Int. J. Nonlinear Anal. Appl. In Press, 1–28
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2025.39047.5547

Mall optimization algorithm (MALL) based on human social behavior in commercial centers

Hamid Hadjiebrahim[a], Abbas Toloie Eshlaghy[a,*], Soleyman Iranzadeh[b], Alireza Pourebrahimi[c], Mohammad Reza Motadel[d]

[a]Department of Industrial Management, Science and Research Branch,Islamic Azad University, Tehran, Iran
[b]Department of Industrial Management, Tabriz Branch Islamic Azad University, Tabriz, Iran
[c]Department of Industrial Management, Karaj Branch Islamic Azad University, Karaj, Iran
[d]Department of Industrial Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran

(Communicated by Haydar Akca)

Abstract

This paper proposes a novel optimization algorithm inspired by human behavior in the shopping center called the Mall optimization algorithm (MALL), to solve challenging optimization problems. The MALL mimics the behavior of human factors present in a shopping mall including distributors of goods and services, sellers with non-commercial interests, new investors (from the group of sellers), and buyers and tourists (from the group of customers), each with specific behavior. The algorithm is evaluated on thirty well-known benchmark functions and ten challenging functions from CEC-C06 2019. The findings are compared with Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), and Teaching Learning Based Optimization (TLBO). The results indicate that the MALL algorithm can provide highly competitive results compared to these expert meta-heuristic algorithms. Statistical analysis confirms the performance of local optima avoidance, exploration, exploitation, and convergence of this proposed algorithm. This article also addresses solving the known problem of supply chain management called the PLOT (Production, Logistics, Outbound, Transportation) system. Experimental results show that the proposed algorithm can be used to solve challenging problems with unknown search spaces.

Keywords: Optimization, Metaheuristics, Swarm-intelligence, human behavior-related algorithms, Supply chain management
2020 MSC: 90C59, 65K05, 90B06

1 Introduction

Usually, solving mathematical and engineering problems requires significant resources. These resources can be powerful computers or a long time to achieve optimal results. This solving duration is challenging when the time to solve the problem increases exponentially with the enhancement of the number of variables. In such cases, managers instead of the most accurate solution with a long solving time, accept a near-optimal solution with a short solving time and use metaheuristic optimization techniques.

*Corresponding author
Email addresses: hamid@Hajiebrahim.com (Hamid Hadjiebrahim), toloie@gmail.com (Abbas Toloie Eshlaghy), dr.s.iranzadeh7@gmail.com (Soleyman Iranzadeh), dr.pourebrahimi@proton.me (Alireza Pourebrahimi), drmrmotadel@gmail.com (Mohammad Reza Motadel)

Received: September 2025 *Accepted:* November 2025

This has led to the popularity of metaheuristic methods among sciences and different fields. The main reason for this popularity is the simplicity of algorithms and their flexibility against of all kinds of problems. Also, metaheuristic algorithms operate regardless of the type of problem, do not interfere with the problems, do not need derivative, and also avoid local optimization points. All of these characteristics indicate that metaheuristic algorithms are fast tools that deliver near-optimal responses with acceptable accuracy, high precision and ease of use [**?**].

Metaheuristic algorithms have a systematic behavior that they optimize by generating random solutions. This, along with preventing early convergence, avoiding the local trap, and having a separate performance from the problem itself, causes it to act as a black box and to have a response by creating a balance between exploration and exploitation, and searching effectively and efficiently in the search space. The exploration phase seeks a good solution in the search space, which is also called global search or random search. This leads to diversification of searches and avoids the local optimal trap. However, in the exploitation phase, we want to modify the solution and avoid large steps in the solution space with the hope of achieving a more optimal solution. This stage is called local search or intensification [7, 71].

Here we have to introduce the “no-free lunch theorems” for optimization. In these theories, it has been proven that, logically, no metaheuristic algorithm can optimize all the problems properly. This means that a specific metaheuristic may provide promising solutions to a set of problems, but the same algorithm performs poorly on other problems [97]. Consequently, each metaheuristic algorithm can have a unique feature for itself. these theorems have caused studies on new metaheuristic algorithms to develop significantly. This motivated us to study a new metaheuristic optimization algorithm inspired by shopping malls.

Researchers pay a lot of attention to the features of metaheuristic algorithms and classify them based on these characteristics. This classification can be based on the number of solutions (based on single or multiple solutions), memory status, neighborhood structure, dynamic or static fitting function, inspired by nature, population-based, or the behavior of algorithms[29, 14, 2]. Algorithms are divided into evolutionary algorithms, swarm intelligence, physic base algorithms, and human behavior-related algorithms classes based on the characteristics of their behavior.

Some algorithms are purely inspired by human behavior. Purity here means that the behavior of the algorithm is different from other classes, including swarm intelligence, and follows only human behavior. Humans have their ways of performing activities and are always growing and learning, and such algorithms are based on social or emotional behaviors, competition, education, and learning [90]. The well-known algorithms of this class are the Teaching–Learning-Based Optimization algorithm (TLBO) [77], Brain Storm Optimization (BSO) [86], and League Championship Algorithm (LCA) [48].

Further, Section 2 presents a literature review of metaheuristic algorithms. Section 3 describes the proposed MALL algorithm. The discussion and experimental results regarding the benchmark functions and the application of the algorithm in a real problem are presented in sections 4 and 5, and finally, in section 6, the results of this research and suggestions for future studies are presented.

2 literature review

One of the most basic classifications of metaheuristic algorithms is based on the number of solutions produced at one time. In this type of classification, algorithms are divided into two categories: single-solution algorithms and population-based algorithms. "Single-solution" algorithms are also known as "trajectory methods," which, at any time, with local search and using a selected solution, try to improve this answer in its neighborhood [16, 92]. The most famous single-solution algorithms such as: Simulated Annealing (SA) [54, 17], Tabu search (TS) [40], Iterated Local Search (ILS) [58], Guided Local Search (GLS) [94], Greedy Randomized Adaptive Search Procedures (GRASP) [34, 35] and Variable Neighborhood Search (VNS) [63, 64]. On the other hand, Population-based algorithms are usually distinguished by their behavior, which can be categorized into four primary types: evolutionary algorithms, swarm intelligence, physic-base algorithms, and human behavior-related algorithms.

Evolutionary algorithms take inspiration from the natural process of evolution. The Genetic Algorithm (GA), is one of the most well-known algorithms introduced by Holland [44], is a prominent example of this type of algorithm that is based on the principles of Darwinian evolution. Some other evolutionary algorithms include Genetic programming (GP) [55], Evolutionary Programming (EP) [103], Differential Evolution (DE) [89], Evolution Strategy (ES) [80, 41], Differential Search (DS) [18], Backtracking Search optimization Algorithm (BSA) [19], Biogeographical-Based Optimization (BBO) [88], Stochastic Fractal Search (SFS) [83], Synergistic fibroblast optimization (SFO) [26].

Another type of algorithm is based on physics which imitates Laws of Physics. Some of the most well-known of them include: Gravitational Search Algorithm (GSA) [78], Gravitational Local Search Algorithm (GLSA) [96],

Charged system search (CSS) [52], Galaxy-based search algorithm (GbSA) [85], Big bang–big crunch (BBBC) [32], Artificial Chemical Reaction Optimization Algorithm (ACROA) [6], Electro-Magnetism Optimization (EMO) [20], Electro-Search algorithm (ES) [91], Black Hole (BH) [43], Thermal exchange optimization (TEO) [50], Small-world optimization algorithm (SWOA) [30], Ray Optimization (RO) [51], Find-Fix-Finish-Exploit-Analyze (F3EA) [49] and Curved space optimization (CSO) [65].

For example, the F3EA algorithm tries to imitate targeting and shooting used in battle, a process during which suitable targets or facilities are selected for destruction. In the process of targeting, the category of goals or resources that the enemy cannot lose and that appear to produce a desirable outcome for achieving the goals is analyzed and prioritized. The algorithm initially identifies targets, including people, locations, and elements, then identifies the exact location of the target on the battlefield and tracks it to shoot the target at the next stage. Further information is collected from the target area, and the level of target sensitivity and damage caused are extracted. Finally, the results of the attacks are analyzed. The algorithm is designed based on this approach, imitating the targeting cycle presented by General McChrystal in the Iraq and Afghanistan wars [36, 49].

The third class is metaheuristic algorithms based on swarm intelligence, which is inspired by the behavior of groups, flocks, herds, and colonies in nature, and the search factors, apart from individual operators, are also dependent on their collective experience. The most popular swarm intelligence algorithm is the particle swarm optimization (PSO), which was developed with inspiration from the social behavior of birds [53]. Another one of the most used algorithms in this class is ant colony optimization (ACO), in which, based on social behavior, ants find the shortest path between nest and food [15]. The most well-known metaheuristic algorithms of this class include Artificial bee colony (ABC) [13], Wasp Swarm Optimization (WSO) [73], Marriage in honey Bees Optimization (MBO) [1], Bee Collecting Pollen Algorithm (BCPA) [59], Butterfly optimization algorithm (BOA) [8], Fruit fly Optimization Algorithm (FOA) [72], Termite Algorithm (TA) [82], Firefly Algorithm (FA) [99], Bat Algorithm (BA) [100], Dolphin Partner Optimization (DPO) [87], Krill Herd (KH) [37], Fast Artificial Fish Swarm Algorithm (FAFSA) [12], Monkey search (MS) [70], Cuckoo Search (CS) [102], Bird Mating Optimizer (BMO) [10], Harmony Search (HS) [57], Grenade Explosion Method (GEM) [5], Emperor Penguins Colony (EPC) [42], Emperor Penguin Optimizer (EPO) [25], Squirrel Search Algorithm (SSA) [46], Spotted Hyena Optimizer (SHO) [24] and Salp Swarm Algorithm (SSA) [61].

For example, the Ant Colony Optimization (ACO) presented by Marco Dorigo et al. was an imitation of an ant colony. By observing the social behavior of the ant, they realized that instead of focusing on individual survival, it focused on the survival of the colony. It also chooses the shortest route to transport food to the nest. Ants first randomly discover the surroundings of their nest and leave a trail of pheromones on the ground as they move. Ants feel the smell of pheromone, and when choosing a path, they choose the path with the highest concentration of pheromone smell. As soon as the ant finds food, it evaluates the quantity and quality of the food and takes some of it to the nest. During the return trip, the amount of pheromone that the ant leaves on the ground depends on the quantity and quality of the food. Pheromone trails guide other ants to the food source. Also, pheromone evaporation prevents ants from being caught in the optimal trap [15, 28].

The last group of behavioral classifications of metaheuristic algorithms are algorithms related to human behavior. In this class, algorithm design has addressed human social behavior or emotional conflicts, education and learning, and even human competition. Social cognitive optimization algorithm (SCOA) [21] and Social-Based Algorithm (SBA) [75] are imitations of human social-emotional behavior, or League championship algorithm (LCA) [48] and Volleyball Premier League algorithm (VPL) [66] Developed based on human competition. Also, Teaching–learning-based optimization (TLBO) [77] originated from the learning process. Other algorithms designed based on human behavior include Human-inspired algorithms (HIA) [104], Brain Storm Optimization algorithm (BSO) [86], Anarchic Society Optimization (ASO) [4], Soccer League Competition algorithm (SLC) [69], Exchange Market Algorithm (EMA) [39], Cultural algorithms (CA) [81], Group Counseling Optimization (GCO) [31], Jaya Algorithm (Jaya) [76], Human behavior-based optimization (HBBO) [3], Society and Civilization Algorithm (SCA) [79], Seeker Optimization Algorithm (SOA) [22], Imperialist Competitive Algorithm (ICA) [11], Election campaign optimization algorithm (ECO) [60], Social Emotional Optimization Algorithm (SEOA) [98], Cohort intelligence (CI) [56], Passing vehicle search (PVS) [84], Football game inspired algorithm (FGA) [33], Dynastic Optimization Algorithm (DOA) [95], Human urbanization algorithm (HUA) [38], Heap-based Optimizer (HBO) [9], social ski-driver (SSD) [93] and Gaining-sharing knowledge based algorithm (GSK) [67].

For example, the "Gaining-sharing knowledge-based algorithm" (GSK) is designed based on the philosophy of acquiring and sharing knowledge throughout human life. At the beginning of birth, humans are in contact with a small population, such as family, relatives, and neighbors, and acquire or share their knowledge through them. At this stage, although a person has very little experience but is interested in passing their information to others, including people outside their communication network, they do not have sufficient competence to diagnose and classify good

and bad. On the other hand, during adulthood, due to interaction with larger networks of people, knowledge increases through personal experience, learning from the experiences of others, or imitating them. This means that at this stage, each person acquires a high ability to judge, think, and classify different people into good, bad, and average classes. This algorithm uses this logic to find an optimal solution [67].

In Table 1, research papers related to literature are listed alphabetically. As can be seen, many optimization techniques have been developed based on human behavior. However, the general buying and selling behavior of humans is less concerned. This motivates us to develop a new algorithm based on human behavior in the shopping mall center. In the following, its ability and effectiveness are discussed.

Table 1: A review of past published literature.

