Spalanzani[4], Nunung Nurhasanah[5], Dedy Khaerudin[6]
Computer Science Department, Universitas Bina Bangsa, Serang, Indonesia[1]
Industrial Engineering Department, Bhayangkara Jakarta Raya University,Jakarta, Indonesia[2, 3, 4]
Industrial Engineering Department, Al Azhar Indonesia University, Jakarta, Indonesia[5]
Industrial Engineering Department, Universitas Bina Bangsa, Serang, Indonesia[6]

***Abstract*—This study examines the integration of pricing and lot-sizing strategies within a system comprising only one producer and retailer. The adoption of a bi-level programming technique is justified in establishing a bi-level joint pricing model guided by the producer owing to the hierarchical nature of the supply chain. This problem maximizes manufacturer and retailer profitability by setting the wholesale quantity, lot size, and retail price simultaneously. We created a bi-level particle swarm optimization to solve bi-level programming challenges. This algorithm effectively addresses BLPPS by eliminating the need for any priori assumptions about the conditions of the problem. The bi-level particle swarm optimization algorithm demonstrated a commendable level of efficacy when applied to a set of eight benchmark bi-level issues. The proposed bi-level model was solved using the BPSO and analyzed using experimental data**.

Keywords—Bi-Level algorithm; joint pricing; optimization; particle swarm optimization; supply chain

I. Introduction

Supply chain participants are driven by the goal of selling goods, and product prices affect the level of demand in the market [1]. Likewise, the practice of determining the optimal order quantity has resulted in decreased expenses, encompassing both the ordering process for retailers and production costs for manufacturers. Consequently, strategic decisions pertaining to pricing and lot sizing assume crucial significance in the pursuit of profit optimization within a supply chain [2]. In fact, it is critical that the sale price is neither excessively low nor exorbitant. Conversely, when the retail price is set at an excessively low level, the retailer experiences diminished profitability or potentially even financial losses. However, if the product is priced too high, customers will exhibit reduced purchasing intent, resulting in surplus inventory and additional inventory-related expenses for the retailer [3]. Furthermore, because of an overabundance of inventory, the retailer will opt to decrease either the frequency of orders or the amounts ordered from the producer. Consequently, this leads to significant financial losses for producers. Hence, the implementation of rational pricing and lot-sizing policies is of utmost importance and indispensable [4]. Considerable research has been conducted on the issue of optimal pricing, owing to the various factors outlined. In their seminal work, Panchal, Jain, and Kumar (2015) presented a comprehensive model that addresses the intertwined issues of ordering and price discounting in a supplier-buyer relationship [5]. The model focuses on a scenario in which a supplier offers a quantity discount to its sole or significant buyer. However, the problem is approached solely from the supplier's standpoint, neglecting the retailer's perspective on retail pricing. Pourrahmani and Jaller (2021) examined the concept of collaborative pricing and replenishment as a means of addressing a network pricing problem known as NP-hard [6]. Ghahremani-Nahr et al. (2019) primarily concentrated on enhancing an exact algorithm that optimize their strategies for a service operating inside an oligopolistic setting [7]. The issue is framed as a leader-follower game conducted over a distribution network. Yuan (2021) proposed bi-level models to address price problems [8].

This paper utilizes several numerical tests to showcase the efficacy of the BPSO method. Later, the BPSO technique was used to solve the bi-level model. Ultimately, the attributes of the suggested bi-level model are examined through multiple illustrations.

II. Literature Review

The lot-sizing problem, considered a critical challenge in supply chain management, has piqued the interest of various experts who have conducted extensive research on the subject. Diabat et al. (2017) investigated the stochastic variant of lot-sizing issues that incorporate inventory bounds and order capabilities [9]. Kulkarni and Bansal (2022) examined a novel topic involving the lot sizing of multiple items in a dynamic setting [10]. They specifically examined a situation in which all items' inventories are concurrently replenished with an equal quantity after a production event. Gáti and Bányai (2023) Explored the problem of optimizing production quantities and managing resource allocation in an industry with several competing enterprises [11]. The authors present a capacity competition model that incorporates the complexities of fluctuating demand, cost functions, and economies of scale arising from dynamic lot-sizing costs.

Curcio, de Lima, Miyazawa, Silva, and Amorim, (2023) were successfully formulated and determined the best pricing and lot-sizing options for a store [12]. In their work published, Tosarkani & Amin, (2018) examines the issue of pricing and lot-sizing in the context of price-sensitive demand [13]. They formulated the coordination problem between a seller and consumer as a two-person fixed bargaining game. In their study, Bazan et al., (2016) approach by include both the backlogging cost and the cost associated with lost goodwill [14]. Abdulah (2020) provided a deterministic model for calculating the economic order quantity in a retail setting [15]. The model's objective is to accurately predict the effectiveness of a proposed ordering algorithm in mitigating the bullwhip effect to a significant degree.

Each individual within the chain possesses autonomous control over a distinct set of decision variables that do not overlap with the others. These individuals are required to make decisions based on their own personal interests while also taking into account the decisions made by others, as these decisions will impact their own interests. Therefore, bi-level programming problems (BLPPs) are extremely suitable for representing price and lot-sizing challenges inside a supply chain.

In general, solving a bi-level programming problem is challenging for two main reasons. First, bi-level programming problems are classified as NP-hard problems [16]. Second, the concavity of bi-level programming problems further adds to their complexity. Currently, numerous approaches exist to resolve this issue. Four major linear bi-level programming methods were identified in [17], [18]. These categories include neuro-fuzzy algorithms, simulated annealing strategies, vertex enumeration, Kuhn-Tucker conditions, and metaheuristics such as genetic algorithm-based strategies [19].

