

---

# *Innovative Resource Distribution Through Multi-Agent Supply Chain Scheduling Leveraging Honey Bee Optimization Techniques*

*Abdulameer Almusawi and S.Pugazhenth<sup>2</sup>*

---

<sup>1</sup>*Department of computers Techniques engineering,  
College of technical engineering,  
Najaf*

<sup>2</sup>*Department of Computer Science,  
Sri Moogambigai Arts and Science College,  
Mallupatti, Dharmapuri*

## **ABSTRACT**

Many people and things involved in the supply chain can be represented by a network of linked agents in a multi-agent system. To achieve distribution and manufacturing targets efficiently, actions in a supply chain must be scheduled by tracking and organizing the allocation of resources. Healthcare, energy, aerospace, agriculture, and manufacturing are some of the industries that make use of it to better allocate and coordinate resources in complex networks. The goal of multi-agent supply chains is to lower costs and meet demand in dynamic contexts through efficient resource allocation and planning. Traditional approaches, such as linear programming and heuristic algorithmic approaches, have problems with adaptation and scalability. This study offers a novel scheduling method, Resource Distribution by Honey Bee Optimization (RDHBO), to circumvent these issues. It is based on a tactic that the honey bee uses when foraging. In this model, scout bees and foragers take on the role of actors, investigating and making the most of potential new ways to divide up resources by employing HBO strategies. The foraging agents look for places that have been profitable in the past, and the scouting agents carefully examine the supply chain to find places where resources might be available. The proposed RDHBO architecture allows for better supply chain scheduling by encouraging effective utilization and investigation, better communication, dynamic adaptability, autonomous decision-making, and efficiency of complicated systems. This method leads to an adaptable supply chain, lower operational costs, and more scalability and adaptability. This paper presents a formidable alternative to conventional optimization methods by demonstrating an efficient approach to dealing with complicated supply chain challenges.

*Keywords:* Multi-agent system, Honeybee optimization, Resource distribution, Supply chain, Resource allocation, Scalability, and Adaptability

## **1. Introduction**

A MAS is a network of independent nodes that can communicate with one another, share data, and work together to complete a task. By allowing MASs to operate autonomously, suppliers can reduce demand and improve device performance [1]. The primary goal of a system with several agents is to control and monitor the access rights of each agent to patient data. Data access privileges are role-specific and assigned to agents according to their job

descriptions. The system uses multiple privacy parameters to make sure that agents can't see any unauthorized data. In real-time, these factors—including secrecy, availability, and integrity—are evaluated to detect any possible privacy violations. The multi-agent system's accuracy of diagnostic and treatment suggestion has been substantially improved with deep learning [2]. Supply chain optimisation is crucial in today's sectors since it aims to promote customer delight, lower expenses, and improve operational efficiency. However, if dealing with complex and dynamic supply chain networks, traditional optimization algorithms struggle to use resources efficiently [3]. Two primary areas of supply chain management see the most use of multi-agent technology: model development and system realization. To address the issue of information variation across platforms and languages, a multi-agent collaboration structure was developed for the supply chain's production and sales processes. This framework outlined the framework's basic operation flow and offered a satisfactory plan for collaborative production and sales [4]. Multi-agent systems (MAS) can improve job effort distribution and scheduling decision-making by extracting their independent and automatic behavior. Conversely, MAS can get solutions quickly by agents working together and using smart control, which is made possible by their capacity to communicate and work together [5].

From simple distribution and production networks to complex global ones, management of supply chains has come a long way. Streamlining operations, minimizing costs, and mitigating risks have all been made possible with the help of data-driven insights, predictive modeling, and powerful optimization algorithms made possible by the emergence of AI technology [6]. Improving visibility across the whole product lifecycle, from order delivery to disposal, requires cooperation between suppliers, producers, and consumers [7]. Multiple parties with varying interests and priorities are at work in this supply chain, including producers, wholesalers, retailers, consumers, and third-party logistics providers. Nevertheless, due to their interconnected nature, every move one participant in a supply chain makes affects the others. Analyzing and predicting their reactions throughout time is problematic in SCNs because of the features of nonlinearity and the numerous different interactions between entities [8]. Adaptable scheduling as well as a management system for production is essential due to the growing personalization of products. While dealing with the high degree of dimension and complexities of the data and the prevalent and ever-changing nature of the environment, it is possible to boost the procedure and goods quality through the accessibility of data linked to quality, the environment, and the process [9]. Traditional scheduling algorithms often struggle to deal with unpredictable and ever-changing situations, which is where scheduling decisions for production jobs often take place. Because it learns and adjusts its tactics in real-time, deep reinforcement learning is better able to adjust to new conditions [10].

This paper suggests incorporating honey bee optimization along with the resource distribution (RDHBO) framework to optimize multi-agent supply chain scheduling. The proposed method presents a fresh perspective on the distribution of supply chain resources. One possible solution to the problem of managing the unpredictable complexity of modern supply chains is to implement a scheduling system that makes use of RDHBO techniques inside a multi-agent framework. The innovative system's objective is to fortify resistance, productivity, and flexibility.

The primary contribution of the work is

- ✓ The main objective of this work is to create a novel framework for organizing multi-agent supply chains that uses RDHBO techniques for resource distribution.
- ✓ The suggested RDHBO structure, which allows for effective and adaptable allocation of resources along the supply chain, is inspired by honey bee foraging behavior.
- ✓ By representing various entities and procedures as agents, this model of the supply chain allows for better task coordination and allocation of resources.