Algorithm	References	SS*	PB*	Ev*	Ph*	SI*	HB*
Anarchic Society Optimization (ASO)	[4]		✓				✓
Ant Colony Optimization (ACO)	[15]		✓			✓	
Artificial bee colony (ABC)	[13]		✓			✓	
Artificial Chemical Reaction Optimization Algorithm (ACROA)	[6]		✓		✓		
Backtracking Search optimization Algorithm (BSA)	[19]		✓	✓			
Bat Algorithm (BA)	[99]		✓			✓	
Bee Collecting Pollen Algorithm (BCPA)	[59]		✓			✓	
big bang–big crunch (BBBC)	[32]		✓		✓		
Biogeography-based optimization (BBO)	[88]		✓	✓			
Bird Mating Optimizer (BMO)	[10]		✓			✓	
Black Hole (BH)	[43]		✓		✓		
Brain Storm Optimization algorithm (BSO)	[86]		✓				✓
Butterfly optimization algorithm (BOA)	[8]		✓			✓	
Charged system search (CSS)	[52]		✓		✓		
Cohort intelligence (CI)	[56]		✓				✓
Cuckoo Search (CS)	[102]		✓			✓	
Cultural algorithms (CA)	[81]		✓	✓			✓
Curved space optimization (CSO)	[65]		✓		✓		
Differential Evolution (DE)	[89]		✓	✓			
Differential Search (DS)	[18]		✓	✓			
Dolphin Partner Optimization (DPO)	[87]		✓			✓	
Dynastic Optimization Algorithm (DOA)	[95]		✓				✓
Election campaign optimization algorithm (ECO)	[60]		✓				✓
electro-magnetism optimization (EMO)	[20]		✓		✓		
Electro-Search algorithm (ES)	[91]		✓		✓		
Emperor Penguin Optimizer (EPO)	[25]		✓			✓	
Emperor Penguins Colony (EPC)	[42]		✓			✓	

* SS: Single Solution, PB: Population Base, Ev: Evolutionary, Ph: Physical, SI: Swarm intelligence, HB: Human Behavior

3 Mall optimization algorithm (MALL)

In this section, the basis of the algorithm's inspiration from the shopping mall is discussed first, and then its mathematical model is presented.

3.1 Basis and inspiration

A shopping center or mall is a large, indoor commercial and administrative building that is full of many shops, stores, entertainment centers, and service centers. Successful shopping malls, in addition to large stores, include specialized markets and festivals, play and entertainment centers, and even libraries, movie and theater centers, and other service centers. The main agents in the mall are customers and sellers, whose objectives are classified according to Fig. 1.

Sellers in the mall are offering their products or services in tight competition. Sellers' behavior is generally greedy, and they choose the best places to sell products. Meanwhile, some sellers choose the worst points due to the type of activity or non-commercial goals and interests. Also, some sellers are looking for investment opportunities. Here, the set of sellers who have chosen the best and worst places are referred to as selected sellers, and those who are looking for investment opportunities are called new investors.

Generally, customers are looking to buy goods, receive services, rest, enjoy entertainment, and even eat food. From this perspective, customers are divided into two major categories: buyers and tourists. Buyers are usually looking for

Table 1: A review of past published literature (Continue).

Algorithm	References	SS*	PB*	Ev*	Ph*	SI*	HB*
Evolution Strategy (ES)	[80, 41]		✓	✓			
Evolutionary Programming (EP)	[103]		✓	✓			
Exchange Market Algorithm (EMA)	[39]		✓				✓
Fast Artificial Fish Swarm Algorithm (FAFSA)	[12]		✓			✓	
Find-Fix-Finish-Exploit-Analyze (F3EA)	[49]		✓		✓		
Firefly Algorithm (FA) (Yang, 2010a)	[99]		✓			✓	
Football game inspired algorithm (FGA)	[33]		✓				✓
Fruit fly Optimization Algorithm (FOA)	[72]		✓			✓	
Gaining-sharing knowledge based algorithm (GSK)	[67]		✓		✓		
Galaxy-based search algorithm (GbSA)	[85]		✓		✓		
Genetic algorithms (GA)	[44]		✓	✓			
Genetic programming (GP)	[55]		✓	✓			
Gravitational Local Search Algorithm (GLSA)	[96]		✓		✓		
Gravitational Search Algorithm (GAS)	[78]		✓		✓		
Greedy Randomized Adaptive Search Procedure (GRASP)	[34, 35]	✓					
Grenade Explosion Method (GEM)	[5]		✓			✓	
Group Counseling Optimization (GCO)	[31]		✓				✓
Guided Local Search (GLS)	[94]	✓					
Harmony Search (HS)	[57]		✓			✓	
Heap-based Optimizer (HBO)	[9]		✓				✓
Human behavior-based optimization (HBBO)	[3]		✓				✓
Human urbanization algorithm (HUA)	[38]		✓				✓
Human-inspired algorithms (HIA)	[104]		✓				✓
Imperialist Competitive Algorithm (ICA)	[11]		✓				✓
Iterated Local Search (ILS)	[58]	✓					
Jaya Algorithm (Jaya)	[76]		✓				✓
Krill Herd (KH)	[37]		✓			✓	
League championship algorithm (LCA)	[48]		✓				✓
Marriage in honey Bees Optimization (MBO)	[1]		✓			✓	
Monkey search (MS)	[70]		✓			✓	
Particle swarm optimization (PSO)	[53]		✓			✓	
Passing vehicle search (PVS)	[84]		✓				✓
Ray Optimization (RO)	[51]		✓		✓		
Salp Swarm Algorithm (SSA)	[61]		✓			✓	
Seeker Optimization Algorithm (SOA)	[22]		✓				✓
Simulated Annealing (SA)	[54, 17]	✓			✓		
Small-world optimization algorithm (SWOA)	[30]		✓		✓		
Soccer League Competition algorithm (SLC)	[69]		✓				✓
Social cognitive optimization algorithm (SCOA)	[21]		✓				✓
Social Emotional Optimization Algorithm (SEOA)	[98]		✓				✓
social ski-driver (SSD)	[93]		✓				✓
Social-Based Algorithm (SBA)	[75]		✓				✓
Society and Civilization Algorithm (SCA)	[79]		✓				✓
Spotted Hyena Optimizer (SHO)	[24]		✓			✓	
Squirrel Search Algorithm (SSA)	[46]		✓			✓	
Stochastic fractal search (SFS)	[83]		✓	✓			
Synergistic fibroblast optimization (SFO)	[26]		✓	✓			
Tabu Search (TS)	[40]	✓					
Teaching–learning-based optimization (TLBO)	[77]		✓				✓
Termite Algorithm (TA)	[82]		✓			✓	
thermal exchange optimization (TEO)	[50]		✓		✓		
Variable Neighborhood Search (VNS)	[63]	✓					
Volleyball Premier League algorithm (VPL)	[66]		✓				✓
Wasp Swarm Optimization (WSO)	[73]		✓			✓	

* SS: Single Solution, PB: Population Base, Ev: Evolutionary, Ph: Physical, SI: Swarm intelligence, HB: Human Behavior

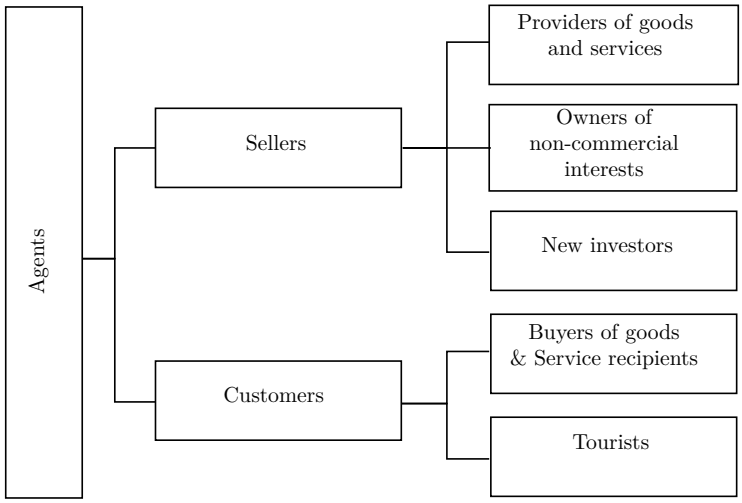

Figure 1: Classification of the objectives of the main agents active in the shopping mall.

the best purchase, and as a result, they choose the sellers who offer the best price as their target. This target is one of the selected sellers. On the way to the selected sellers, buyers inquire about prices from other sellers. However, tourists are not looking for the best purchase, and their target is not the best price. The target of each tourist is to spend time in the mall and move toward random points.

The duration of the presence of customers in the mall has a significant relationship with the individual characteristics of customers, which are independent of other customers and sellers. These individual characteristics are static factors that do not change during the time spent in the mall. These characteristics include the gender, age, financial status, and free time of customers. Women spend more time in the mall than men. Also, younger customers, those with rich financial status, or those with free time, spend more time in the shopping center. As a result, the number of customers' price inquiries is related to the duration of the customer's presence in the mall, which itself has a significant dependence on the individual characteristics of the customers.

Each of the customers has moved toward their target (selected seller) during the time they are in the mall, and the closer they get to the target, the more price inquiries are taken, which means that as they get closer to the target, the length of the steps becomes smaller. While moving toward the target, advertisements from other selected sellers can influence the customer's path. Once the duration of the presence of customers in the mall has ended, the customer leaves the mall, meaning that the buyers have purchased or the tour of the tourists has ended. While moving towards the target, customers consider several opportunities to find a better price for themselves and eliminate those opportunities as a penalty when their query price does not improve. If the whole opportunity ends, they change their target and move toward a new target. Also, if all selected sellers are targeted as buyers but lose all opportunities, the buyer becomes a tourist.

Customers entering the mall usually try to get in from the nearest entrance to their target or enter the system in a place where there is more gathering of auctions.

3.2 Mathematical model and algorithm

This subsection describes the behavior of agents in shopping malls, including sellers' and customers' entry to the mall, moves, opportunities and actions, and new investors' actions. Then the proposed MALL algorithm is outlined.

Step 1 – The first selected sellers: The selected sellers, for the first time, are chosen randomly in the justified area.

Step 2 – Customer entry to the mall: In this step, the individual characteristics of the customer are determined, and based on that, the number of price inquiries is calculated. Each customer chooses a destination and enters the mall from the closest entrance to their target. Each of these sub steps is as follows:

Step 2-1- Determining individual characteristics: Each of the customers has a specific characteristic, and according to Table 2, the characteristics and a sequential number ratio of those characteristics are randomly determined.

Step 2-2- Determining the number of price inquiries: First, the duration of the customer's journey is determined based on their characteristics according to Eq.(3.1), and then the number of price inquiries is specified according to

Table 2: Individual characteristics of the customer and its quantitative ratio.

Characteristic	Option	Quantitative ratio
Gender	Male	1
	Female	0
Age	Old	1
	Young	0
Financial Situation	Poor	2
	medium	1
	rich	0
Free Time	Busy	1
	Free	0
Motivation	Buyer	1
	Tourist	0

Eq.(3.2). In this equation the *RandNormal* parameter is a random number with a normal distribution, which has a mean of 170 and a standard deviation of 30. Also, the parameter *RandInteger* is a random integer number with a uniform distribution between zero and three. The numerical values of the other parameters of Age, Free Time, Financial Situation, and Gender are determined based on Table 2 and randomly with a uniform distribution according to the states listed in this table.

$$\begin{aligned} VisitDuration = {} & RandNormal - 26 \times Age - 10 \times Free\ Time \\ & - 29 \times Financial\ Situation - 45 \times Gender \end{aligned} \tag{3.1}$$

$$Inquiries = Max\left(7, \left\lfloor \frac{VisitDuration}{8 + RandInteger} \right\rfloor\right) \tag{3.2}$$

Step 2-3- Determining the destination: In this step, each buyer randomly selects one of the selected sellers as their target. Since sellers who have better fitness will be more desirable to buyers, the roulette wheel is used in determining the random destination based on the degree of fit. Also, each of the tourists chooses a random point in the justified area as their destination.

Step 2-4- Customer entry: The distribution of customers in the problem space is determined randomly based on their intended destination and the distance from the destination to the nearest seller. Therefore, the customer's entry is a random point with a normal distribution with an average of μ and a standard deviation of σ, determined according to Eq.(3.3).

$$\begin{aligned} X_{Customer} &= \underset{\mu,\sigma}{RandNormal}, \\ \mu &= X_{Destination}, \\ \sigma &= \frac{1}{3} \times min\{absX_{Destination} - XSeller | Destination \neq Seller\} \end{aligned} \tag{3.3}$$

In Eq.(3.3), the X vector is the position of an agent in the mall. Therefore, $X_Customer$ and $X_Destination$, are the positions of the customer and the determined destination, respectively and X_Seller is a nearest seller position, which was not the destination. The RandNormal parameter is a random number with a normal distribution, which has a mean of μ and a standard deviation of σ. In Fig. 2, the distribution of new customers entering the mall is presented. If the customer choose Seller 2 as destination, nearest seller is Seller 3 then the dispersion of customer arrival is a third of distance 1.

Step 5-2- Distance to the destination: The customer's distance vector to the destination is calculated according to Eq. (3.4).

$$Distance = X_{Destination} - X_{Customer} \tag{3.4}$$

Step 3 – Movement (exploration): In this step, customers move towards their destinations and search the problem space.