It is important to note that certain limitations exist when using methods that rely on vertex enumeration and the Kuhn-Tucker conditions to solve bi-level programming problems [20]. These constraints encompass the necessity for the objective function to be differentiable or for the search space to be convex. Metaheuristic approaches are capable of resolving exceedingly complex nonlinear problems in contrast to traditional search algorithms. Management extensively uses metaheuristic optimization [21]. In practice, most bi-level problems extend beyond linear programming and encompass a range of intricate scenarios. Therefore, it is imperative to devise efficient and optimal approaches for addressing these issues.

A Bi-level Linear Programming (BLPP) is a type of multilevel programming issue [22]. Within the framework of BLPP, the decision maker situated at the higher level initially formulates a strategy [23]. The general bi-level formula can be expressed as follows:

$$\min_{x \in X} f_1(x, y)$$

$$s.t\ \ G(x, y) \leq 0,$$

where, the y vector solves

$$\min_{y \in Y} f_2(x, y)$$

$$s.t\ \ g(x, y) \leq 0,$$

Let $x \in X\ h \subset \mathbb{R}^{n1}$ and $y \in Y \subset \mathbb{R}^{n2}$

where,

x: Set of control-level variables

and

y is the set of follower-level variables.

The leader has an objective function, $f_1(x, y)$, and the follower has an objective function of - $f_2(x, y)$.

Additionally, $G(x, y) \leq 0$ and $g(x, y) \leq 0$ represent the constraints associated with upper and lower-level problems, respectively.

III. Method

A. Development Model

After acquiring the raw materials from the supplier, the producer carries out the production and processing procedures (see Fig. 1). The completed products are subsequently delivered to the retailer.

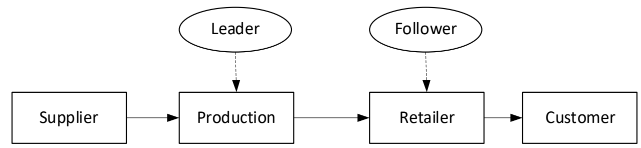

Fig. 1. Two-echelon system.

The producer has the potential to influence the retailer's decisions, but lacks complete control over them. The initial step in the pricing process involves the producer establishing a wholesale price for their items. Subsequently, the retailer responds by considering the producer's assortment and subsequently determines the retail price.

The producer negotiates the price at the wholesale level and quantity ordered with the supplier in order to maximize the net profit, as determined by the primary objective function [24], [25]. The net profit of the producer was calculated by subtracting the costs of purchasing, production, transportation, holding, and ordering from the money generated from sales. The second-level objective function sets forth the retail prices and order quantities for retailers. The primary objective is the retailer's net profit, which is calculated as the ratio of sales to the costs associated with purchasing, holding, and ordering.

A bi-level model was constructed based on specific assumptions [17], [26].

1) The producer or retailer can make separate decisions to maximize earnings.

2) Retail prices decrease consistently with the market demand.

3) Producers and retailers quickly restock.

4) Each replenishment period was consistent, and shortages were avoided.

5) The cost of purchasing components and the price of the product for the end consumer remain constant throughout the planning horizon.

This paper uses these notations:

Aspects that impact the producer's decision

β : The anticipated of orders quantity that the producer will place in the near future.

$\wp_m$: wholesale unit price

Decision-making factors used by the retailer

α : quantity of retailer lots divided by producer lots

ϱ :retailer lot size

p_r :The unit retail price

Additional Relevant Parameters

T : the planned horizon's weekly length

D : weekly demand rate D= b – $a_r.p_r$

h_m : amount of the producer's on a weekly basis storage cost

h_r : retail store's weekly storage cost rate

O_m: Purchasing cost of each order incurred by the producer.

O_r : Purchasing fee for each transaction for the retailer.

p_s : Fee for purchasing a unit from the producer.

T_c : unit transportation cost

M_c : Cost of procuring units for producer.

Q_m : a company's net profit throughout the planning period

Q_r : net profit for the store throughout the planning period

B. Construction Model

The producer and retailer net income during planning are indicated by I_m and I_r, respectively.

$$I_m = (p_m - p_s - T_c - M_c)\alpha\beta Q$$

$$I_r = (p_r - p_m)\alpha\beta Q$$

The average inventory level of the product–retailer combination is determined by the equation α Q/2, where α is a constant. Consequently, the producer's inventory level can be calculated as Q multiplied by the quantity (α − 1)/2.

The producer and retailer holding costs during planning are indicated by H_m and H_r, respectively.

$$H_m = h_m T.\, p_s.\frac{Q\,(\alpha-1)}{2} = \frac{h_m T p_s\, Q\,(\alpha-1)}{2}$$

$$H_r = h_r T.\, p_m.\frac{Q}{2} = \frac{h_r T p_m\, Q}{2}$$

Producer and retailer order expenses within planning are indicated by C_m and C_r, respectively.

$$C_m = \beta O_m$$

$$C_r = \alpha\beta O_r$$

The net earnings of the producer and retailer during planning are indicated by U_m and U_r, respectively.

$$\prod m = I_m - H_m - C_m = (p_m - p_s - T_c - M_c)(\alpha\beta Q) - \left(\frac{h_m T p_s\, Q\,(\alpha-1)}{2}\right) - (\beta O_m) \quad (1)$$

$$\prod r = I_r - H_r - C_r = p_r TD - (p_m \alpha\beta Q) - \left(\frac{h_r T p_m Q}{2}\right) - (\alpha\beta O_r) \quad (2)$$