- ✓ RDHBO has several benefits over older methods, such as the ability to optimize complicated systems, decentralize decision-making, improve communication, and increase dynamic adaptability.
- ✓ This results in reduced operational expenses, improved scalability, and a more responsive supply chain network.

This study presents a new approach for organizing a multi-agent supply chain by studying how honey bees forage. Individuals representing supply chain members engage in autonomous environmental exploration to identify resource-efficient allocation strategies. Use of a pheromones deposits mechanism for communication, waggle dance for information sharing, and job splitting into "scout" and "forager" agents allows for the utilization of known solutions and the search for new work, respectively, are all examples of heuristic approaches. Enhancing the resource utilization rate, reducing costs, sensitivity to disturbances, enhancing the value of infrastructure, and making systems extensible as well as robust are some benefits that can be derived from the use of this theory.

## 2. Literature Survey

The honey-bee mating optimization (HBMO) technique was used by Haddad B et al. [11] to address mathematics and engineering issues, and it has been demonstrated to be effective in finding optimal solutions for domains with many challenges. The HBMO method, handling projects with limited and unlimited resources, was introduced in this study. Findings demonstrate that the HBMO method applies to both resource-constrained and non-constrained projects. In addition, the outcomes are encouraging and hold their own against popular heuristic and gradient-based methods. However, problems with parameter tweaking and regional search algorithms persist.

To improve the model's performance, Ren, L. et al. [12] suggested a multi-agent reinforcement learning system that uses an encoder-decoder architecture. Iteratively, the decoder determines the routes for each vehicle, while the encoder extracts the relationships between the consumer nodes in the problem. Specifically, we develop numerous route recorders to gather data about vehicles' past routes and enable their interconnection. As soon as a new instance is created, the model may begin inferring and immediately begin generating routes for all vehicles.

In their study, Ahmed et al. [13] used an algorithm modeled after a beehive to streamline the partner selection procedure for a supply chain network with multiple tiers. To optimize profit and satisfy customers, the artificial bee colony algorithm is used to choose the best partners. The optimization technique for artificial bee colonies is based on observations of how bees act in the wild. The ABC is based on the premise that a colony of artificial bees or other multi-agent systems can effectively solve challenging combinatorial optimization issues. A faster rate of convergence is required of the suggested technology.

To enhance the mechanism for allocating virtual machines, Ouham, S. et al. [14] suggested a hybrid approach. A first improvement happens in the local search section of the Grey Wolf Optimization (GWO) and Artificial Bee Colony (ABC) algorithms, respectively. A second improvement happens in the fitness function and energy parameter. Energy intake, throughput network stability, average network execution time, and resource allocation systems in virtual machines for cloud computing are the four primary scheduling metrics that are improved by the aforementioned technique. Its exploring skills and convergence speed might use some improvement, though.

Undertaking several assumptions, the model put forth by Komasilova, O. et al. [15] calculates the ideal number of colonies of honey bees required for foraging in a given region.

The authors suggest thinking about things like field production, chemical contamination, and the apiary's environment. Before the model can be run, many tasks must be fulfilled. These include choosing an area of interest, converting it to polygons for additional computations, and finally, outlining the roads within the chosen area. The user is shown the result of the model, which is the recommended number of colonies to plant.

An approach called Chaotic Improved Honey Bee Mating Optimization (CIHBMO) was proposed by Niknam, T. et al. [16]. It combines the improved HBMO with a Chaotic Local Search (CLS). Considerations of nonlinear generator properties, such as multi-fuel operation, forbidden operating zones, and valve position loading effects, are incorporated into the ED problem-solving process. We compare the CIHBMO algorithm to various approaches in the literature and test it on three test systems. Despite the persisting difficulty of striking a balance between exploration and exploitation, the results demonstrate that the suggested approach is quick and efficient for ED situations involving fuel cost functions that are neither smooth nor continuous.

In their study, Katiyar et al. [17] employed Swarm Intelligence (SI), a subset of nature-based algorithms that may explore globally and provide adaptive, optimal solutions to real-time situations. Optimizing a distribution route for fresh food delivery within a time window while considering other factors such as a fixed number of delivery vehicles, fixed cost and fuel, and all service locations has been tackled by researchers using an artificial bee colony (ABC) algorithm and a cuckoo search (CS) algorithm. The study's findings demonstrate an effective method that does not sacrifice either efficiency or food quality. Not taking into account the dynamics of real-time traffic for delivery vehicles and other environmental circumstances means that some constraints persist.

Combining the greedy algorithm with a feature-processing sequence, a processing method, and a machine tool, Yang, G., et al. [18] developed a honey-bee mating optimization algorithm (HBMO) and five worker bee cultivation strategies to address the issues of infeasible solutions and local optimization during drone mating. Lastly, the combined optimization of organizing processes and workshop scheduling is executed, using the complicated parts of such an aerospace industry as an example. However, unexpected events like urgent orders and machine breakdowns during production cannot be eliminated.

This article discusses the scheduling and multi-objective integrated process planning issues that Wen, X., et al. [19] encountered, which involve unpredictable processing times and due dates. To address the suggested model, an efficient modified honey bee mating optimization method was developed. To preserve the non-dominated solutions discovered during optimization, a queens set is built. A method for calculating the mating probability of drones and queen bees using the formula based on the geometrical distance between them is developed. The outcome is a highly efficient solution for scheduling and planning multi-objective uncertain integrated processes.

Using a multi-agent deep reinforcement learning (RL) method, Naderializadeh, N., et al. [20] were able to manage distributed resources and mitigate interference in wireless networks. Agents can make judgments in a dispersed fashion and at the same time without knowing what other agents are doing at the same time. This proves that our instructed agents can withstand differences in test and train environments and keep their performance improvements.