Step 3-1- Calculation of steps: The calculation of each step for customers' movement is based on the distance to the destination, the number of steps, the number of steps that have a better or worse result, and the fit of other sellers,

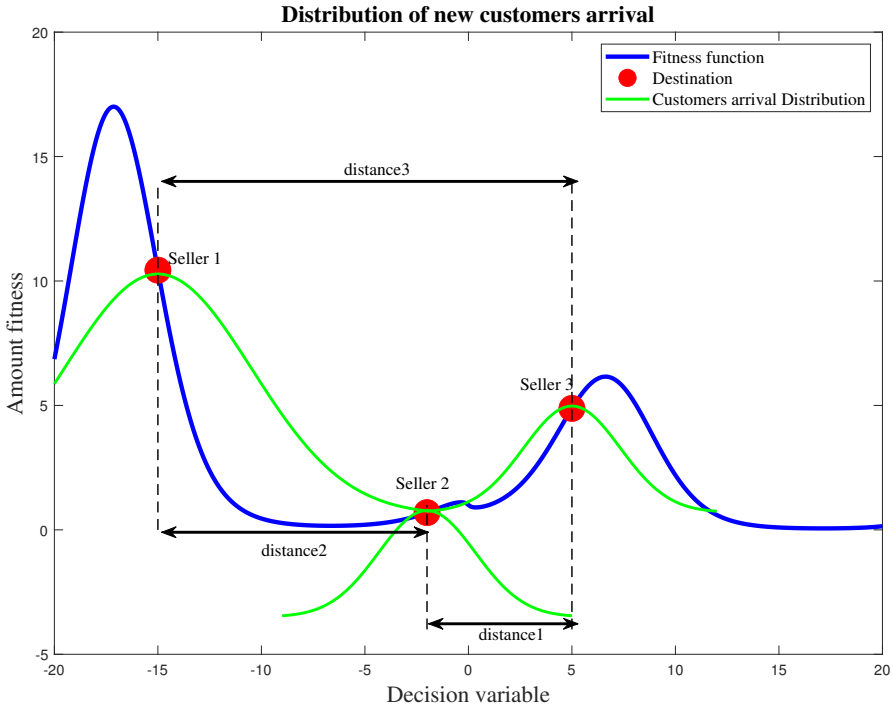

Figure 2: Distribution of new customer arrivals.

according to Eq. (3.5).

$$Step = StepSize \times \underset{0\sim 2}{UnifRand} \times \Big[\alpha \times Distance + (1-\alpha) \times \sum_{i} \big((X_{Seller_i} - X_{Customer}) \times Adv_i\big)\Big] \tag{3.5}$$

$$StepSize = StepController \times \left[\exp\left(\frac{Inquiries \times (s - NBS - 1)}{21.4 - 3.529 \times Inquiries}\right) - \exp\left(\frac{Inquiries \times (s - NBS)}{21.4 - 3.529 \times Inquiries}\right)\right] \tag{3.6}$$

$$StepController = 1 + \frac{NBS}{Opportunities} - \frac{NGS}{Inquires + 1} \tag{3.7}$$

$$Adv_i = \left|\frac{SellerObj_i - WorsObj}{BestObj - WorsObj}\right| \tag{3.8}$$

In Eq. (3.5), the necessary steps for the movement of customers in each price inquiry are calculated. The amount of $StepSize$ is calculated on the basis of Eq. (3.6) and ensures that the closer the customer gets to the target, the shorter the customer's step length. Fig. 3 shows the change in step length, excluding the control of step and random number. $UnifRand$ is a random number with a uniform distribution between zero and two. The α coefficient controlled the impacts of advertisement on customer trajectory, which was 0.9 on the first day and decreases every day. X_{Seller_i} is the position vector of the i^{th} seller, and $X_{Customer}$ is the current position vector of the customer. Adv_i is the attractiveness coefficient of the i^{th} seller's advertisement, which is calculated according to Eq. (3.8).

In Eq. (3.6), $StepController$ is the step length controller coefficient, and Inquiries is the number of price inquiries determined by Eq. (3.2). 's' is the number of steps taken by the customer, which is at least one and at most equal to Inquiries. NBS counts the number of steps that have a worse fit than the previous step.

Also in Eq. (3.7), NGS counts the number of steps that have a better fit than the previous step, and the Opportunities

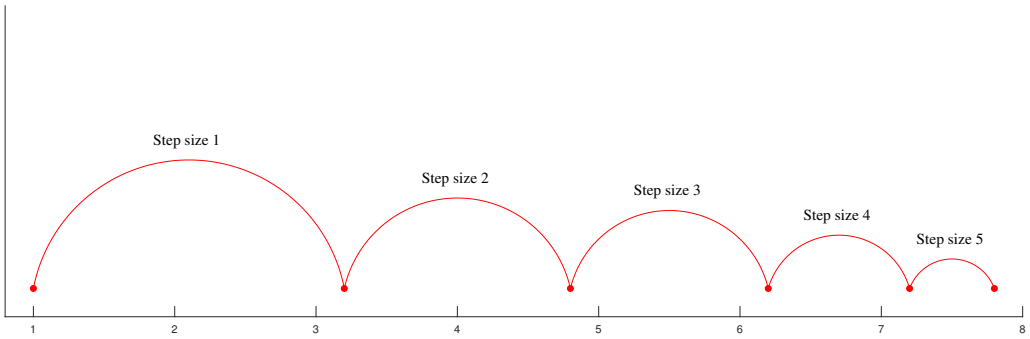

Figure 3: The length of the customer's steps excluding step control and random number.

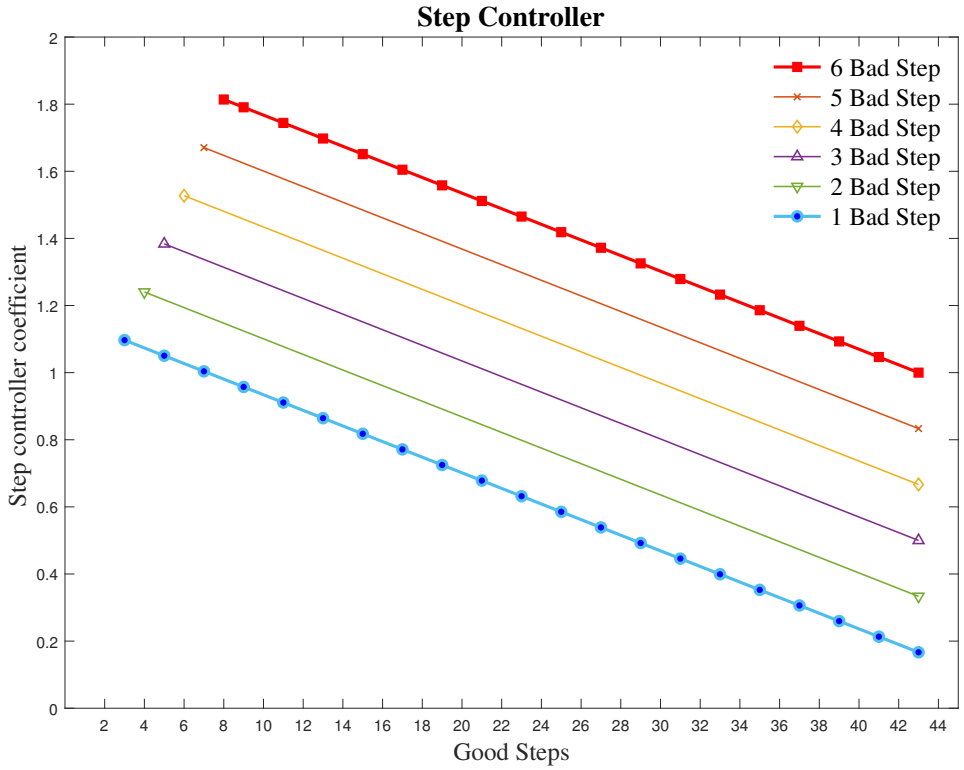

Figure 4: The effect of the step controller coefficient based on number of good and bad steps.

parameter is the opportunities that the customer considers for themselves. Fig. 4, shows the effect of the step controller coefficient ($StepController$). As the number of bad steps (NBS) increases, the step length becomes larger to move away from bad fits, and as the number of good steps (NGS) increases, the step length becomes smaller to move towards exploiting a better solution.

Eq. (3.8) calculates the advertising attractiveness coefficient (Adv_i) of other sellers, where $SellerObji$ is the fit of the i_{th} seller's position and $BestObj$ and $WorsObj$ are the best and worst fit of sellers, respectively. Based on this, the attractiveness of the worst seller is always zero and the best seller is one, which affects the attraction of customers.

Step 3-2- New Position: Here, according to Eq. (3.9), the customer moves toward the destination by the length of the step and determines the fit with the new position. In this equation, $X_(i+1)$ is the customer's new position, X_i is the customer's previous position, and $Step$ is the step vector.

$$X_{i+1} = X_i + Step \tag{3.9}$$

Step 4 – Opportunities and Actions: Buyers compare the fitness of their current decision step with the previous step every time they inquire about the price, and if the conditions get worse, they lose one of their opportunities as a

penalty. But if things get better, consider another opportunity as a reward for themselves. When all the considered opportunities are lost, they choose a new seller as the destination. If the buyer loses all of its opportunities for all the sellers, it becomes a tourist and moves toward a random destination.

Step 5 – New investor search (exploitation): New investors are looking for a better fit position around selected sellers. According to Eq. (3.10), new investors search the space around the selected sellers, and as the number of days of system implementation increases, the search rate of new investors also increases.

$$X_{NewInvestor} = \begin{cases} X^j_{Seller} + & SellerDistance \times \exp(\beta \times \theta) \times \cos(\theta), \\ & \text{if} \quad \text{mod}\ (j,2) > 0. \\ X^j_{Seller} + & SellerDistance \times \exp(\beta \times \theta) \times \sin(\theta), \\ & \text{if} \quad \text{mod}\ (j,2) = 0. \end{cases} \tag{3.10}$$

$$SellerDistance = \sqrt{\sum_j (X^j_{Seller} - X^j_{nearest})^2} \tag{3.11}$$

In Eq. (3.10), $X_NewInvestor$ is the new investor position. Also, X^j_{Seller} is the position vector of each selected seller, whose components are indicated by j, as their odd components are multiplied by cosine θ and their even components are multiplied by sine θ. $SellerDistance$ is the direct distance between the selected seller and the nearest selected seller, which is calculated according to Eq. (3.11). The θ parameter is the new investor's search angle, which starts from zero and increases randomly. The β coefficient is a random number with a uniform distribution between the negative one tenth and the positive one hundredth. In Fig. 5, it shows the search around the selected seller with three decision variables, and in the cases of β equal to one hundredth and also β equal to negative one tenth. By increasing the amount of θ, the number of rotations around the seller increases, while the search with a negative β coefficient is convergent and a positive β coefficient is divergent.

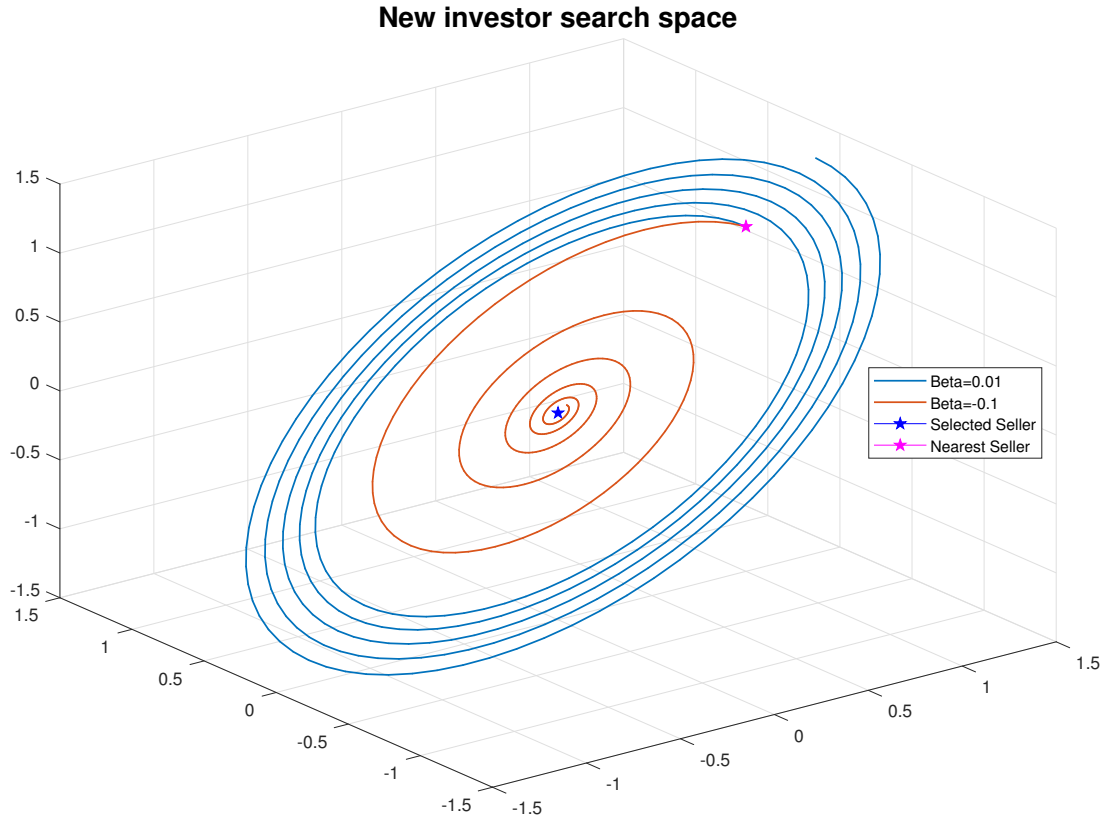

Figure 5: New investor search space with different beta coefficients.