By integrating Eq. (1) and Eq. (2), the following model is established for the supply chain:

$$\max_{\beta, p_m} \prod m = (p_m - p_s - T_c - M_c)(\alpha\beta Q) - \left(\frac{h_m T p_s\, Q\,(\alpha-1)}{2}\right) - (\beta O_m)$$

$$\text{s.t. } \beta \in N^+, p_s + T_c + M_c \leq p_m \leq p_m^* \quad (3)$$

$$\max_{\alpha, p_r, Q} \prod r = (p_r TD) - (p_m \alpha\beta Q) - \left(\frac{h_r T p_m Q}{2}\right) - (\alpha\beta O_r) \quad \text{s.t. } \alpha \in N^+,$$

The correlation between the quantity of deliveries and the lot size is

$$Q = \frac{TD}{\alpha\beta} \quad (4)$$

Retail prices are believed to be multiples of wholesale prices,

$$P_r = K p_m \quad (5)$$

Let K be the ratio of wholesale to retail prices where K > 1.

It is critical that wholesale and retail pricing remain within the appropriate limits and do not surpass specific levels. By utilizing Eq. (4) and Eq. (5), it is possible to convert Problem (3) into Problem (6) through the following transformation:

$$\max_{\beta, p_m} \prod m = (p_m - p_s - T_c - M_c)(TD) - \left(\frac{h_m T^2 p_s\, D\,(\alpha-1)}{2\alpha\beta}\right) - (\beta O_m)$$

$$\text{s.t. } \beta \in N^+, p_s + T_c + M_c \leq p_m \leq p_m^* \quad (6)$$

$$\max_{\alpha, k} \prod r = (k p_m TD) - (p_m TD) - \left(\frac{h_r T^2 p_m D}{2\alpha\beta}\right) - (\alpha\beta O_r)$$

$$\text{s.t. } \alpha \in N^+, 1 \leq k \leq k^*$$

The variables p_m^* and k^* denote the maximum values of p_m and k.

IV. Result

A. Bi-level PSO-based Algorithms

Kennedy (1995) present a Particle Swarm Optimization (PSO) technique [27]. The Particle Swarm Optimization (PSO) algorithm, as described in [28], is a stochastic evolutionary algorithm that operates on a population-based approach. PSO has demonstrated its effectiveness in addressing complex optimization problems, offering advantages such as simplified coding and a reduced number of parameters [29], [30]. Particle Swarm Optimization (PSO) considers each individual as a particle, disregarding any notions of quality or bulk [31], [32].

The orientation and speed of the particles are denoted by $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, and $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ respectively. The variable $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ represents the best position that the swarm have reached, while $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ represents the best position that the swarm have reached. The manipulation of particles is governed by the following equations:

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 \left(p_{gd}(t) - x_{id}(t)\right) \quad (7)$$

and

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (8)$$

The given conditions are as follows: for any values of d and D where 1 ≤ d ≤ D, and for any values of i and N where 1 ≤ i ≤ N, there exist two non-negative constants denoted as c_1 and c_2 . Additionally, there are two randomly generated integers, denoted as x and y, that follow an equal distribution, in the range of (0,1).

The maximum current value is denoted as $v_{id} \epsilon [-v_{max}, v_{max}], v_{max}$.

When *j*, we set *t*.

The prior velocity of the particle impacts its present velocity through the inertia weight w. The inertia weight w is crucial for balancing the global and local search capabilities. The global search capacity increased as w increased, whereas the local search capacity decreased. Conversely, when the value of *w* decreased, the opposite situation occurred.

Shi and Eberhart (1999) offer an approach known as linear decline, which is outlined as follows.

$$w(t) = w_{min} + (w_{max} - w_{min}).\left(\frac{t_{max}-t}{t_{max}}\right) \quad (9)$$

The *"t"* represents the t^{th} iteration, while " t_{max} " represents the maximum number of iterations for that iteration. The initial inertia weight is represented by variables w_{min} and w_{max}, which indicate the smallest and highest values, respectively.

B. Managing Constraint

The upper and lower-level problems in BLPP (see Eq. (1)) are standard constraint optimization that do not consider the leader-follower information interaction [33]. Managing constraint is crucial for constraint optimization.

Consider the constraint optimization problem as follows:

$$\min F(x)$$

$$\text{s.t. } g_i(x) \leq 0, i = 1,2,\ldots,p \quad (10)$$

where, S is the search space, x ϵ S, and S ⊆ R^n.

By incorporating a penalty element, Eq. (10) can be reformulated as follows:

$$\min F(x) = f(x) + M \sum_{i=1}^{p} (\max\{g_i(x), 0\})^2,$$

Variable M represents a predetermined and significantly large positive constant, denoted as a penalty factor.

This approach is analogous to treating upper-level programming problems. It is assumed that the lower-level programming problem comprises p and q inequality constraints. In addition, it is assumed that the variable x of the upper-level programming problem is predetermined. Within the realm of the search area, a particle that adheres to the given restrictions is referred to as a feasible particle, whereas a particle that fails to meet the criteria is labeled as an infeasible particle. In the present scenario, it is possible to determine the finesse of all particles, both feasible and infeasible, using the following equations:

$$\text{fit}(x,y) = \begin{cases} f(x,y), if\ y \in \Omega(x) \\ f(x,y), if\ y \in S \backslash \Omega(x) \end{cases} \quad (11)$$

and

$$F(x,y) = f(x,y) + M \sum_{i=1}^{m} (\max\{g_i(x,y), 0\})^2 , \quad (12)$$

where, *S* indicates search area, while Ω (x) is the feasible set of lower-level problem.