### 3. Proposed Methodology

#### a. Dataset Explanation

A supply chain analysis allows for a deep investigation and improvement of different supply chain components by providing a comprehensive analysis of the company's orders and distribution activities [21]. It includes information on sales, inventory control, procurement, and customer satisfaction and offers insights for making decisions based on data. All things considered, it makes cost reduction, increased customer satisfaction, and efficiency improvement possible through improved operations.

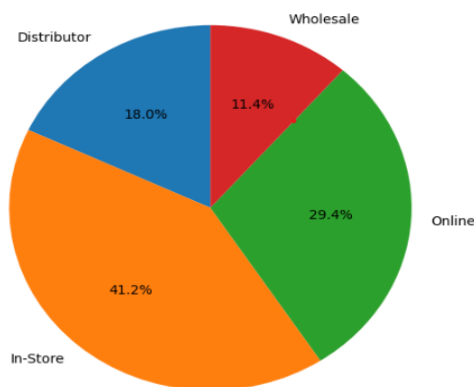


Figure.1 Profit for Different Sales Channels

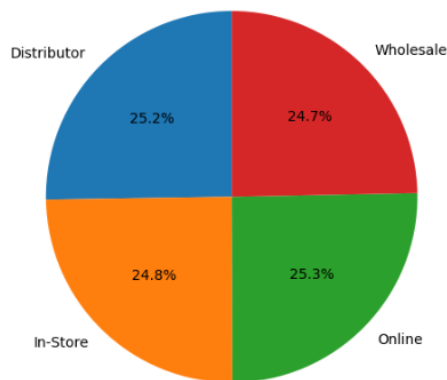


Figure.2 Average delivery time for different sales channels

Figure 1 illustrates analyzing profit across several sales channels offers insightful information about how well each channel generates income and adds to total profitability. Figure 2 shows the average delivery time for several sales channels. Businesses can determine which channels are doing effectively and which ones might need to be adjusted or optimized by analyzing the earnings from several channels, including sales online, physical retail stores, wholesale distribution, and sales to customers directly. To optimize profitability across the board, this study assists firms in allocating resources efficiently, giving top priority to high-performing channels, and possibly restructuring or improving underperforming ones.

#### b. Problem definition and Goal setting

Interdependent links in a supply chain include data systems, producers, distributors, retailers, logistics companies, and suppliers as shown in Figure 3.

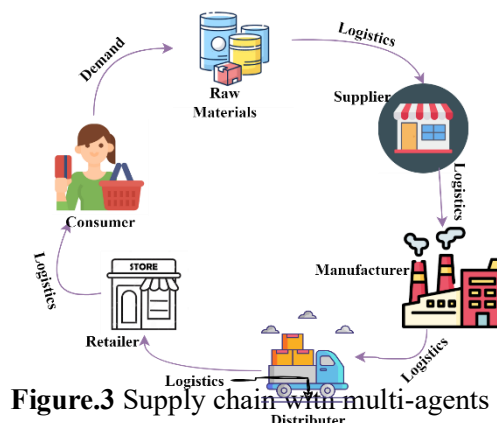


Figure.3 Supply chain with multi-agents

The primary focus is on ensuring the efficient and effective control of the flow of data, services, and products from their origin to their consumption point, all while satisfying customer expectations and reducing costs. The end goal is to minimize expenses and maximize efficiency by delivering the proper products or services to clients on time. By coordinating the efforts of all involved parties, supply chain management boosts efficiency, which in turn increases profits, delights customers, and gives businesses an edge in the market. In Figure 3, the supply chain agents are depicted as nodes, and the logistical systems are shown as edges.

### c. Honey Bee Optimization Algorithm

A bio-inspired algorithm that uses honey bee foraging behavior as a model to address optimization issues is Honey Bee Optimization (HBO). It shines in dealing with scheduling, resource allocation, and other complicated decision-making situations. Honey bees have an impressive foraging procedure, in that to indicate prospective areas and randomly search the area for possible food sources the scout bees perform waggle dance. Volunteer forager bees graze for nectar and pollen at these designated food sources, while observer bees use dances to determine the best places to visit depending on distance and quality data. Once a food supply runs dry, foragers will leave it and hunt for something else to eat. The most efficient food sources are located and used by the colony through an ongoing cycle of scout discovery, forager exploitation, dance information exchange, adaptation to abandon diminished resources, and group decision-making using shared knowledge. With these guiding ideas in mind, the HBO algorithm solves difficult optimization problems by communicating, adapting, exploring, and exploiting.

Honey Bee Optimization (HBO) assigns separate responsibilities to the scout agents, who explore potential solutions, and the forager agents, who actively seek new ones.

Assume that a vector  $z$  represents the resource allocation solution space.  $z = \{z_1, z_2, \dots, z_n\}$

where the quantity of resources is  $n$ . The way scout & forager agents would revise their positions is as follows:

#### i) Forager Agent (Exploitation)

It is the job of forager agents to make use of the existing solutions and also look for superior ones in their immediate vicinity. Forager agents often utilize the following Eqn.1 while doing local searches.

$$z_i^{new} = z_i^{old} + \emptyset \times (z_i^{old} \times z_r) \quad (1)$$

where  $z_i^{new}$  is the new solution,  $z_i^{old}$  is the current solution being exploited.  $z_r$  is the population's solution chosen at random, and  $\emptyset$  is the random number between -1 and 1. This equation adds a perturbation to the existing solution ( $z_i^{old}$ ) proportionate to the difference among  $z_i^{old}$  and other randomly picked solutions ( $z_r$ ), thus changing the solution. The size and orientation of the disturbance can be controlled by the  $\emptyset$  parameter, which enables the forager agent to investigate the immediate vicinity of the present solution.