Step 6 – Determining the selected sellers: In this step, we choose the selected sellers based on the inquiries. *Step 6-1- Today's Best Seller*: The seller who had the best fitness during this iteration.

Step 6-2- Global Best Seller: The seller who had the best fitness during all the past iterations.
Step 6-3- Global Worst Seller: The seller who had the worst fitness during all the past days.
Step 6-4- New Investor: The best fitness for a new investor is based on *step 5*.

Step 7 – Algorithm stop condition: If the stop conditions are met, including the number of iterations, the algorithm is stopped, and the global best seller is selected as the best fitness. Otherwise, it goes to *step 2*, and the operation continues.

The pseudocode of the MALL algorithm is presented in Fig. 6 and its flowchart in Fig. 7.

3.3 Performance analysis of MALL algorithm

The MALL optimization algorithm consists of simple steps that move closer to the optimal point with each iteration. Here, to analyze the optimization performance of this algorithm, the sphere objective function is selected, and in Fig. 8, its contours are drawn. In this function, we are looking for minimization, and the optimal point is in the center of the sphere with coordinates (0 and 0).

At first, a series of parameters and preparations are made in the algorithm, where the advertising coefficient (1-α) is set at 0.005 and there are two decision variables in the range between [-3, 3] and 5 opportunities for each customer. We also have three customers; two of them are buyers, and one is a tourist.

Algorithm MALL

Begin
Generate initial population of sellers and determine fitness, Nearest Seller and Attraction
Define parameters (Number of decision variables, Opportunities, Reward & Punishment)
Generate initial population of Customers and determine fitness
While stop condition not met ***do***
  For each customer
    Define Customers Characteristics (Table 2)
    Determine the number of inquiries (Eqs. (3.1) ,(3.2))
    Set destination (one of sellers by roulette wheel or random location)
    Generate customer populations around the destination (Eq. (3.3))
    Determine the distance to the destination (Eq. (3.4))
      For each inquiries
        Calculate the step size (Eqs. (3.1) and (3.5) to (3.8))
        Move towards destination (Eq. (3.9)) and determine new fitness
        Update Opportunity
        If Opportunity < 1 ***then*** Set new destination ***End if***
      End for
  End for
New investor tour around selected sellers (Eqs. (3.10), (3.11))
Determine selected sellers (Global Best, Global Worst, Today Best, New investor)
End While
Output the best solution found

Figure 6: Pseudocode of the MALL algorithm.

According to Fig. 8a, the first step of the algorithm is implemented, and the sellers are randomly selected in the mall. Sellers who are closer to the center of the circle have a better fit, and here, seller 2 has better fitness.

In the next step of the algorithm, according to Fig. 8b, customers enter the mall randomly in the range of their destination. As is clear in this figure, the destination of Customer 1 is Seller 3, and the destination of Customer 2 is Seller 2. Sellers with better fitness are more likely to be chosen by customers. Here, Customer 3, as a tourist, has a random destination.

In the third and fourth steps of the algorithm, customers move towards their destinations and, based on their fit, modify their opportunities and their destination. According to Fig. 8c, Customer 1 moved toward its destination (that is, Seller 3), and since after a few initial price inquiries, the fitness has decreased, it has changed its destination toward Seller 1. The same thing happened to customer 3. However, Customer 2 has achieved a better fit in every price inquiry and has always taken smaller steps to reach its destination. Deviation from the path evident in customer 2 inquiries has occurred based on other sellers' advertisements. After completing the inquiries, customers exit the system. Here, changing the destination, and increasing or decreasing the length of the customers' steps based on their experiences, helps to discover better answers, and by using random coefficients, the variety of answers also increases.

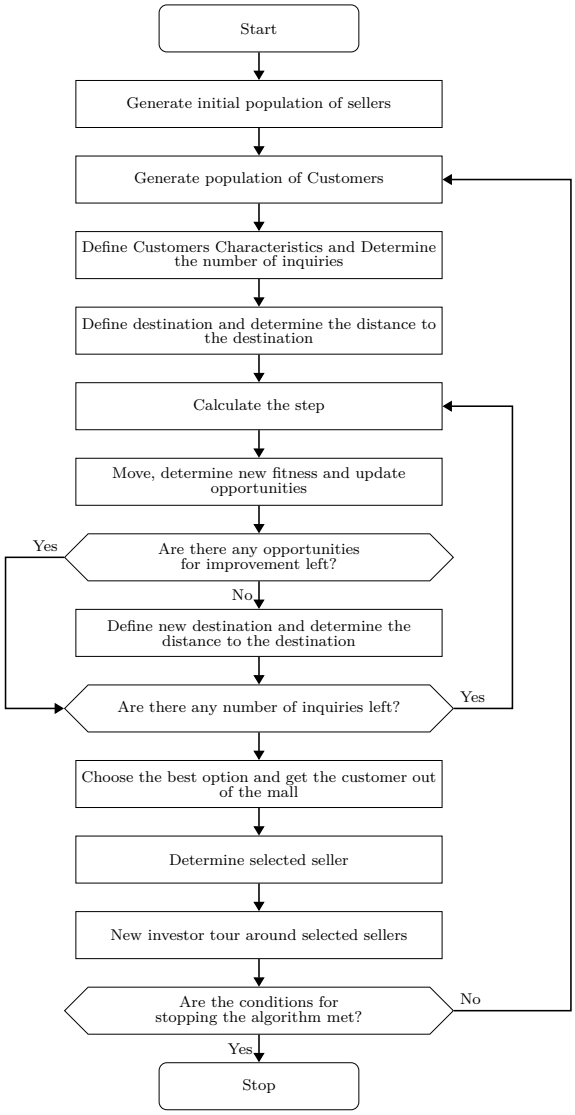

Figure 7: Flowchart of the MALL algorithm.

In the fifth step, according to Fig. 8d, the search for new investment in the neighborhood of sellers is done through a logarithmic spiral. The convergence or divergence coefficient of the search is random, where the search around seller 1 is divergent and the other sellers are convergent. This step guarantees the extraction of better solutions for problems.

In the first iteration of the sixth step, the selected sellers are determined according to Fig. 8e. Selected sellers are the basis for customer entry in subsequent iterations. In this way, in the next iteration, customers will explore the system randomly and within the scope of their destination seller. As shown in Fig. 8f, in the second iteration of the algorithm, the best global seller has significantly improved compared to the previous iteration and is closer to the center of the circle.

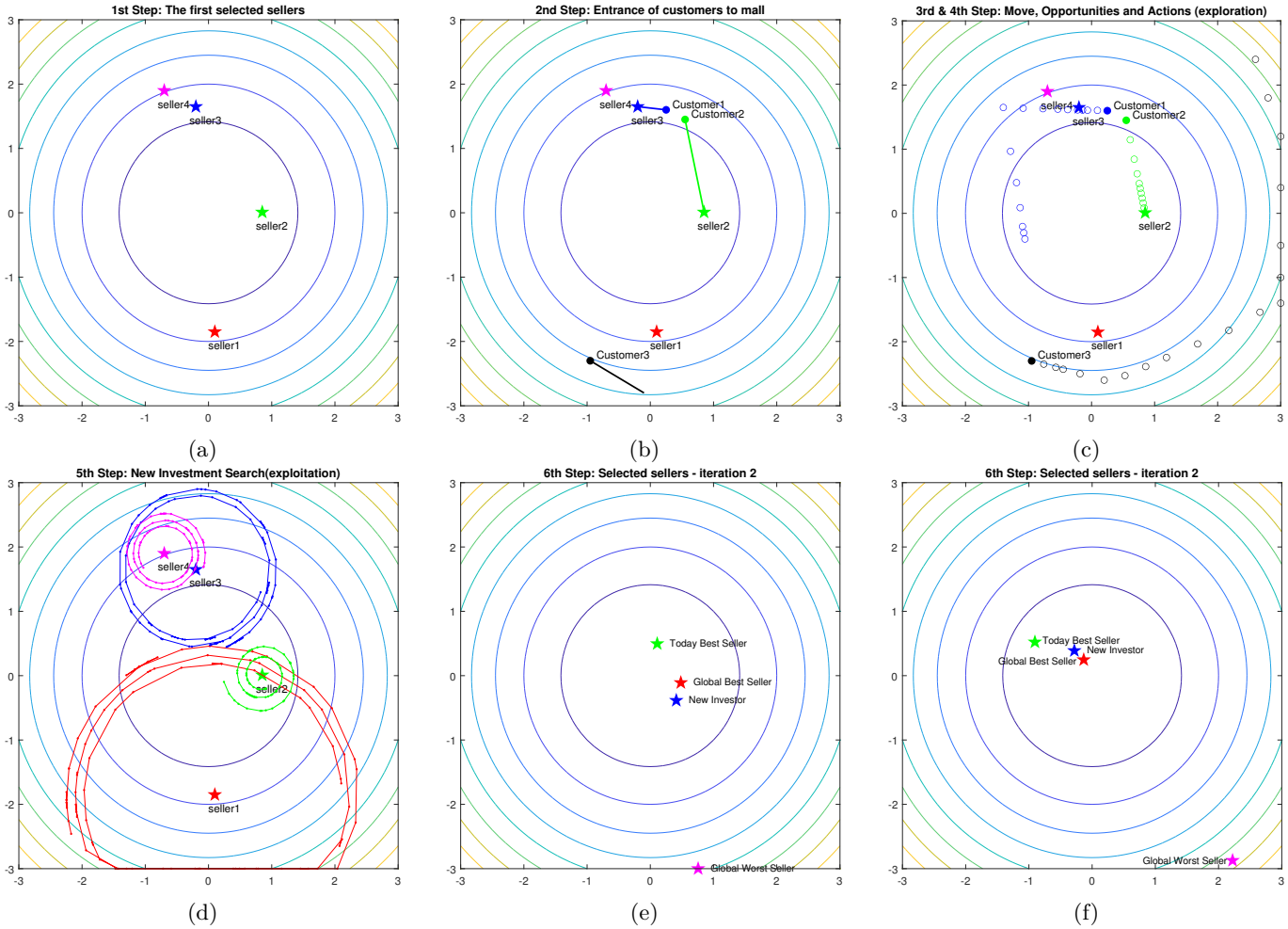

Figure 8: Analysis of the optimization performance of MALL algorithm.

In the following, some points that theoretically enable the MALL algorithm to solve optimization problems are noted:

- Communication between iterations is established only by selected sellers, and other agents do not keep information records after exiting the system (no memory and solution diversity).
- Tourists ensure that local optima are avoided by randomly searching and moving towards a random target.
- The best and worst sellers always maintain the domain of the search space.
- In cases of worsening of the solution, customers search with larger steps to find better exploratory solutions.
- In cases of improving the solution, customers search with smaller steps to find better exploitative solutions.
- The α coefficient creates inertia to keep the customer on track toward their destination in cases of the high attractiveness of other sellers. This coefficient is reduced during each iteration to make the extraction solution more diverse and desirable.
- Advertisements and the attractiveness of other sellers increase greedy searches.
- Today's best seller, maintains the best performance of customers for exploitation.
- The new investor searches in a logarithmic spiral around each selected seller, with the radius of the distance to the nearest selected seller, and exploits a more optimal solution.
- If the β coefficient is positive, the search is done in a divergent manner; otherwise, it is done in a convergent way around the selected seller.
- By increasing the theta coefficient, more solutions around the selected seller are extracted, and with the increase in the number of iterations of the algorithm, the limit of the theta coefficient also increases.

Table 3: Benchmark functions.