C. Implementation

Based on an analysis of the interacting iterations of two fundamental PSO algorithms, it has been observed that Binary Particle Swarm Optimization (BPSO) can solve BLPP without relying on any specific assumptions [34], [35], such as the availability of gradient information for the objective functions or the convexity of constraint regions. The details are presented in the subsequent sections.

Algorithm 1:

Step 1: Preparation of parameters by measuring population N_1, determining maximum iteration T_{max1}, with learning factors c_1 and c_2, determining maximum and minimum inertia weights w_{max} and w_{min}, as well as maximum speed v_{max} x, and enforcement factor M.

Step 2: Set the position x_i and velocity v_{xi} of each particle to the upper level's decision variables. The begin position y_i and velocity v_{yi} are based on lower-level decision parameters.

Step 3: Set the loop counter of the leader to $t_1 = 0$.

Step 4: If the algorithm fulfills completion conditions or an upper limit of iterations, move to the final step; otherwise, follow Steps 4.1–4.5.

Step a: For each x_i, Algorithm 2 solves lower-level programming problems and determine the optimal solution for y^*, as the follower response.

Step b: Calculate the particle fitness values using Eq.(11) and (12)

Step c: The best particle (p_{xi}) and population (p_{xg}) positions are recorded. If p_{xi} is higher than the best in history, a new pxi is declared. Choose the particle with the highest fitness value for p_{xg}.

Step d: Update the particle positions using Eq.(7)–(9).

Step e: $t_1 = t_1+1$.

Step 5: If it the maximum number of iterations, proceed to Step 5. Otherwise, proceed to step 3.

Step 6: Final best results.

Algorithm 2 :
Step 1: Preparation of parameters N_2 (population), and T_{max2} (maximum iteration)
Step 2: Start with the subsequent loop counter $t_2 = 0$.
Step 3: Eq. (11) and (12) are used to calculate particle fitness.
Step 4: Record the particle's i^{th} best location p_{yi} and population's p_{yg}.
Step 5: Update the follower position and velocity using Eq. (7) – (9).
Step 6: $= t_2 + 1$.
Step 7: Move to Step 8 if the algorithm reaches its maximum number of iterations. Otherwise, proceed to step 5.
Step 8: Optimum lower-level problem solutions y^* .

D. Simulation

For every bi-level problem, we perform 20 different executions of the BPSO algorithm. Table I presents the optimal and average outcomes obtained from solving the four linear bi-level programming problems using the BPSO algorithm. Comparative analyses of these results are provided in Tables II and III. Table IV presents the optimal outcomes obtained by solving the four nonlinear bi-level programming problems using the BPSO, along with the corresponding comparisons.

Table I lists the four linear test function solutions for the BPSO algorithm. Based on the findings presented in Tables II and III, it can be inferred that the outcomes obtained from solving the four linear bi-level programming problems using BPSO exhibit superior performance in terms of both the best and average results, as compared to the results obtained using the GA and PSO algorithms.

The BPSO algorithm finds the optimal solutions for four nonlinear bi-level problems: Problem 5's best solution is (0.0422, 1.9339, 2.8674, 1.4395), the optimal solution for the 6th problem is (0.7886, 1.8556), the optimal solution for 7th problem is (3.9960, 0.0004), and the optimal solution for 8th problem is (0.5014, 0.2026, 0.8275).

Based on Table IV, in terms of optimizing the upper-level objective function of the 5th problem, it was observed that the BPSO algorithm exhibited superior accuracy compared to the other three algorithms. However, disparities between their performances were minimal. To optimize the upper-level objective function of the 7th problem, the performance of BPSO was found to be comparable to that of the other three algorithms, with only a small difference. However, the BPSO outperformed the other three methods when considering the lower-level objective function.

TABLE I. BPSO-BASED RESULTS

Simulations	*Best Result*		*Average Result*	
	f_1	f_2	f_1	f_2
1st	977090	- 478546	880162	- 458824
2nd	109967	- 109967	109471	- 109471
3rd	157155	- 39212	156555	- 39345
4th	297327	- 30834	290413	- 32731

TABLE II. BEST ALGORITHM COMPARISONS

Test problems		Binary PSO		GA		PSO	
		The best solution	The best value	The best solution	The best value	The best solution	The best value
1st	*f1*	x = (13.713,11.378)	97.7092	x = (17.458,10.906)	85.0551	x = (17.454,10.907)	85.08
	f2		- 47.854		- 50.170		- 50.175
2nd	*f1*	x = (15.759,10.997)	10.997	x = (15.998,10.998)	10.998	x = (15.999,10.999)	10.999
	f2		- 10.997		- 10.998		- 10.999
3rd	*f1*	x = (3.952,3.922)	15.717	x = (3.999,3.997)	15.997	x = (4,6)	18
	f2		3.921		- 3.9466		- 5
4th	*f1*	x = (0.193,0.392)	29.733	x = (0.000,0.898)	29.148	x = (0.004,0.899)	29.179
	f2	y = (0.544,0.675, 0.445)	3.0835	y = (0.000,0.600, 0.400)	- 3.193	y = (0,0.600, 0.400)	- 3.198