#### ii) Scout Agent (Exploration)

Scout agents are tasked with discovering new and improved solutions by exploring the whole solution space. This is a typical eqn.2 that scout agents employ as they go exploring:

$$z_i^{new} = z_i^{min} + ra_1 \times (z_i^{max} - z_i^{min}) \quad (2)$$



where  $z_i^{new}$  is the new solution generated by the scout agent,  $z_i^{max}$ ,  $z_i^{min}$  are the upper & lower bounds of the space of possible solutions, and  $ra_1$  is the random integer between zero and one. Scout agents' exploration keeps the population diverse and stops the algorithm from reaching local optima. The scout agents' ability to generate novel solutions increases the likelihood that they will find superior answers in uncharted parts of the solution space.

*d. Proposed Resource Distribution Using Honey Bee Optimization (RDHBO) Framework*

The supply chain is modeled by the RDHBO framework as a multi-agent system, in that forager agents use local search to find possible solutions and scout agents investigate novel resource distribution schemes or schedules. Agents' direct exploration and exploitation via waggle dancing to convey the quality of the result. Solutions are evaluated using a fitness function that considers both the satisfaction of demand and the minimizing of costs. The framework adapts to changes by dropping less-than-ideal schemes and converges towards the best options through iterative improvement that combines scout and forager agent actions. RDHBO optimizes complex supply chains by utilizing collective intelligence and decentralized agent decision-making that is modeled after the foraging behavior of honey bees. In comparison to previous approaches, it improves flexibility, adaptability, lower expenses, and responsiveness by facilitating dynamic ability to adapt, effective exploration/utilization, improved interaction, and system efficiency optimization.

<p><b>Algorithm 1</b> Pseudocode of HBOA</p> <p><b>Initialize parameters:</b></p> <p>S = number of scout agents</p> <p>F = number of forager agents</p> <p>T = maximum number of iterations</p> <p>solution_space = define_soln_space()</p> <p><b>Generate initial population of solutions:</b></p> <p>solns = initialize_solns(soln_space, S)</p> <p>evaluate_fitness(solns)</p> <p>Main optimization loop:</p> <p><b>for</b> iteration = 1 to T:</p> <p><b>Step 2:</b> Exploration by scout agents</p> <p>new_solns = []</p> <p><b>for</b> i = 1 to S:</p> <p>    new_soln = explore_soln_space(soln_space)</p> <p>    new_solns.append(new_soln)</p> <p><b>end for</b></p> <p>evaluate_fitness(new_solns)</p> <p><b>Step 3:</b> Combine and select the best solutions</p> <p>combined_solns = solns + new_solns</p>
--

```

    sort combined_solns in descending order of fitness
    best_solns = first F solns from combined_solns
Step 4: Exploitation by forager agents
    forager_solns = []
    for each soln in best_solns:
        improved_soln = local_search(soln)
        forager_solns.append(improved_soln)
    end for
    evaluate_fitness(forager_solns)
Step 5: Communication and update
    solns = best_solns + forager_solns
    sort solutions in descending order of fitness
    solns = first S solns from solns
Step 6: Check convergence (optional)
    if check_convergence(solns):
        break
    end if
end for
Output the best solution found:
    best_soln = solns[0]
    print("Best soln:", best_soln)

```

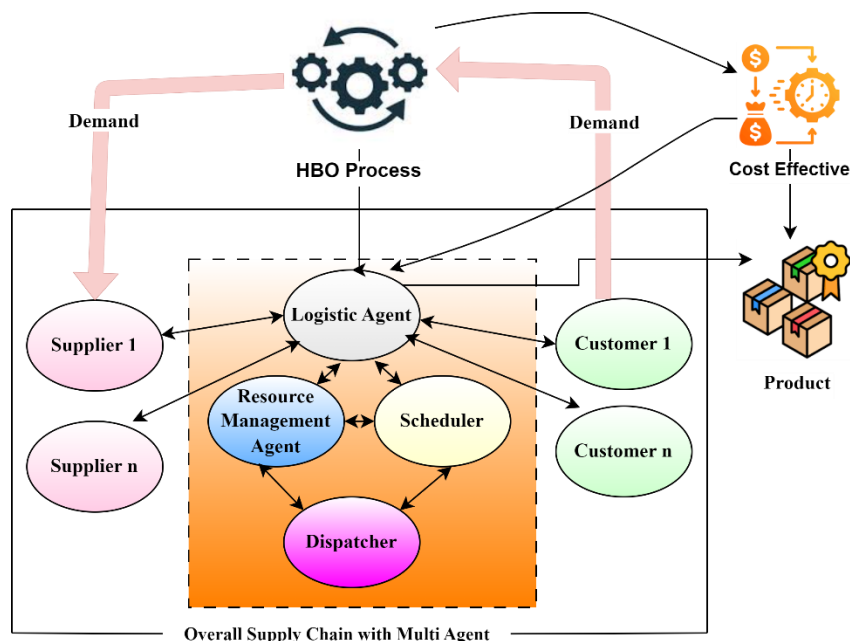
Algorithm 1 starts by initializing the highest possible number of iterations, the outcome space, and the total amount of scout and forager agents. It creates a starting population of solutions and assesses how suitable they are. Once inside the primary optimization loop, it continues for the predetermined number of iterations. Each cycle ends with a fitness assessment of the scout agents' newly generated concepts, that they derived from their search for answers space. The best answers are chosen after combining these new and already-existing solutions and sorting them according to fitness. Using local search, forager agents take advantage of the best results to produce better results that are then assessed for fitness. The optimal and enhanced solutions are merged, and arranged according to fitness, and the best results from the new population. The convergence check is optionally carried out, and if it is satisfied, the loop is ended. The method outputs the best answer identified after the main loop.