ID	Benchmark function	Type				Formula	Dim*	Range	Opt*
		Mm*	Un*	Sp*	NS*				
$f1$	Sphere	✓		✓		$f(x)=\sum_{i=1}^{n} x_i^2$	30	$[-100,100]$	0
$f2$	Beale		✓		✓	$f(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2^2)^2$ $+(2.625-x_1+x_1x_2^3)^2$	30	$[-4.5,4.5]$	0
$f3$	Cigar		✓		✓	$f(x)=x_1^2+10^6\sum_{i=2}^{n} x_i^2$	30	$[-100,100]$	0
$f4$	Step		✓	✓		$f(x)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2)$	30	$[-100,100]$	0
$f5$	Quartic function with noise		✓	✓		$f(x)=\sum_{i=1}^{n} ix_i^4+rand(0,1)$	30	$[-1.28,1.28]$	0
$f6$	Bohachevsky	✓			✓	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$	30	$[-100,100]$	0
$f7$	Ackley	✓			✓	$f(x)=-20\exp\Big(-0.2\times\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\Big)$ $-\exp\big(\frac{1}{n}\times\sum_{i=1}^{n}\cos(2\pi x_i)\big)+20+\exp(1)$	30	$[-32,32]$	0
$f8$	Griewank	✓			✓	$f(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\Big(\frac{x_i}{\sqrt{i}}\Big)+1$	30	$[-600,600]$	0
$f9$	Levy	✓		✓	$f(x)=\sin^2(3\pi x_1)+(x_1-1)^2[1+\sin^2(3\pi x_2)]$ $+(x_1-1)^2[1+\sin^2(2\pi x_2)]$	2	$[-10,10]$	0
$f10$	Michalewiz	✓		✓	$f(x)=-\sum_{i=1}^{n}\sin(x_i)\big[\sin\big(\frac{ix_i^2}{\pi}\big)\big]^{20}$	10	$[0,\pi]$	-9.66015
$f11$	Rastrigin	✓		✓	$f(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	30	$[-5.12,5.12]$	0
$f12$	Alpine	✓		✓	$f(x)=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	30	$[-10,10]$	0
$f13$	Schaffer	✓			✓	$f(x)=(x_1^2+x_2^2)^{0.25}\times[50(x_1^2+x_2^2)^{0.1}+1]$	30	$[-100,100]$	0
$f14$	Rosenbrock		✓		✓	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	$[-10,10]$	0
$f15$	Easom		✓		✓	$f(x)=-\cos(x_1)\cos(x_2)\times\exp\big[-(x_1-\pi)^2+(x_2-\pi)^2\big]$	2	$[-100,100]$	-1
$f16$	Shubert	✓		✓	$f(x)=\Big(\sum_{i=1}^{5} i\cos((i+1)x_1+i)\Big)\times$ $\times\Big(\sum_{i=1}^{5} i\cos((i+1)x_2+i)\Big)$	3	$[-10,10]$	-186.7309
$f17$	Shubert 1.2		✓		✓	$f(x)=\sum_{i=1}^{n}\Big(\sum_{j=1}^{i} x_j\Big)^2$	2	$[-10,10]$	0
$f18$	Shubert 2.21		✓	✓		$f(x)=MAX(\lvert x_i\rvert,1\le i\le n)$	30	$[-10,10]$	0
$f19$	Shubert 2.22		✓		✓	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	$[-10,10]$	0
$f20$	Schwefel 2.26	✓		✓		$f(x)=418.9829n-\sum_{i=1}^{n}\Big[x_i\sin\Big(\sqrt{\lvert x_i\rvert}\Big)\Big]$	30	$[-500,500]$	0
$f21$	Booth		✓		✓	$f(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	30	$[-10,10]$	0
$f22$	Goldstein price	✓			✓	$f(x)=\big[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2+14x_2$ $+6x_1x_2+3x_2^2)\big]\times\big[30+(2x_1+3x_2)^2(18-32x_1$ $+12x_1^2+48x_2-36x_1x_2+27x_2^2)\big]$	2	$[-2,2]$	0
$f23$	Matyas		✓		✓	$f(x)=0.26(x_1^2+x_2^2)-0.48x_1x_2$	2	$[-10,10]$	0
$f24$	Powell		✓		✓	$f(x)=\sum_{i=1}^{n/4}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2$ $+(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4$	32	$[-4,5]$	0
$f25$	Power sum	✓			✓	$f(x)=\sum_{i=1}^{n}\Big[(\sum_{k=1}^{n} x_k^i)-b_i\Big], b=(8,18,44,11)$	4	$[0,n]$	0
$f26$	Shekel 4.5	✓			✓	$f(x)=-\sum_{i=1}^{m}\Big(\sum_{j=1}^{4}(x_j-C_{ji})^2+\beta_i\Big)^{-1}$, *where* $m=10,\quad \beta=\frac{1}{10}(1,2,4,4,6,3,7,5,5)^T$ $C=\begin{bmatrix}4&1&8&6&3&2&5&8&6&7.0\\4&1&8&6&7&9&3&1&2&3.6\\4&1&8&6&3&2&5&8&6&7.0\\4&1&8&6&7&9&3&1&2&3.6\end{bmatrix}$	4	$[0,10]$	-10.5364
$f27$	Sum squares		✓	✓		$f(x)=\sum_{i=1}^{n} ix_i^2$	30	$[-10,10]$	0
$f28$	Trid	✓			✓	$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}(x_ix_{i-1})$	6	$[-n^2,n^2]$	-50
$f29$	Zettl		✓		✓	$f(x)=(x_1^2+x_2^2-2x_1)^2+0.25x_1$	2	$[-1,5]$	-0.00379
$f30$	Leon		✓		✓	$f(x)=100(x_2-x_1^3)^2+(1-x_1)^2$	2	$[-1.2,1.2]$	0

* Mm: Multimodal, Un: Unimodal, Sp: Separable, NS: Non-Separable, Dim:Dimension, Opt: Optimal

4 Results and discussion

In this section, the MALL algorithm has been evaluated on 40 test functions that are generally used by many researchers. A detailed description of this benchmark is provided below.

4.1 Benchmark and test functions

The proposed algorithms are evaluated on forty test functions. According to Table 3 the first thirty functions are divided into four categories: multimodal, unimodal, separable, and non-separable, and include functions with fixed or variable dimensions [27, 68, 101]. In Fig. 9 some of these functions are drawn three-dimensionally with two decision variables. According to Table 4 the number of ten functions used in CEC-2019 is among the last functions evaluated in this section [74]. CEC-2019 functions include black box and constraint problems.

Table 4: CEC-C06 2019 benchmark functions.

ID	Benchmark function	Dimension	Range	Optimal
*cec*1	Storn's Chebyshev Polynomial Fitting Problem	9	$[-8192, 8192]$	1
*cec*2	Inverse Hilbert Matrix Problem	16	$[-16384, 16384]$	1
*cec*3	Lennard-Jones global optimization problem	18	$[-4, 4]$	1
*cec*4	Rastrigin's Function	10	$[-100, 100]$	1
*cec*5	Griewangk's Function	10	$[-100, 100]$	1
*cec*6	Weierstrass Function	10	$[-100, 100]$	1
*cec*7	Modified Schwefel's Function	10	$[-100, 100]$	1
*cec*8	Expanded Schaffer's F6 Function	10	$[-100, 100]$	1
*cec*9	Happy Cat Function	10	$[-100, 100]$	1
*cec*10	Ackley Function	10	$[-100, 100]$	1

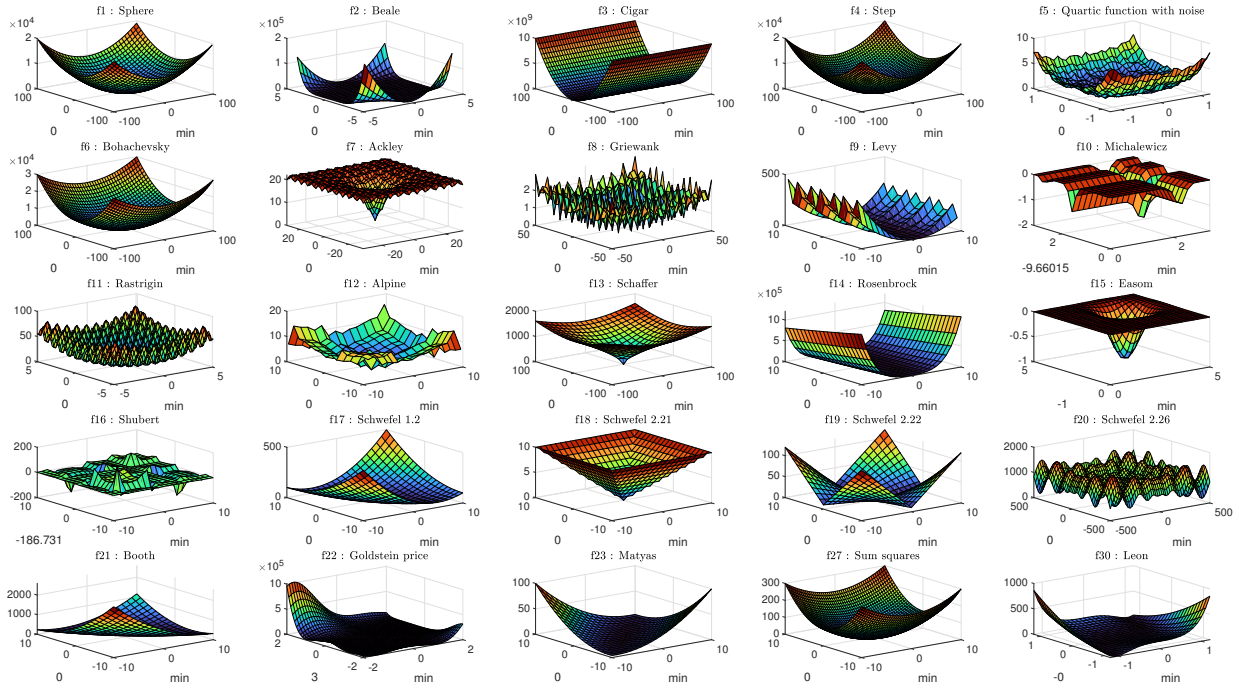

Figure 9: two-dimensional benchmark functions.

4.2 Elite algorithms for comparison

To validate the performance of the MALL algorithm, four elite optimization algorithms include Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), and Teaching-Learning-Based Optimization (TLBO) are used for comparison.

Genetic Algorithm (GA) [44]: It is the most famous metaheuristic algorithm that was inspired by the theory of natural selection and developed evolutionarily. This algorithm uses selection, crossover, and mutation operators to find the optimal solution.

Particle Swarm Optimization (PSO) [53]: It is one of the well-known population-based optimization algorithms that is inspired by the social behavior of birds. In this algorithm, each particle in the search space moves in the direction of self-improvement and global optimization and updates its position. The small number of parameters, high speed, and optimal quality of the solution have led to more attention than other algorithms.

Simulated Annealing (SA) [54, 17]: is an optimization algorithm based on trajectory methods and a single solution. This algorithm works by imitating the physical process that, by heating and slowly cooling a solid material, gives an orderly arrangement to its crystals and causes the minimum energy configuration.

Teaching-Learning-Based Optimization (TLBO) [77]: It is a swarm optimization algorithm that is inspired by human behavior. In this algorithm, a group of learners learns and achieves the global optimum with the influence of the teacher and interaction with other learners.

4.3 Experiments settings

The recommended settings of the parameters of the MALL algorithm and other optimization algorithms used in this research are shown in Table 5. All the test processes and reported results are in Matlab software version R2021b, which was implemented and executed in a Microsoft Windows 10 environment on an Intel (R) Core (TM) i7-4500U processor with 1.80 GHz and 8 GB of RAM memory.

Table 5: Parameter values in the compared algorithms.

Algorithms	Parameters	Values
Mall optimization algorithm (MALL)	Number of Iterations	300
	Population Size	12
	Number of opportunities	4
	Inertia to target & update coefficient	0.9, 0.998
Particle Swarm Optimization (PSO)	Number of Iterations	500
	Population Size	1000
	Inertia Weight	0.99
	Personal & Global Learning Coefficient	1.4962, 1.4962
Genetic Algorithm (GA)	Number of Iterations	500
	Population Size	200
	Crossover and Mutation	0.8, 0.02
Simulated Annealing (SA)	Number of Iterations	500
	Temperature & Number of tests	500, 500
	Boltzmann's constant	1
	Reduction factor	0.95
Teaching learning based optimization (TLBO)	Number of Iterations	500
	Population Size	50

4.4 Performance comparison

To show the performance of the MALL algorithm, its results on the benchmark functions in multimodal, unimodal, separable, and non-separable forms and CEC-C06 2019 black box functions in Tables 6 and 7 and Figs. 10 and 11 have been evaluated. All results are derived from 30 independent runs. In the mentioned tables, bold results are the best solutions.

4.4.1 Exploitation analysis

Unimodal benchmark functions provide a more robust assessment of exploitation to find the optimal solution. As can be seen in Tables 6 and Figs. 10, the MALL algorithm has provided a better optimal solution in most of the test functions. This algorithm has a competitive solution in functions $f19, f21, f23$, and $f30$, which could not provide the best solution, but it has found the best solution in 12 other unimodal benchmark functions.

4.4.2 Exploration analysis

Multimodal benchmark functions provide a more robust assessment of exploration to find the optimal solution. As can be seen in Figs. 10 and Tables 6, the MALL algorithm was able to provide the best solution for all multifaceted benchmark functions except $f12$ and $f13$. In the two mentioned test functions, the distance to the best solution is $1.31E$-11 and $2.04E$-35, respectively, which means that the answer of the MALL algorithm is very competitive. In general, it can be said that it has provided the best solution for 13 other multimodal benchmark functions.

4.4.3 Evaluation of CEC-C06 2019 functions

The set of CEC-C06 2019 test functions is a real approach to solving single-objective optimization problems. These test functions are considered constrained black-box problems. As can be seen in Figs. 11 and Tables 7, the MALL algorithm provides a better solution than other competitors in test functions $cec1, cec2, cec3$, and $cec8$, and in other test functions, the answer provided is completely competitive.