TABLE III. AVERAGE RESULTS FROM DIFFERENT ALGORITHMS

Test problems		Average values		
		Binary PSO	G_A	PSO
1st	*f1*	88.016	84.658	84.852
	f2	- 45.883	- 50.030	- 50.078
2nd	*f1*	10.947	10.808	10.997
	f2	- 10.947	- 10.808	- 10.997
3rd	*f1*	15.656	15.827	15.988
	f2	- 3.933	- 3.947	- 3.9963
4th	*f1*	29.043	21.529	24.816
	f2	- 3.271	- 3.392	- 3.198

TABLE IV. BEST ALGORITHM RESULTS COMPARISONS

Test problems		Results			
		Binary PSO	Hybrid_PSO_BLP	T_R_M	Original
5th	*f1*	- 15.203	- 14.758	- 12.78	- 12.78
	f2	1.036	0.207	- 1.026	- 1.026
6th	*f1*	66.956	88.777	88.80	88.89
	f2	- 15.837	- 0.770	- 0.87	- 0.87
7th	*f1*	2.019	2	2	2
	f2	23.981	24.018	24.04	24.22
8th	*f1*	1.038	2.709	2.76	2.85
	f2	- 0.517	0.562	0.76	0.67

E. Evaluation

This section solves bi-level joint pricing and lot-sizing model (6) using the BPSO method. Determination of Eq. (6) Parameters

T = 52, hm = 0.001,

O_m = 2000,

p_s = 4,

h_r = 0.001,

O_r = 200,

T_c = 0.5, and M_c = 1.

To enhance the analysis of our model, we initially established an upper constraint for the wholesale price, denoted as p = 10. We examine two upper limits for the retail-to-wholesale pricing ratio, designated k* = 2 and k* = 5. The parameters' configurations of BPSO are consistent with those outlined in the previous section. The outcomes of the optimization process are presented in Tables V and VI.

TABLE V. RESULTS FROM DIFFERENT COEFFICIENT GROUPS (K*=2)

Demand function coefficients	α	β	p_m	k	p_r	Q	$\prod m$	$\prod r$
a = 2, b = 600	4	4	9.8 84	1.9 69	19.4 55	18 26	1192 92	2761 20
a = 4, b = 600	4	2	9.4 08	1.9 43	18.2 62	34 27	1018 60	2407 60
a = 6, b = 600	3	4	9.7 95	1.9 34	18.9 37	21 18	1001 82	2288 30
a = 8, b = 600	3	3	9.6 86	1.9 36	18.7 38	26 11	9133 4	2101 50

TABLE VI. RESULTS FROM DIFFERENT COEFFICIENT GROUPS (K*= 5)

Demand function coefficients	α	β	p_m	k	p_r	Q	$\prod m$	$\prod r$
a = 2, b = 600	3	4	9.32 23	4.68 85	43.70 76	22 21	951 88	9141 10
a = 4, b = 600	5	2	9.82 55	4.45 34	43.75 69	22 10	906 68	7478 30
a = 6, b = 600	5	3	9.59 64	4.77 33	45.80 65	11 27	627 95	6092 50
a = 8, b = 600	2	2	9.81 84	4.30 1	42.22 89	34 08	545 17	4410 50

Fig. 2 and 3 depict a decrease in the net profits of both the producer and retailer in response to an increase in demand price sensitivity.

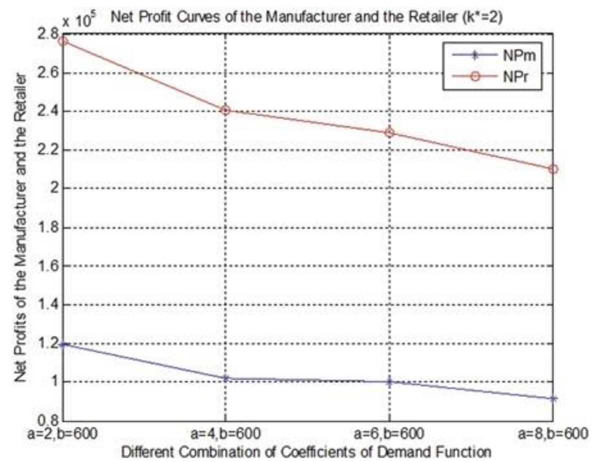

Fig. 2. Producer and retailer net profits curves under k =2.

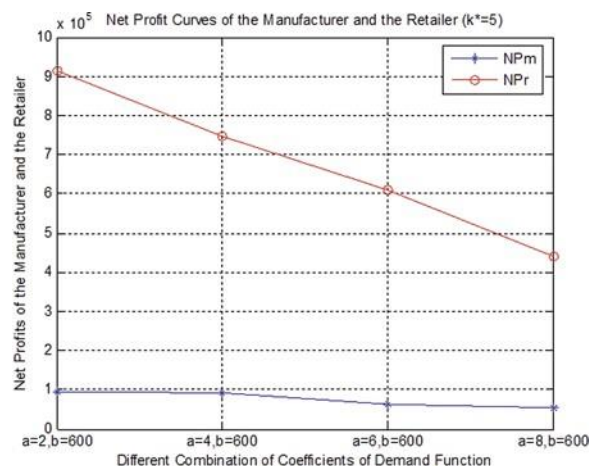

Fig. 3. Producer and retailer net profit curves under k =5.

Fig. 4 and Fig. 5 illustrate the variations in net profits for both the producer and the shop at different levels of demand. Based on the examination of Fig. 4 and Fig. 5, it can be observed that the net profit of the producer is comparatively reduced when the value of k is 5 in contrast to when it is 2, assuming an identical demand function. In contrast, the net profit of the store exhibits a notable increase when the value of k is 5 as opposed to 2. The observed gap can be ascribed to the differential behavior of retail and wholesale prices, specifically when k is equal to 5. In this scenario, the retail price increases, whereas the wholesale price remains comparatively stable. This is in contrast to the situation where k is equal to 2.