*e. Resource distribution using the proposed RDHBO Framework*

The multi-agent supply-chain is similar to how honey bees forage. Scout bees locate needs similarly to how they plan for them. Choosing lucrative food sources is a planning process for both production and procurement. Manufacturing completed things for inventory is the process of gathering nectar. Nectar delivery preparation is triggered by customer orders, just as order processing. Nectar transportation and distribution resemble supply network logistics and distribution. Reverse logistical planning for returns is what it means to give up on exhausted resources. The cooperation and information exchange across supply chain participants are



shown in communications on food sources. Importantly, the best worker bee distribution between food sources according to profitability is similar to how resources are distributed throughout the supply chain to effectively satisfy consumer expectations. This chain process is obtained from Figure 4.



**Figure.4** The Multi-agent Supply-Chain Process of the Suggested RDHBO

The following equations can be defined to mathematically express the main elements and procedures of the RDHBO framework:

i) *Solution Representation:*

Assume that  $\sigma$  is the solution space, and each solution  $\omega \in \sigma$  corresponds to a supply chain timetable or resource distribution plan.

ii) *Fitness Function:*

A solution  $\omega$ 's fitness is assessed using a fitness function  $f(\omega)$  of eqn.3, which takes into account several goals, including demand fulfillment and cost minimization:

$$f(\omega) = a \times cost(\omega) + b \times ds(\omega) + c \times obj(\omega) \tag{3}$$

where  $cost(\omega)$ ,  $ds(\omega)$ , and  $obj(\omega)$  are functions that assess the individual objectives for the solution  $\omega$ , and  $a, b$ , and  $c$  are weights that establish the relative relevance of each objective.

iii) *Scout Agent Discovery:*

To come up with fresh ideas, scout agents investigate the solution space. The exploration function, denoted as  $E(\omega, \sigma)$ , produces a new solution  $\omega'$  by utilizing the solution space  $\sigma$  and the existing solution  $\omega'$  as shown in eqn.4.

$$\omega' = E(\omega, \sigma) \tag{4}$$

A variety of strategies, including heuristic-based approaches, problem-specific operators, and random sampling, can be used to create the exploration function  $E(\omega, \sigma)$ .

iv) *Exploitation by Forager Agents:*

By using local search algorithms, forager agents take advantage of viable solutions. The exploitation function, denoted by  $L(\omega)$ , in eqn.5, produces an enhanced solution  $\omega'$  by leveraging the existing solution  $\omega$ .

$$\omega' = L(\omega) \quad (5)$$

There are several ways to implement the exploitation function  $L(\omega)$  in local search, including hill climbing, neighborhood search, and other problem-specific heuristics.

v) *Communication and Information Sharing:*

Like in honey bee colonies, agents waggle dance to convey the standards (fitness) of their solutions.

$$C(\omega, f(\omega)) \quad (6)$$

Let eqn.6 be the representation of the communication function. It communicates with other agents the fitness information  $f(\omega)$  of the solution  $\omega$ .

vi) *Iterative Improvement:*

The following eqn.7 illustrates the iterative improvement process:

$$\sigma_{t+1} = \sigma_t \cup \{E(\omega, \sigma_t) \mid \omega \in \sigma_t\} \cup \{L(\omega) \mid \omega \in \sigma_t\} \quad (7)$$

where  $\sigma_t$  represents the set of solutions at iteration  $t$  and  $\sigma_{t+1}$  represents the updated set of solutions following the incorporation of fresh solutions produced by scout agents ( $E(\omega, \sigma_t)$ ) and enhanced solutions produced by forager agents ( $L(\omega)$ ).

The main elements and procedures of the RDHBO framework are represented mathematically by these equations. With all these equations the proposed framework allocates resources to the supply chain, provides the product with the lowest cost, and improves supply chain responsiveness.

## 4. Results and Discussion

By simulating intricate multi-agent supply chains as decentralized systems modeled after the foraging behavior of honey bees, the RDHBO framework optimized them. Local forager agents looked for promising resource schedules, and scout agents investigated new prospects. To assess the quality of the solution based on cost and demand fulfillment goals, agent communication was modeled after waggle dance. Through iterative collective intelligence, RDHBO showed adaptability and flexibility, quickly responding to changes and converging towards ideal answers. In comparison to traditional methods, it significantly reduced operational costs and improved flexibility, ability to adapt, cost-effectiveness, and reactivity in dynamic supply-chain environments. It also significantly improved crop yields, shipping losses, patient wait times, and asset utilization across industries. Section 3.1 discusses the datasets that were utilized to analyze the suggested model.