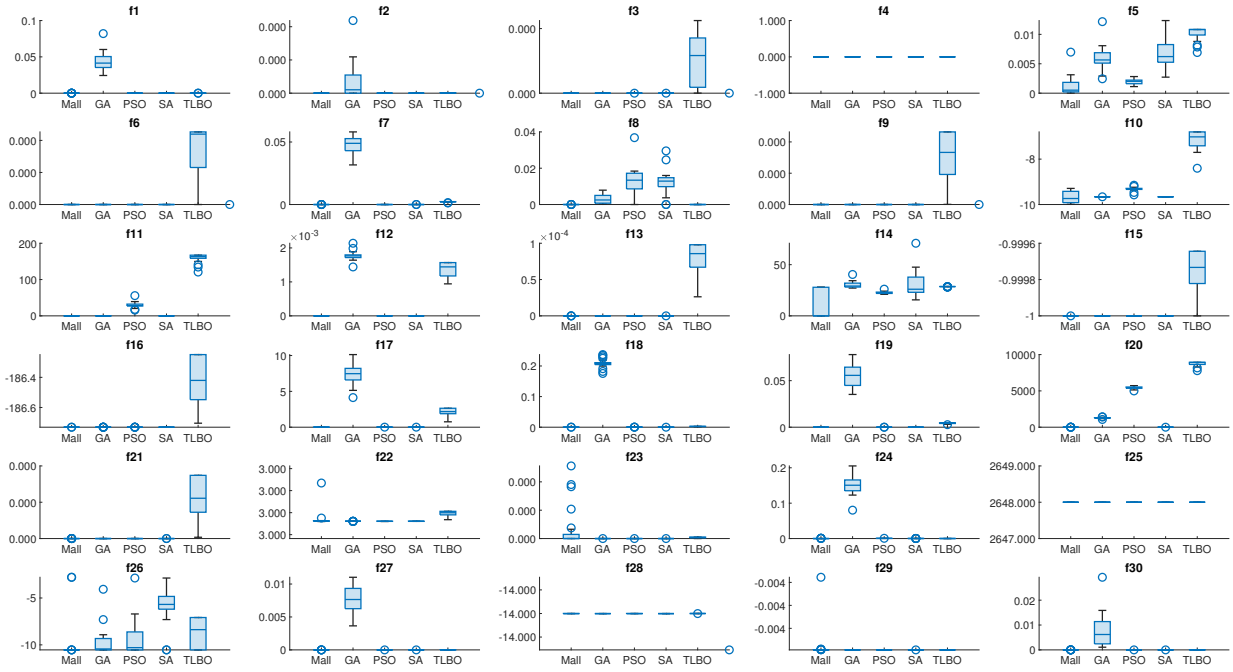

Figure 10: Boxplot analysis of benchmark functions.

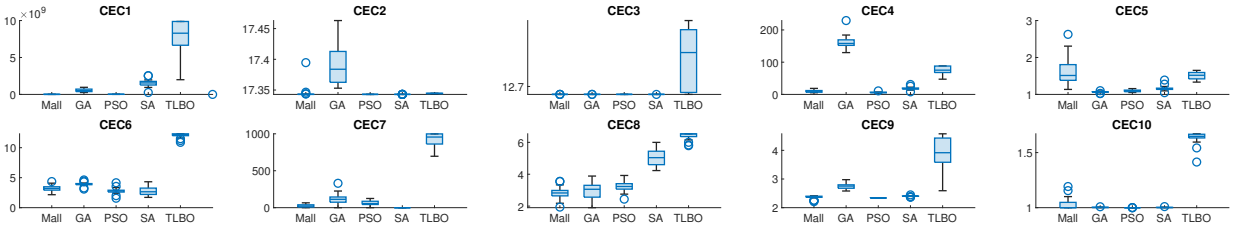

Figure 11: Boxplot analysis of CEC-C06 2019 functions.

Table 6: Results of benchmark functions.

ID	MALL		GA		PSO		SA		TLBO	
	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
$f1$	**5.33E-18**	8.67E-18	4.32E-02	1.19E-02	2.39E-17	1.59E-17	6.39E-17	3.50E-17	5.76E-05	8.62E-06
$f2$	**0.00E+00**	0.00E+00	3.66E-08	5.16E-08	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	9.67E-10	4.86E-10
$f3$	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	7.67E-58	8.90E-58	1.11E-197	0.00E+00	2.67E-30	2.00E-30
$f4$	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
$f5$	**1.10E-03**	1.45E-03	5.91E-03	1.77E-03	1.94E-03	3.86E-04	6.75E-03	2.03E-03	1.01E-02	1.03E-03
$f6$	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	8.32E-15	4.42E-15
$f7$	**5.36E-10**	1.32E-09	4.79E-02	6.53E-03	1.13E-09	2.90E-10	1.50E-09	2.36E-10	2.17E-03	2.43E-04
$f8$	**2.12E-06**	5.54E-06	3.05E-03	2.48E-03	1.30E-02	7.05E-03	1.24E-02	5.80E-03	3.33E-06	1.75E-06
$f9$	**1.35E-31**	6.68E-47	**1.35E-31**	6.68E-47	**1.35E-31**	6.68E-47	**1.35E-31**	6.68E-47	1.53E-26	8.71E-27
$f10$	**-9.68E+00**	2.49E-01	-9.66E+00	2.73E-05	-9.32E+00	7.20E-02	-9.66E+00	1.62E-15	-7.14E+00	3.79E-01
$f11$	**0.00E+00**	0.00E+00	2.48E-02	8.64E-03	2.97E+01	7.28E+00	**0.00E+00**	0.00E+00	1.60E+02	1.08E+01
$f12$	5.09E-11	5.59E-11	1.77E-03	1.13E-04	**3.78E-11**	1.32E-11	4.21E-10	1.21E-10	1.38E-03	1.96E-04
$f13$	2.04E-35	2.90E-35	**8.57E-81**	5.61E-81	2.42E-13	5.80E-14	6.25E-41	5.66E-41	7.96E-05	2.18E-05
$f14$	**9.27E+00**	1.33E+01	3.03E+01	2.96E+00	2.27E+01	1.05E+00	3.10E+01	1.20E+01	2.86E+01	1.61E-01
$f15$	**-1.00E+00**	2.06E-17	**-1.00E+00**	0.00E+00	**-1.00E+00**	0.00E+00	**-1.00E+00**	0.00E+00	**-1.00E+00**	1.25E-04
$f16$	**-1.87E+02**	1.43E-13	**-1.87E+02**	2.59E-14	**-1.87E+02**	2.64E-14	**-1.87E+02**	3.08E-14	**-1.87E+02**	1.54E-01
$f17$	**1.51E-03**	1.50E-03	7.30E+00	1.30E+00	3.80E-03	1.07E-03	2.96E-03	1.36E-03	2.13E+00	4.93E-01
$f18$	**1.90E-05**	3.22E-05	2.08E-01	1.29E-02	1.90E-04	4.07E-05	2.45E-05	3.77E-06	2.39E-03	4.57E-04
$f19$	1.57E-10	2.11E-10	5.44E-02	1.17E-02	**1.11E-10**	3.25E-11	1.35E-09	1.95E-10	4.17E-03	5.83E-04
$f20$	**3.81E-04**	1.11E-03	1.28E+03	6.41E+01	5.44E+03	1.48E+02	3.82E-04	1.25E-12	8.75E+03	2.96E+02
$f21$	1.37E-34	5.17E-34	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	9.12E-32	1.61E-31	2.29E-15	1.02E-15
$f22$	**3.00E+00**	6.32E-14	**3.00E+00**	1.29E-15	**3.00E+00**	1.83E-15	**3.00E+00**	1.12E-15	**3.00E+00**	2.32E-14
$f23$	1.43E-14	3.28E-14	4.00E-37	5.63E-37	1.58E-51	1.58E-51	**2.63E-86**	1.83E-86	1.72E-15	9.30E-16
$f24$	**1.19E-04**	2.67E-04	1.51E-01	2.39E-02	4.33E-04	9.20E-05	1.82E-04	1.12E-05	1.22E-04	4.71E-05
$f25$	**2.65E+03**	0.00E+00	**2.65E+03**	0.00E+00	**2.65E+03**	0.00E+00	**2.65E+03**	0.00E+00	**2.65E+03**	0.00E+00
$f26$	**-1.00E+01**	1.97E+00	-9.79E+00	1.32E+00	-9.26E+00	1.82E+00	-5.73E+00	1.69E+00	-8.65E+00	1.48E+00
$f27$	**1.43E-18**	2.54E-18	7.64E-03	1.96E-03	2.60E-18	1.37E-18	2.09E-17	1.05E-17	1.14E-05	3.88E-06
$f28$	**-1.40E+01**	1.48E-15	**-1.40E+01**	1.81E-15	**-1.40E+01**	1.48E-15	**-1.40E+01**	1.81E-15	**-1.40E+01**	3.30E-16
$f29$	**-3.79E-03**	2.88E-13	**-3.79E-03**	1.76E-18	**-3.79E-03**	1.76E-18	**-3.79E-03**	1.75E-18	**-3.79E-03**	1.05E-18
$f30$	2.11E-19	7.16E-19	7.59E-03	6.31E-03	**2.75E-29**	2.31E-29	1.35E-12	7.50E-13	2.86E-05	1.19E-05

Table 7: Results of CEC-C06 2019 functions.

ID	MALL		GA		PSO		SA		TLBO	
	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
*cec*1	**2.58E+06**	2.11E+06	5.59E+08	2.01E+08	8.00E+06	2.66E+06	1.52E+09	4.42E+08	7.69E+09	2.38E+09
*cec*2	**1.73E+01**	9.42E-03	1.74E+01	3.00E-02	**1.73E+01**	7.23E-15	**1.73E+01**	8.21E-15	**1.73E+01**	5.10E-04
*cec*3	**1.27E+01**	5.18E-15	**1.27E+01**	9.11E-10	**1.27E+01**	3.61E-15	**1.27E+01**	2.67E-14	**1.27E+01**	6.17E-06
*cec*4	1.03E+01	3.63E+00	1.61E+02	1.96E+01	**5.98E+00**	1.39E+00	1.84E+01	3.89E+00	7.60E+01	1.09E+01
*cec*5	1.64E+00	3.48E-01	**1.06E+00**	1.58E-02	1.10E+00	2.69E-02	1.16E+00	5.77E-02	1.51E+00	1.07E-01
*cec*6	3.19E+00	5.25E-01	3.92E+00	2.76E-01	2.78E+00	4.43E-01	**2.76E+00**	6.51E-01	1.20E+01	3.61E-01
*cec*7	2.54E+01	1.90E+01	1.11E+02	6.80E+01	6.33E+01	3.29E+01	**0.00E+00**	0.00E+00	9.21E+02	8.89E+01
*cec*8	**2.81E+00**	3.75E-01	2.96E+00	4.65E-01	3.24E+00	2.91E-01	5.05E+00	4.77E-01	6.40E+00	2.13E-01
*cec*9	2.36E+00	5.85E-02	2.75E+00	8.55E-02	**2.34E+00**	1.00E-04	2.40E+00	1.32E-02	3.92E+00	5.49E-01
*cec*10	1.03E+00	5.08E-02	**1.00E+00**	2.03E-03	**1.00E+00**	2.10E-10	**1.00E+00**	2.17E-03	1.63E+00	5.09E-02

4.4.4 Statistical analysis

In addition to the basic statistical analysis discussed above, the results of the algorithms were compared through the Wilcoxon rank-sum test in Table 8. This is one of the sets of non-parametric tests in the comparison of pairs of algorithms that attracted the attention of many researchers [23]. This test was performed by considering the best solution obtained by each algorithm for each test function during 30 independent executions with a confidence level of 95% ($\alpha = 0.05$), and the results of the P-value and the status of the answer are presented in Table 8. In this table, the "tick" sign indicates that the MALL algorithm is better and wins the test and the "cross" sign indicates that there is no reason for the MALL algorithm to be better and lose the test. In cases where all the samples from both groups are zero, it is not possible to test. In the last line of this table, the number of wins and losses of the MALL algorithm is presented. The Wilcoxon rank-sum test at a significance level of 95% confirms that the MALL algorithm has statistically significant and superior performance compared to other compared algorithms.

4.4.5 Convergence analysis

The graphical results of the convergence analysis are shown in Fig. 12. Here, all the studied algorithms were executed with 1000 iterations. The MALL algorithm shows promising convergence behavior. In most of the functions, the MALL algorithm moves towards the optimal solution with a consistent trend. but in the benchmark functions $f3, f6, f11, f14, f20$, and $f21$, it faces a sudden jump in the convergence path and has moved towards the optimal solution in a stepwise manner.

According to the literature, when there is a balance between exploration and exploitation, complex problems may be solved effectively. From the convergence analysis in the mentioned figure, it seems that in the MALL algorithm, through the appropriate selection of the parameters of the number of opportunities, the inertia of the target, and the update rate, the balance between exploration and exploitation is maintained.

4.4.6 Scalability study

In order to analyze the scalability of the MALL algorithm, benchmark functions that had the ability to change the number of decision variables were used. Here, the scalability of the MALL algorithm was tested in 1000 iterations with a number of 30, 50, 80, and 100 decision variables. Fig. 13 shows the performance of the MALL algorithm with different behaviors in different dimensions. It can be seen that even if the number of decision variables increases, the MALL algorithm has the ability to find the optimal solution.