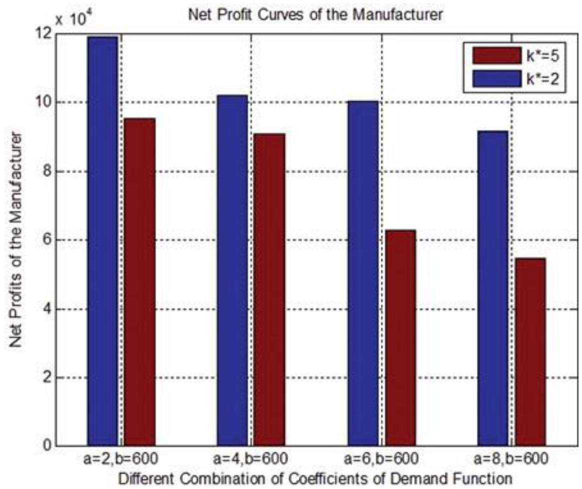

Fig. 4. Comparison of producer net profits.

If the producer's interests are violated because of an increased retail price, the producer may choose to increase the wholesale price. As a result, this course of action possesses the capacity to increase retail pricing, culminating in a decline in the sales volume of products as a consequence of the augmented selling price. The drop in profitability for both the producer and retailer will result in a subsequent decline in the overall efficiency of the entire supply chain. Therefore, to effectively maximize their respective profits and avoid being caught in a harmful cycle where market demand decreases due to rising retail prices, it is recommended that both the producer and retailer refrain from continuously increasing the price of the product.

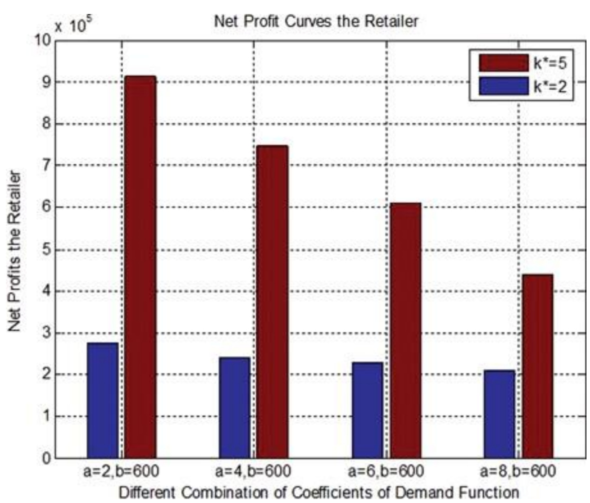

Fig. 5. Comparison of retailer net profits.

V. DISCUSSION

The bi-level particle swarm optimization (BLPSO) is an effective method for solving intricate decision-making problems in supply chain management. The BLPSO algorithm has been utilized in many supply chain models, providing effective solutions for combined pricing and strategic sourcing strategies [36]. Research has demonstrated that it has exceptional search performance and rapid convergence, rendering it a highly useful instrument for addressing bi-level optimization problems in supply chain management [37]. Moreover, the simulation findings indicate that the large-scale BLPSO method has enhanced stability and supremacy when it comes to tackling supply chain optimization challenges [38].

In addition, researchers have investigated the combination of BLPSO with other metaheuristic methods, such as genetic algorithms, to improve its effectiveness in solving bi-level linear programming issues related to supply chain management. The hybrid strategy has demonstrated potential in effectively tackling the combined issues of pricing and inventory control in a two-level supply chain [39]. In addition, the creation of innovative dynamic Pareto BLPSO algorithms has broadened the use of BLPSO in addressing multi-objective optimization challenges in supply chain management [40].

BLPSO has been applied to solve optimization challenges beyond supply chain management, including job shop scheduling and the selection of locations for electric vehicle charging infrastructure [41]. The BLPSO algorithm's ability to effectively handle various optimization issues demonstrates its potential to tackle intricate decision-making challenges across a wide range of fields.

In summary, utilizing BLPSO in the context of joint pricing within a supply chain provides a strong and effective method for tackling intricate decision-making challenges. The tool's capacity to manage multi-objective optimization, strategic sourcing, and large-scale supply chain models renders it a significant asset for optimizing decision-making processes in supply chain management.

VI. CONCLUSION

The primary objective of this model is to optimize the profits of both the producer and retailer. This optimization is achieved by simultaneously determining the number of orders for both parties, lot size for the retailer, and wholesale and retail prices. Based on the characteristics of the bi-level programming issue and the resulting bi-level model discussed in this article, we provide a BPSO technique to identify the most effective solutions.

The BPSO algorithm was employed to address the bi-level model presented in this study. The results obtained include the ideal number of orders for both the producer and retailer, optimal lot size for the retailer, and optimal wholesale and retail prices. These optimal values were simultaneously determined by considering the specified constraints.

Based on the collected data, the analysis reveals certain outcomes that align with market principles. Notably, one of these outcomes indicates a negative correlation between the sensitivity of demand to price and net profits of both the producer and retailer. It is also observed that in cases where market demand is responsive to changes in selling price, both the producer and retailer should not consistently increase the wholesale and retail prices if their goal is to maximize individual profits. This approach may result in unfavorable outcomes and reduce overall supply chain efficiency. The findings of this study further corroborate the effectiveness of the BPSO in addressing BLPPS.