### a. *Experimental Setup*

In this module, the proposed Resource Distribution using the Honey Bee Optimization (RDHBO) framework is compared with conventional algorithms such as Artificial Bee

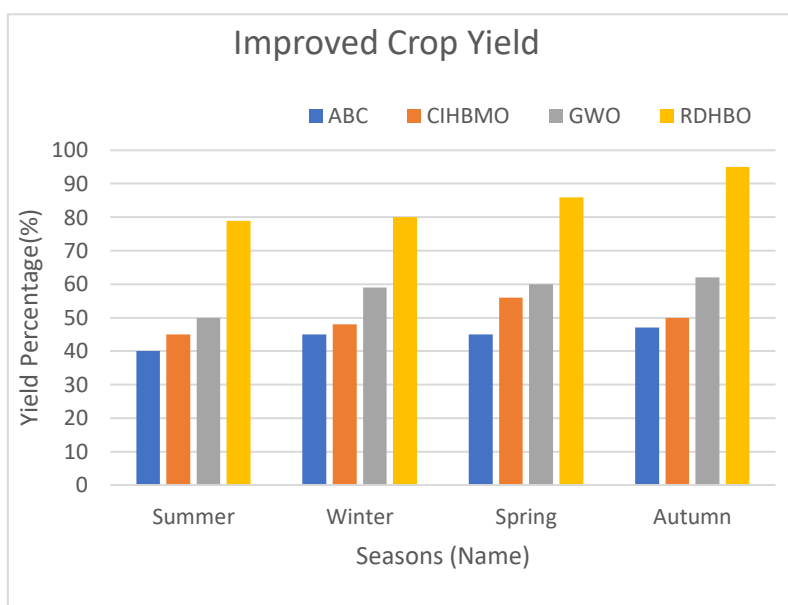
Colony[17], Chaotic Improved Honey Bee Mating Optimization (CIHBMO) [16], and Grey Wolf Optimization[14] for the better results. We use a set of standard cases and real-world supply chain data to compare the algorithm based on metrics like cost reduction, Improved crop yield, and overall supply chain performances. To investigate the impact of various issue instance properties and algorithm parameters on the outcomes, the experimental strategy will incorporate sensitivity and statistical investigations.

*b. Improved crop yield in the Agricultural supply chain.*

Increased crop yields, especially when employed in agricultural supply chains, can be a crucial performance parameter for optimizing supply chain frameworks like RDHBO. This statistic evaluates how optimization efforts have increased agricultural productivity. The quantity of crop cultivated per unit area of land is referred to as crop yield. Usually, it is expressed in terms of tons per hectare (t/ha).

$$YI \% = \left( \frac{Y_o - Y_i}{Y_i} \right) \times 100 \tag{8}$$

where *YI %* in eqn.8 refers to the percentage increase in yield,  $Y_o$  is the post-optimization yield after the implementation of the RDHBO framework and  $Y_i$  is the initial yield before the implementation of the RDHBO framework. This increases the agricultural supply chain's efficiency and output.



**Figure.5** RDHBO Framework's Effect on Crop Yields: A Comparative Examination

As shown in Figure.5 the yield percentage of crops in agricultural fields is increased by implementing the RDHBO framework. This, in turn, leads to higher efficiency and better service levels and also increases in profit margin.

*c. Lead Time Variability*

The reliability of the time required to complete an order is quantified by the Lead Time Variability by eqn 9. A more flexible and dependable supply chain is one with lower variability since it can reliably fulfill delivery promises regardless of variations in demand or supply. several independent agents, each standing in for distinct supply chain entities or processes.

$$\Phi_{LT} = \sqrt{\frac{1}{N} \sum_{j=1}^N (LT_j - LT_a)^2} \quad (9)$$

where  $LT_j$  is the lead time for the order  $j$ ,  $LT_a$  is the average lead time over  $N$  orders. The lower  $\Phi_{LT}$  indicates less variability and higher consistency in lead times, showing that the supply chain is more flexible and adaptive.

#### d. Interaction and Communication Metrics

The RDHBO framework encompasses metrics for collaboration and communication within the supply chain, evaluating the effectiveness of agent cooperation and the efficiency of information exchange. This can be quantified using equation 10.

$$\text{Overall Communication efficiency} = \frac{WDE \times ACI}{CL} \quad (10)$$

where  $WDE$  refers to the Waggle Dance Efficiency, which guarantees that the appropriate resources are identified and utilized through the assessment of the accuracy and effectiveness of quality communication.  $ACI$ , on the other hand, stands for the Agent Collaboration Index, which assesses the level of collaboration among agents.  $CL$  is the Communication Latency, determining how quickly data can be transferred, an essential factor for making decisions and responding quickly. The values of eqn 10 can be calculated by the following eqns 11, 12, and 13.

$$WDE = \frac{\sum_{j=1}^N (RQ_s(j) - AQ_s)(SQ_f(j) - AS_f)}{\sqrt{\sum_{j=1}^N (RQ_s(j) - AQ_s)^2 \sum_{j=1}^N (SQ_f(j) - AS_f)^2}} \quad (11)$$

where  $RQ_s(j)$  is the quality of resource  $j$  as assessed by scout agents,  $SQ_f(j)$  is the success of resource  $j$  as assessed by forager agents,  $AQ_s$  is the average quality score by scout agents, and  $AS_f$  is the average success score by forager agents. When the  $WDE$  of eqn 11 is high, which means that the communication is efficiently conveying the quality of the resources, and a greater correlation.

$$ACI = \frac{sc}{t} \quad (12)$$

where  $sc$  is the successful collaborative interactions, and  $t$  is the total number of possible interactions. Agents collaborate more effectively with a higher  $ACI$ .