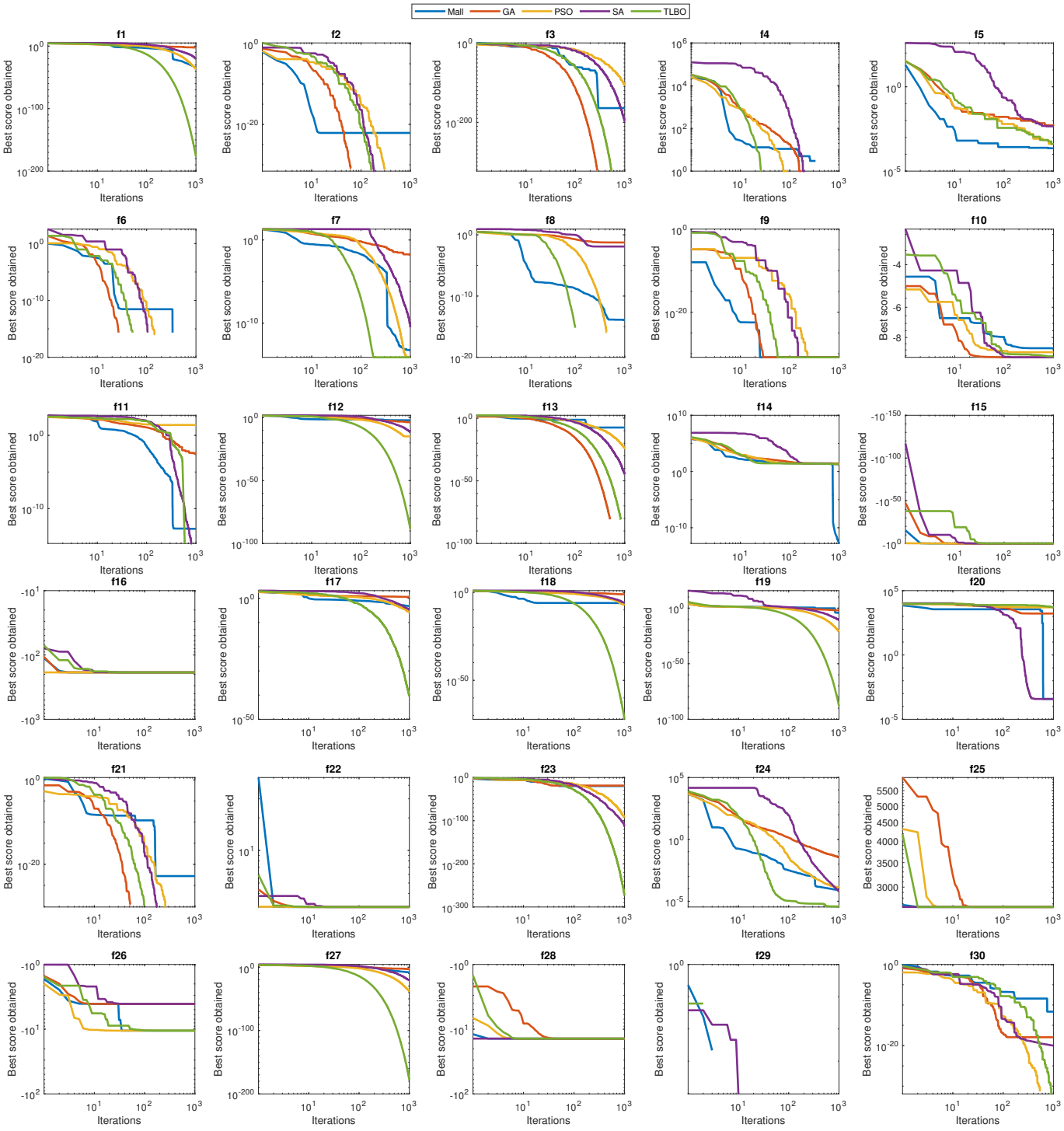

Figure 12: Convergence analysis of the MALL algorithm and competitor algorithms in benchmark functions.

Table 8: Results of Wilcoxon ranksum test.

Function	MALL vs GA		MALL vs PSO		MALL vs SA		MALL vs TLBO	
	P-Value	Win	P-Value	Win	P-Value	Win	P-Value	Win
$f1$	3.02E-11	✔	4.11E-07	✔	1.07E-09	✔	2.26E-11	✔
$f2$	5.32E-06	✔	NaN	-	NaN	-	1.17E-12	✔
$f3$	NaN	-	1.21E-12	✔	1.21E-12	✔	1.19E-12	✔
$f4$	NaN	-	NaN	-	NaN	-	NaN	-
$f5$	4.62E-10	✔	1.49E-04	✔	1.96E-10	✔	2.16E-11	✔
$f6$	NaN	-	NaN	-	NaN	-	3.79E-11	✔
$f7$	3.02E-11	✔	3.26E-07	✔	1.16E-07	✔	2.71E-11	✔
$f8$	3.00E-11	✔	4.24E-08	✔	5.84E-09	✔	4.17E-05	✔
$f9$	NaN	-	NaN	-	NaN	-	8.31E-13	✔
$f10$	3.79E-01	✗	1.43E-08	✔	0.3736787	✗	2.80E-11	✔
$f11$	1.21E-12	✔	1.21E-12	✔	NaN	-	1.08E-12	✔
$f12$	3.02E-11	✔	5.69E-01	✗	3.02E-11	✔	2.80E-11	✔
$f13$	3.01E-11	✔	3.02E-11	✔	5.00E-09	✔	2.39E-11	✔
$f14$	3.50E-09	✔	2.61E-02	✔	1.75E-05	✔	1.83E-11	✔
$f15$	3.33E-01	✗	3.34E-01	✗	3.34E-01	✗	1.60E-12	✔
$f16$	2.44E-02	✔	1.26E-02	✔	4.16E-02	✔	1.77E-11	✔
$f17$	3.02E-11	✔	3.52E-07	✔	2.68E-04	✔	2.80E-11	✔
$f18$	3.02E-11	✔	4.50E-11	✔	3.18E-03	✔	2.92E-11	✔
$f19$	3.02E-11	✔	7.24E-02	✗	3.02E-11	✔	2.71E-11	✔
$f20$	3.02E-11	✔	3.02E-11	✔	1.06E-07	✔	1.61E-11	✔
$f21$	1.93E-10	✔	1.93E-10	✔	1.57E-01	✗	2.78E-11	✔
$f22$	3.18E-08	✔	6.77E-09	✔	1.14E-08	✔	6.31E-10	✔
$f23$	1.45E-10	✔	3.02E-11	✔	3.02E-11	✔	1.59E-02	✔
$f24$	3.02E-11	✔	1.03E-06	✔	9.51E-06	✔	1.07E-05	✔
$f25$	NaN	-	NaN	-	NaN	-	NaN	-
$f26$	1.42E-03	✔	6.60E-01	✗	7.54E-07	✔	7.81E-09	✔
$f27$	3.02E-11	✔	3.37E-05	✔	2.61E-10	✔	2.80E-11	✔
$f28$	6.17E-04	✔	0.192462	✗	6.17E-04	✔	3.57E-07	✔
$f29$	3.35E-07	✔	3.35E-07	✔	1.15E-06	✔	1.36E-01	✗
$f30$	3.02E-11	✔	8.35E-08	✔	3.02E-11	✔	2.80E-11	✔
$cec1$	3.02E-11	✔	4.57E-09	✔	3.02E-11	✔	2.79E-11	✔
$cec2$	1.61E-10	✔	1.21E-12	✔	7.57E-12	✔	2.47E-08	✔
$cec3$	1.02E-09	✔	8.15E-02	✗	1.01E-09	✔	3.03E-12	✔
$cec4$	3.02E-11	✔	8.84E-07	✔	1.10E-08	✔	2.91E-11	✔
$cec5$	3.02E-11	✔	3.69E-11	✔	6.72E-10	✔	4.91E-01	✗
$cec6$	2.20E-07	✔	5.56E-04	✔	1.07E-02	✔	2.71E-11	✔
$cec7$	2.03E-07	✔	3.96E-06	✔	5.77E-11	✔	2.51E-11	✔
$cec8$	1.71E-01	✗	1.17E-05	✔	3.02E-11	✔	2.11E-11	✔
$cec9$	3.02E-11	✔	9.51E-06	✔	3.01E-04	✔	2.92E-11	✔
$cec10$	1.09E-01	✗	6.74E-02	✗	1.74E-01	✗	2.03E-11	✔
Win/Lose		31/4		28/7		30/4		36/2

5 Solving a real problem

In this section, a well-known constrained supply chain management problem is discussed. The model of this problem is known as PLOT, which includes production, logistics, outbound, and transportation, and seeks to optimize the strategic and operational costs of the entire supply chain [47]. This problem is compared with other algorithms reported in the literature, and its results confirm the performance of the MALL algorithm.

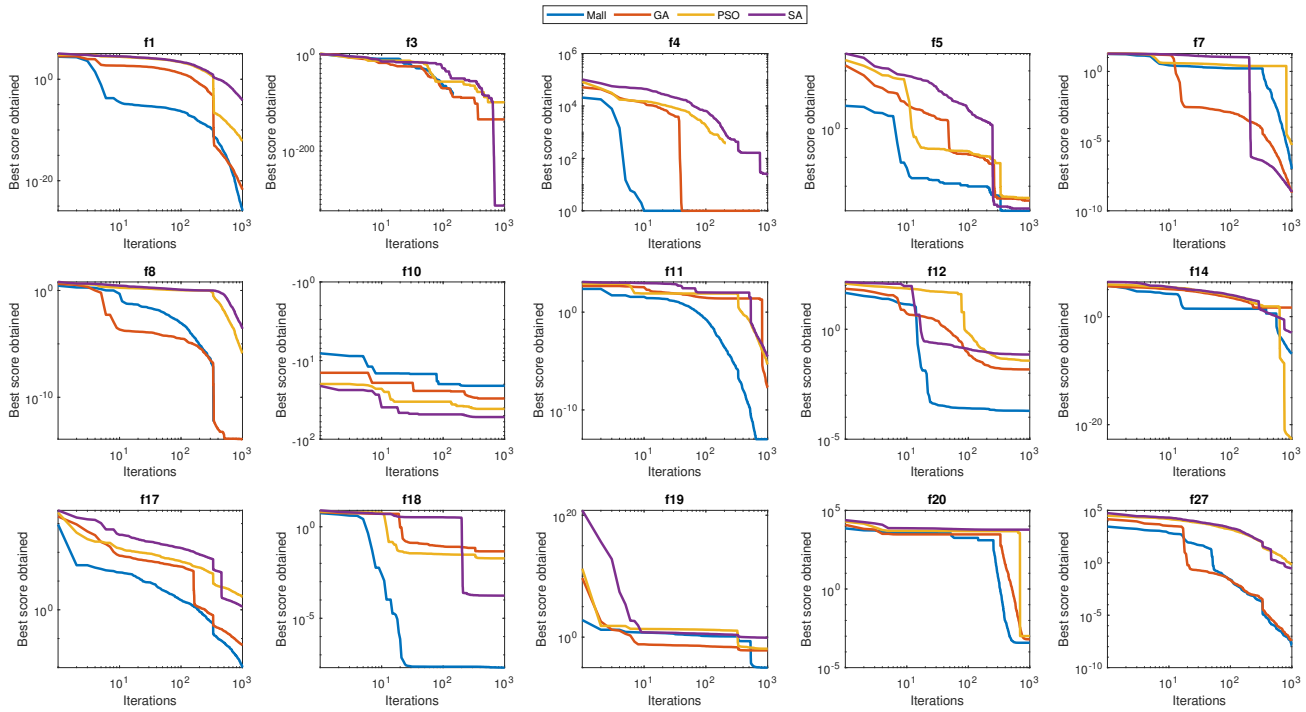

Figure 13: Scalability analysis of the MALL algorithm.

Table 9: Model notations.

Sets	**Description**
I	Set of customer zones $i \in I$
J	Set of cross dock zones $j \in J$
K	Set of warehouse zones $k \in K$
L	Set of product family $l \in L$
Parameters	**Description**
F_j	Fixed operating cost to operate/open cross dock j
V_k	Fixed operating cost to operate/open warehouse k
C_{ijl}	Cost of supply product l from cross-dock j which would be used by customer zone i
C_{jkl}	Cost of transport product l from warehouse k to cross-dock j
TC_{kl}	Transportation cost to deliver one unit of product l to warehouse k
S_{jkl}	Cost to transport one unit of product l from warehouse k to cross-dock j
S_{ijl}	Cost to supply one unit of product l from cross-dock j which would be used by customer zone i
a_{il}	Demand from customer zone i for product l
b_j	Capacity for cross-dock j to handle product families
d_k	Capacity of warehouse k to handle product families
R	Maximum number of cross-docks to be opened
W	Maximum number of warehouses to be opened
Variables	**Description**
Z_{jk}	1 if cross-dock j is open; otherwise 0
P_k	1 if warehouse k is open; otherwise 0
X_{ijl}	1 if customer zone i is assigned to cross-dock j for product l; otherwise 0
Y_{jkl}	1 if cross-dock j is assigned to warehouse k for product l; otherwise 0
B_{kl}	Amount of product l shipped to warehouse k
Q_{jkl}	Amount of product l shipped from warehouse k to cross-dock j
$V_{ij;}$	Amount of product l shipped from cross-dock j to customer i

In the PLOT problem, the manufactured products are first stored in warehouses, then sent to cross-docks, and finally sent to the customer's zone. Fig. 14 schematically shows the movement of the product family as well as the types of strategic and operational decision variables used in this problem. Model notations are presented in Table 9.