REFERENCES

[1] G. Baralla, A. Pinna, and G. Corrias, "Ensure traceability in european food supply chain by using a blockchain system," Proc. - 2019 IEEE/ACM 2nd Int. Work. Emerg. Trends Softw. Eng. Blockchain, WETSEB 2019, pp. 40–47, 2019, doi: 10.1109/WETSEB.2019.00012.

[2] A. Hendalianpour, "Optimal lot-size and Price of Perishable Goods: A novel Game-Theoretic Model using Double Interval Grey Numbers," Comput. Ind. Eng., vol. 149, p. 106780, 2020, doi: 10.1016/j.cie.2020.106780.

[3] M. J. Knuth, H. Khachatryan, and C. R. Hall, "How consistent are consumers in their decisions? Investigation of houseplant purchasing," Behav. Sci. (Basel)., vol. 11, no. 5, p. NA, 2021, doi: 10.3390/bs11050073.

[4] R. Li and J.-. T. Teng, "Pricing and lot-sizing decisions for perishable goods when demand depends on selling price, reference price, product freshness, and displayed stocks," Eur. J. Oper. Res., vol. 270, 2018, doi: 10.1016/j.ejor.2018.04.029.

[5] G. B. Panchal, V. Jain, and S. Kumar, "Multidimensional utility analysis in a two-tier supply chain," J. Manuf. Syst., vol. 37, pp. 437–447, 2015, doi: 10.1016/j.jmsy.2014.07.001.

[6] E. Pourrahmani and M. Jaller, "Crowdshipping in last mile deliveries: Operational challenges and research opportunities," Socioecon. Plann. Sci., vol. 78, 2021, doi: 10.1016/j.seps.2021.101063.

[7] J. Ghahremani-Nahr, R. Kian, and E. Sabet, "A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm," Expert Syst. Appl., vol. 116, pp. 454–471, 2019, doi: 10.1016/j.eswa.2018.09.027.

[8] G. Yuan, Y. Gao, B. Ye, and Z. Liu, "A bilevel programming approach for real-time pricing strategy of smart grid considering multi-microgrids connection," Int. J. Energy Res., vol. 45, no. 7, pp. 10572–10589, 2021, doi: 10.1002/er.6545.

[9] A. Diabat, A. A. Taleizadeh, and M. Lashgari, "A lot sizing model with partial downstream delayed payment, partial upstream advance payment, and partial backordering for deteriorating items," J. Manuf. Syst., vol. 45, pp. 322–342, 2017, doi: 10.1016/j.jmsy.2017.04.005.

[10] K. Kulkarni and M. Bansal, "Discrete multi-module capacitated lot-sizing problems with multiple items," Oper. Res. Lett., vol. 50, no. 2, pp. 168–175, 2022, doi: 10.1016/j.orl.2022.01.002.

[11] K. V. Gáti and T. Bányai, "Impact of dynamic lot sizing techniques on costs of material requirement planning," Adv. Logist. Syst. - Theory Pract., vol. 17, no. 1, pp. 71–78, 2023, doi: 10.32971/als.2023.009.

[12] E. Curcio, V. L. de Lima, F. K. Miyazawa, E. Silva, and P. Amorim, "The integrated lot-sizing and cutting stock problem under demand uncertainty," Int. J. Prod. Res., vol. 61, no. 20, pp. 6691–6717, 2023, doi: 10.1080/00207543.2022.2136279.

[13] B. M. Tosarkani and S. H. Amin, "A possibilistic solution to configure a battery closed-loop supply chain: Multi-objective approach," Expert Syst. Appl., vol. 92, pp. 12–26, 2018, doi: 10.1016/j.eswa.2017.09.039.

[14] E. Bazan, M. Y. Jaber, and S. Zanoni, "A review of mathematical inventory models for reverse logistics and the future of its modeling: An environmental perspective," Appl. Math. Model., vol. 40, no. 5–6, pp. 4151–4178, 2016, doi: 10.1016/j.apm.2015.11.027.

[15] A. A. Alabdulkarim, "Minimizing the bullwhip effect in a supply chain: a simulation approach using the beer game," Simulation, vol. 96, no. 9, pp. 737–752, 2020, doi: 10.1177/0037549720930284.

[16] A. Hiassat, A. Diabat, and I. Rahwan, "A genetic algorithm approach for location-inventory-routing problem with perishable products," J. Manuf. Syst., vol. 42, pp. 93–103, 2017, doi: 10.1016/j.jmsy.2016.10.004.

[17] H. Sun, Z. Gao, and J. Wu, "A bi-level programming model and solution algorithm for the location of logistics distribution centers," Appl. Math. Model., vol. 32, no. 4, pp. 610–616, 2018, doi: 10.1016/j.apm.2007.02.007.

[18] Y. Ma, F. Yan, K. Kang, and X. Wei, "A novel integrated production-distribution planning model with conflict and coordination in a supply chain network," Knowledge-Based Syst., vol. 105, pp. 119–133, 2016, doi: 10.1016/j.knosys.2016.05.007.

[19] G. Haeser and A. Ramos, "Constraint Qualifications for Karush–Kuhn–Tucker Conditions in Multiobjective Optimization," J. Optim. Theory Appl., vol. 187, no. 2, pp. 469–487, 2020, doi: 10.1007/s10957-020-01749-z.

[20] H. A. E.-W. Khalifa et al., "Utilization of neutrosophic Kuhn-Tucker's optimality conditions for Solving Pythagorean fuzzy Two-Level Linear Programming Problems," Int. J. Neutrosophic Sci., vol. 23, no. 3, pp. 87–96, 2024, doi: 10.54216/ijns.230308.