$$CL_{jk} = T_j - T_k \quad (13)$$

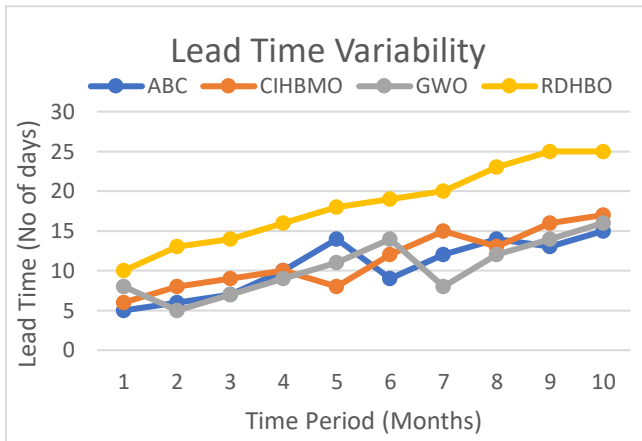
$$ACL = \frac{1}{N} \sum_{l=1}^N CL_{jk} \quad (14)$$

Communication Latency for each iteration  $CL_{jk}$  can be calculated by the eqn 13. Average latency  $ACL$  for overall iteration  $N$  can be calculated by the eqn 14.

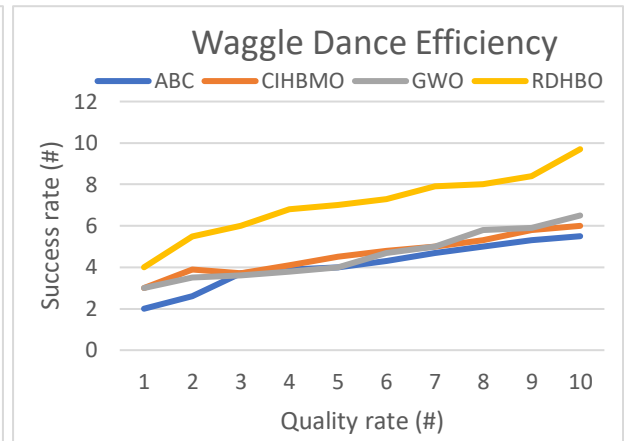
where  $T_j$  is the time for generating the information by the agent  $j$ ,  $T_k$  is the time when the same information is received and acted upon by the agent  $k$ . For quicker decision-making, lower latency is preferable.

Figure 6 shows the comparative study for the Variability of Lead Time throughout RDHBO framework with the traditional methods. Figure 7 shows the Waggle Dance Efficiency gauges how effectively scout and forager agents convey the quality of resources. It illustrates how well the waggle dance communicates important information by comparing it with the traditional methods. Improved communication results from higher correlation. By comparing real relationships to all potential interactions, the Agent Collaboration Index quantifies the

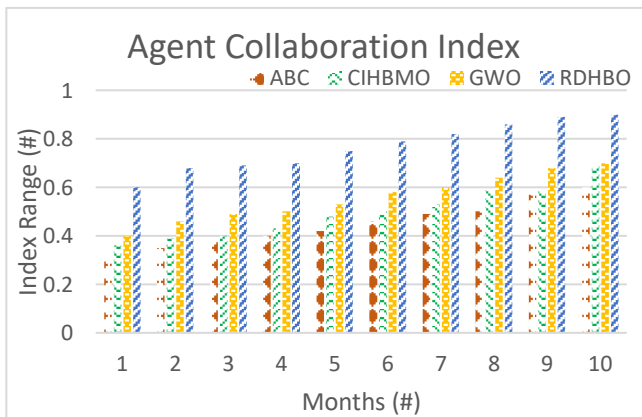
efficacy of agent collaboration as shown in Figure 8. The data it offers is more valuable than those of more conventional methods. Figure 9 shows the inverse operation of communication latency by monitoring the time it takes for a specific agent to transmit data and others is receive and analyze it.



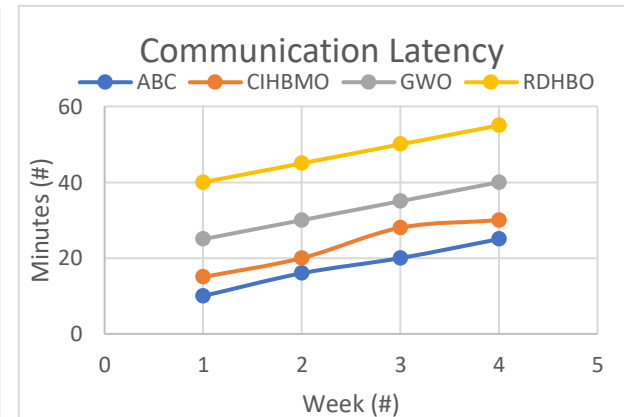
**Figure.6** Variability of Lead Time Over Time



**Figure.7** Waggle Dance Efficiency Over Scenario



**Figure.8** Over Time – The Agent Collaboration Index



**Figure.9** Communication Latency Over Time

## 5. Conclusion

The suggested Resource Distribution by Honeybee Optimization (RDHBO) framework offers a novel bio-inspired strategy to tackle the complex issues related to scheduling and resource allocation in multi-agent supply chain systems. This strategy aims to distribute resources across intricate supply chain networks by mimicking the actions of honey bees, that deploy scouts and foragers to seek out and grab opportunities. The RDHBO methodology considers the important issues of scalability and adaptability, unlike traditional optimization methods such as heuristic algorithms and linear program design. It allows agents to be dynamically adaptable by letting them react independently to changes in demand and supply. Scout and forager agents utilize decentralized decision-making to enhance the effectiveness of resource exploration and consumption. Efficient sharing of resources and well-organized efforts are guaranteed by improved communication between agents. The ability to optimize complex

systems is one of the main features of the RDHBO framework. As an effective alternative to conventional optimization methods, it mimics natural approaches. It is believed that by using the framework, supply chain procedures in many industries will become more efficient, timely, and cost-effective. To establish a foundation for more robust and sustainable operations, the RDHBO method demonstrates how nature-based techniques can enhance supply chain management procedures. Although the RDHBO framework is currently tested and proven in theoretical and computational contexts, it has to be put into practice and validated in real-world supply chain situations. Future studies could focus on applying the RDHBO structure to particular industries or supply-chain topologies, such as logistics, e-commerce, or healthcare. Modifications made to fit a certain domain may necessitate adjustments to the requirements of the framework, including adding domain-specific restrictions or objectives.

### References

- [1]. Binyamin, S. S., & Ben Slama, S. (2022). Multi-agent systems for resource allocation and scheduling in a smart grid. *Sensors*, 22(21), 8099
- [2]. Benmamoun, Z., Khlie, K., Dehghani, M., & Gherabi, Y. (2024). WOA: Wombat Optimization Algorithm for Solving Supply Chain Optimization Problems. *Mathematics*, 12(7), 1059.
- [3]. Dhasarathan, C., Shanmugam, M., Kumar, M., Tripathi, D., Khapre, S., & Shankar, A. (2024). A nomadic multi-agent based privacy metrics for e-health care: a deep learning approach. *Multimedia Tools and Applications*, 83(3), 7249-7272.
- [4]. Wei, D. (2020). Modeling and Simulation of a Multi-agent Green Supply Chain Management System for Retailers. *Journal Européen des Systèmes Automatisés*, 53(4).
- [5]. Alves, F., Rocha, A. M. A., Pereira, A. I., & Leitão, P. (2022). Conceptual Multi-Agent System Design for Distributed Scheduling Systems. In *Smart and Sustainable Manufacturing Systems for Industry 4.0* (pp. 129-148). CRC Press.
- [6]. Abaku, E. A., Edunjobi, T. E., & Odimarha, A. C. (2024). Theoretical approaches to AI in supply chain optimization: Pathways to efficiency and resilience. *International Journal of Science and Technology Research Archive*, 6(1), 092-107.
- [7]. Fierro, L. H., Cano, R. E., & García, J. I. (2020). Modeling of a multi-agent supply chain management system using Colored Petri Nets. *Procedia Manufacturing*, 42, 288-295.
- [8]. Dominguez, R., & Cannella, S. (2020). Insights on multi-agent systems applications for supply chain management. *Sustainability*, 12(5), 1935.
- [9]. Heik, D., Bahrpeyma, F., & Reichelt, D. (2024). Adaptive manufacturing: dynamic resource allocation using multi-agent reinforcement learning.
- [10]. Pu, Y., Li, F., & Rahimifard, S. (2024). Multi-Agent Reinforcement Learning for Job Shop Scheduling in Dynamic Environments. *Sustainability*, 16(8), 3234.
- [11]. Bozorg Haddad, O., Mirmomeni, M., Zarezadeh Mehrizi, M., & Mariño, M. A. (2010). Finding the shortest path with honey-bee mating optimization algorithm in project management problems with constrained/unconstrained resources. *Computational Optimization and Applications*, 47, 97-128.
- [12]. Ren, L., Fan, X., Cui, J., Shen, Z., Lv, Y., & Xiong, G. (2022). A multi-agent reinforcement learning method with route recorders for vehicle routing in supply chain management. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 16410-16420.
- [13]. Ahmed, N., Ador, M. S., & Islam, S. (2020). Partner Selection for Multi-Echelon Supply Chain Using Artificial Bee Colony Algorithm. *International Journal of Applications of Fuzzy Sets and Artificial Intelligence*, 10, 65-86.
- [14]. Ouham, S., Hadi, Y., & Arifullah, A. (2020). A hybrid grey wolf optimizer and artificial bee colony algorithm used for improvement in resource allocation system for cloud technology.
- [15]. Komasilova, O., Komasilovs, V., Kvišis, A., & Zacepins, A. (2021). Model for finding the number of honey bee colonies needed for the optimal foraging process in a specific geographical location. *PeerJ*, 9, e12178.
- [16]. Niknam, T., Mojarrad, H. D., Meymand, H. Z., & Firouzi, B. B. (2011). A new honey bee mating optimization algorithm for non-smooth economic dispatch. *Energy*, 36(2), 896-908.



- [17]. Katiyar, S., Khan, R., & Kumar, S. (2021). Artificial bee colony algorithm for fresh food distribution without quality loss by delivery route optimization. *Journal of Food Quality*, 2021, 1-9.
- [18]. Yang, G., Tan, Q., Tian, Z., Jiang, X., Chen, K., Lu, Y., ... & Yuan, P. (2023). Integrated Optimization of Process Planning and Scheduling for Aerospace Complex Component Based on Honey-Bee Mating Algorithm. *Applied Sciences*, 13(8), 5190.
- [19]. Wen, X., Li, X., Gao, L., Wang, K., & Li, H. (2020). Modified honey bees mating optimization algorithm for multi-objective uncertain integrated process planning and scheduling problem. *international journal of Advanced Robotic Systems*, 17(3), 1729881420925236.
- [20]. Naderializadeh, N., Sydir, J. J., Simsek, M., & Nikopour, H. (2021). Resource management in wireless networks via multi-agent deep reinforcement learning. *IEEE Transactions on Wireless Communications*, 20(6), 3507-3523.
- [21]. <https://www.kaggle.com/datasets/dorothyjoel/us-regional-sales>