[21] S. Abbaspour, A. Aghsami, F. Jolai, and M. Yazdani, "An integrated queueing-inventory-routing problem in a green dual-channel supply chain considering pricing and delivery period: A case study of construction material supplier," J. Comput. Des. Eng., vol. 9, no. 5, pp. 1917–1951, 2022, doi: 10.1093/jcde/qwac089.

[22] N. A. Alessa, "Bi-Level linear programming of intuitionistic fuzzy," Soft Comput., vol. 25, no. 13, pp. 8635–8641, 2021, doi: 10.1007/s00500-021-05791-5.

[23] J. F. Monge and J. L. Ruiz, "Setting closer targets based on non-dominated convex combinations of Pareto-efficient units: A bi-level linear programming approach in Data Envelopment Analysis," Eur. J. Oper. Res., vol. 311, no. 3, pp. 1084–1096, 2023, doi: 10.1016/j.ejor.2023.05.034.

[24] S. Ahmadian, H. Malki, and A. R. Sadat, "Modeling Time of Use Pricing for Load Aggregators Using New Mathematical Programming with Equality Constraints," 2018 5th Int. Conf. Control. Decis. Inf. Technol. CoDIT 2018, pp. 38–44, 2018, doi: 10.1109/CoDIT.2018.8394778.

[25] A. Aazami and M. Saidi-Mehrabad, "Benders decomposition algorithm for robust aggregate production planning considering pricing decisions in competitive environment: A case study," Sci. Iran., vol. 26, no. 5 E, pp. 3007–3031, 2019, doi: 10.24200/sci.2018.5563.1346.

[26] S. Pramanik, P. P. Dey, P. Pratim, and D. " Bi, "Bi-level Linear Programming Problem with Neutrosophic Numbers," Neutrosophic Sets Syst., vol. 21, p. 1, 2019.

[27] J. C. Bansal, "Particle swarm optimization," Stud. Comput. Intell., vol. 779, pp. 11–23, 2019, doi: 10.1007/978-3-319-91341-4_2.

[28] Q. Yang et al., "Stochastic Cognitive Dominance Leading Particle Swarm Optimization for Multimodal Problems," Mathematics, vol. 10, no. 5, 2022, doi: 10.3390/math10050761.

[29] R.-O. Dynhora-Danheyda, P.-D. L. Asunción, and M.-G. E. Adán, "Comparison of PSO with the Hybrid Algorithms MOORA-PSO and DA-PSO for Decision Making," Adv. Model. Anal. B, vol. 66, no. 1–4, pp. 26–30, 2023, doi: 10.18280/ama_b.661-404.

[30] S. Grassi and L. Pareschi, "From particle swarm optimization to consensus based optimization: Stochastic modeling and mean-field limit," Math. Model. Methods Appl. Sci., vol. 31, no. 8, pp. 1625–1657, 2021, doi: 10.1142/S0218202521500342.

[31] Y. W. Chen, L. C. Wang, A. Wang, and T. L. Chen, "A particle swarm approach for optimizing a multi-stage closed loop supply chain for the solar cell industry," Robot. Comput. Integr. Manuf., vol. 43, pp. 111–123, 2017, doi: 10.1016/j.rcim.2015.10.006.

[32] G. Sen, J. Sharma, G. Goyal, and A. Singh, "A Multi-objective PSO (MOPSO) algorithm for optimal active power dispatch with pollution control," Math. Model. Eng. Probl., vol. 4, no. 3, pp. 113–119, 2017, doi: 10.18280/mmep.040301.

[33] R. Arora and K. Gupta, "A linear fractional bilevel programming problem with multichoice parameters," Croat. Oper. Res. Rev., vol. 8, no. 2, pp. 499–513, 2017, doi: 10.17535/crorr.2017.0032.

[34] M. I. Alghamdi, "Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization (BPSO)," Sustain., vol. 14, no. 19, 2022, doi: 10.3390/su141911982.

[35] A. Da Li, B. Xue, and M. Zhang, "Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies," Appl. Soft Comput., vol. 106, 2021, doi: 10.1016/j.asoc.2021.107302.

[36] S. P. Venkatesan and S. Kumanan, "Multi-Objective Supply Chain Sourcing Strategy Design Under Risk Using PSO and Simulation," Int. J. Adv. Manuf. Technol., 2011, doi: 10.1007/s00170-011-3710-y.

[37] Z. Zhang and J.-X. Xu, "Bi-Level Multiple Mode Resource-Constrained Project Scheduling Problems Under Hybrid Uncertainty," J. Ind. Manag. Optim., 2015, doi: 10.3934/jimo.2016.12.565.

[38] J. Jiang, W. Wen-xue, W.-L. Shao, and Y. Qu, "Research on Large-Scale Bi-Level Particle Swarm Optimization Algorithm," Ieee Access, 2021, doi: 10.1109/access.2021.3072199.

[39] A. Mahmoodi, "Pricing and Inventory Decisions in a Manufacturer-Stackelberg Supply Chain With Deteriorating Items," Kybernetes, 2020, doi: 10.1108/k-03-2020-0149.

[40] A. Aboud, R. Fdhila, A. Hussain, and A. M. Alimi, "A Novel Dynamic Pareto Bi-Level Multi-Objective Particle Swarm Optimization (DPb-MOPSO) Algorithm," 2020, doi: 10.36227/techrxiv.13325354.v1.

[41] Y. Yan, "Research on Location Selection of Electric Vehicle Charging Facilities Based on Hybrid Algorithm," 2024, doi: 10.1117/12.3024048.