In this problem, there are seven groups of decision variables: X_{ijl} are the variables of assigning the customer Zones to the cross-dock, Y_{jkl} are the variables of the cross-dock assignment to the warehouse, Z_j and P_k and are the variables of active status cross-dock and warehouse, respectively. Also, B_{kl} is the number of products transferred

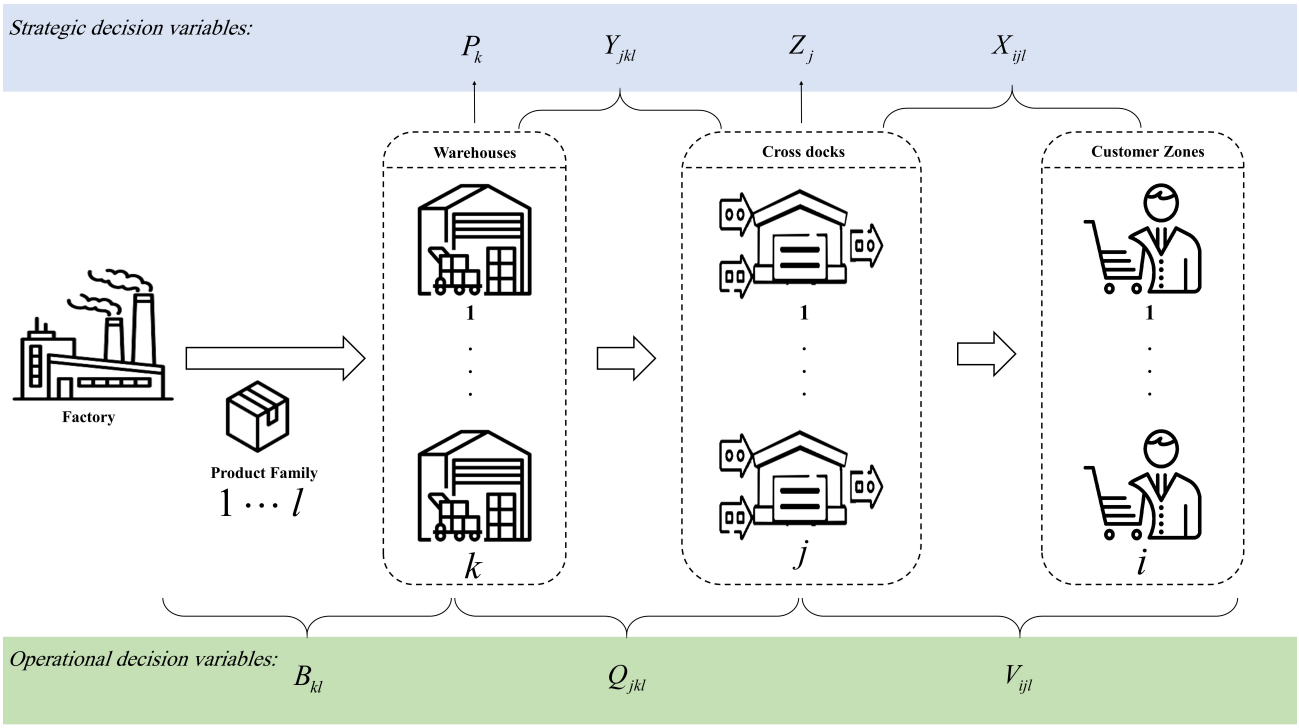

Figure 14: The schematic flow of the supply chain and its strategic and tactical decision variables.

to the warehouse, Q_{jkl} is the number of products transported from the warehouse to the cross-dock, and V_{ijl} is the number of products transported from the cross-dock to the customer's zone.

In Eq. (5.1), the objective of this problem is to minimize OBJ, the operational and strategic costs of the entire supply chain, which includes the fixed costs of warehouses and cross-docks, the costs of assigning customer zones to cross-docks and assigning cross-docks to warehouses, and the costs of supply and transportation between warehouses, cross-docks, customer zones, and the factory.

The first constraint of this problem shown in Eq. (5.2) guarantees the supply of customer demand, and the second one, according to Eq. (5.3), ensures that the active cross-dock supplies products only from the active warehouse. Eqs. (5.4) and (5.5) indicate the limitations of the capacity of cross-docks and warehouses. Eq. (5.6) ensures that customer demand is met by active cross-dock, while Eq. (5.7) guarantees that only active warehouses have product flow. Eqs. (5.8) and (5.9) control the available resources to determine the number of active warehouses and cross-docks. The next constraint, according to Eq. (5.10), guarantees that the total amount of products sent to the warehouses from the factory is equal to the total amount of products sent from the warehouses to the cross-docks. Eq. (5.11) also guarantees that the total number of products sent from warehouses to cross-docks is equal to the total number of products sent from cross-docks to customer zones. Also, the next constraint in Eq. (5.12) satisfies the demand of the customer's zones so that the total products sent from the cross-dock are equal to the demand of the customer's zones. Finally, Eqs. (5.13) and (5.14) indicate the types of decision variables, which are binary and non-negative, respectively.

Objective function:

$$\begin{aligned} minOBJ = &\sum_{j} F_j Z_j + \sum_{k} V_k P_k + \sum_{i}\sum_{j}\sum_{l} C_{ijl} X_{ijl} + \sum_{j}\sum_{k}\sum_{l} C_{jkl} Y_{jkl} \\ &+ \sum_{k}\sum_{l} TC_{kl} B_{kl} + \sum_{j}\sum_{k}\sum_{l} S_{jkl} Q_{jkl} + \sum_{i}\sum_{j}\sum_{l} S_{ijl} V_{ijl} \end{aligned} \tag{5.1}$$

Subject to:

$$\sum_{j} X_{ijl} = 1 \; \forall \, i \text{ and } l, \tag{5.2}$$

$$\sum_{k} Y_{jkl} = Z_j \; \forall \, i \text{ and } l, \tag{5.3}$$

$$\sum_{i} \sum_{l} a_{il} X_{ijl} \leq b_j \; \forall \, j, \tag{5.4}$$

$$\sum_{j} \sum_{l} b_j Y_{jkl} \leq b_k \; \forall \, k, \tag{5.5}$$

$$X_{ijl} \leq Z_j \; \forall \, i, j \text{ and } l, \tag{5.6}$$

$$Y_{jkl} \leq P_k \; \forall \, j, k \text{ and } l, \tag{5.7}$$

$$\sum_{j} Z_j \leq R, \tag{5.8}$$

$$\sum_{k} P_k \leq W, \tag{5.9}$$

$$B_{kl} = \sum_{j} Q_{jkl} \; \forall \, k \text{ and } l, \tag{5.10}$$

$$\sum_{k} Q_{jkl} = \sum_{i} V_{ijl} \; \forall \, j \text{ and } l, \tag{5.11}$$

$$\sum_{j} V_{ijl} = a_{il} \; \forall \, i \text{ and } l, \tag{5.12}$$

$$X_{ijl}, Y_{jkl}, Z_j, P_k = \{0, 1\} \; \forall \, i, j, k \text{ and } l, \tag{5.13}$$

$$B_{kl}, V_{jkl}, Q_{jkl} \geq 0 \; \forall \, i, j, k \text{ and } l, \tag{5.14}$$

In the following, five PLOT problems with different dimensions were considered. These problems have different numbers of customer zones (i), cross-docks (j), product families (l), and warehouses (k). Each of these problems was solved 30 times by the MALL algorithm and other competing algorithms; the statistical results are presented in Table 10 and Fig. 15. The MALL algorithm has provided very favorable competitive results, which statistically show better quality than other competing algorithms.

Table 10: results of PLOT problems.

PLOT size	MALL		GA		PSO		SA		TLBO	
$i \times j \times l \times k$	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
$8 \times 4 \times 4 \times 3$	2275.5	426.7	3188.9	930.6	2860.2	669.8	4841.9	64.3	2750.4	504.5
$10 \times 5 \times 5 \times 4$	3864.5	393.3	4508.6	825.8	3957.6	387.8	5230.9	174.7	4220.8	471.3
$12 \times 6 \times 6 \times 4$	4708.7	852.3	5312.9	1016.5	4965.6	767.6	6914.8	216.7	5769.7	682.8
$15 \times 7 \times 7 \times 5$	5295.5	461.9	7050.0	485.8	7050.0	485.8	8527.5	70.3	6540.0	134.4
$20 \times 8 \times 5 \times 8$	6194.8	144.6	8554.3	1397.1	8107.2	1145.1	9682.8	185.1	6954.7	542.6

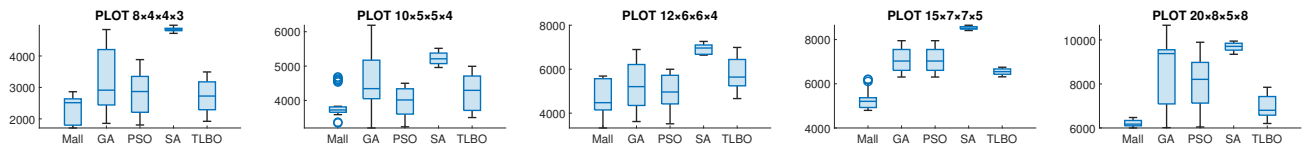

Figure 15: Boxplot analysis of PLOT problems.

This comprehensive study states that the proposed MALL algorithm has desirable competency among other competing algorithms. In the benchmark functions, the performance of the MALL algorithm in terms of avoiding local optima, exploration, exploitation, and convergence proved, and the results of real problems indicate the superior performance of the proposed algorithm in solving these problems. Finally, this algorithm is proposed to solve supply chain problems.

6 Conclusion

This research presents a novel optimization algorithm based on population called the Mall Optimization Algorithm (MALL), inspired by human behavior in shopping malls. The proposed method of this algorithm is an imitation of the behavior of customers, tourists, and all kinds of sellers in malls. Thirty benchmark functions and ten challenging functions of CEC-C06 2019 were evaluated to test the performance of this algorithm in terms of avoiding local optima, exploration, exploitation, and convergence.

The results indicate that MALL can produce highly competitive results compared to the well-known algorithms GA, PSO, SA, and TLBO. First, the results on unimodal functions showed a more favorable exploitation of the MALL algorithm. Second, the exploration ability of MALL was confirmed by the results presented in multimodal functions. Third, the results of the composite functions proved the effectiveness of avoiding the local optimum. Fourth, statistical analyses were discussed in comparison with other metaheuristic algorithms and confirmed the desirability of this proposed algorithm. Finally, the MALL convergence analysis proved the convergence of this algorithm.

In addition, the results of the supply chain problems showed that the MALL algorithm has high performance in the unknown and challenging search space. This algorithm showed a significant improvement in the PLOT problem compared to the current approaches, which shows its application in solving real problems.

For future work, we plan to applying MALL to other problems, and control the performance of the suggested algorithm, and develop a multi-objective version of the MALL algorithm. Also redevelop this algorithm according to behaviors of other interested parties of shopping mall.

References

[1] H.A. Abbass, *MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach*, Proc. 2001 Congr. Evol. Comput. (IEEE Cat. No. 01TH8546), IEEE, vol. 1, pp. 207–214, 2001.

[2] P. Agrawal, H.F. Abutarboush, T. Ganesh, and A.W. Mohamed, *Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009–2019)*, IEEE Access **9** (2021), 26766–26791.

[3] S.-A. Ahmadi, *Human behavior-based optimization: A novel metaheuristic approach to solve complex optimization problems*, Neural Comput. Appl. **28** (2017), no. Suppl 1, 233–244.

[4] A. Ahmadi-Javid, *Anarchic Society Optimization: A human-inspired method*, IEEE Congr. Evol. Comput. (CEC), IEEE, 2011, pp. 2586–2592.

[5] A. Ahrari and A.A. Atai, *Grenade explosion method—a novel tool for optimization of multimodal functions*, Appl. Soft Comput. **10** (2010), no. 4, 1132–1140.

[6] B. Alatas, *ACROA: artificial chemical reaction optimization algorithm for global optimization*, Expert Syst. Appl. **38** (2011), no. 10, 13170–13180.

[7] E. Alba and B. Dorronsoro, *The exploration/exploitation tradeoff in dynamic cellular genetic algorithms*, IEEE Trans. Evol. Comput. **9** (2005), no. 2, 126–142.

[8] S. Arora and S. Singh, *Butterfly optimization algorithm: A novel approach for global optimization*, Soft Comput. **23** (2019), no. 3, 715–734.

[9] Q. Askari, M. Saeed, and I. Younas, *Heap-based optimizer inspired by corporate rank hierarchy for global optimization*, Expert Syst. Appl. **161** (2020), 113702.

[10] A. Askarzadeh, *Bird mating optimizer: An optimization algorithm inspired by bird mating strategies*, Commun. Nonlinear Sci. Numer. Simul. **19** (2014), no. 4, 1213–1228.

[11] E. Atashpaz-Gargari and C. Lucas, *Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition*, IEEE Congr. Evol. Comput.„ 2007, IEEE, pp. 4661–4667.

[12] M.A. Awadel-Bayoumy, M. Rashad, M. Elsoud, and M. El-Dosuky, *Fafsa: Fast artificial fish swarm algorithm*, Int. J. Inf. Sci. Intell. Syst. **2** (2013), no. 4, 60–70.

[13] B. Basturk, *An artificial bee colony (ABC) algorithm for numeric function optimization*, IEEE Swarm Intell. Symp., Indianapolis, IN, USA, 2006, vol. 2006, p. 12.

[14] M. Birattari, L. Paquete, T. Stützle, and K. Varrentrapp, *Classification of Metaheuristics and Design of Experiments for the Analysis of Components*, Tech. Rep. AIDA-01-05, Intellektik, Darmstadt Univ. Technol., Darmstadt, Germany, 2001.

[15] C. Blum, *Ant colony optimization: Introduction and recent trends*, Phys. Life Rev. **2** (2005), no. 4, 353–373.

[16] I. Boussaïd, J. Lepagnot, and P. Siarry, *A survey on optimization metaheuristics*, Inform. Sci. **237** (2013), 82–117.

[17] V. Černý, *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*, J. Optim. Theory Appl. **45** (1985), 41–51.

[18] P. Civicioglu, *Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm*, Comput. Geosci. **46** (2012), 229–247.

[19] P. Civicioglu, *Backtracking search optimization algorithm for numerical optimization problems*, Appl. Math. Comput. **219** (2013), no. 15, 8121–8144.

[20] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, and H. Sossa, *Circle detection using electro-magnetism optimization*, Inform. Sci. **182** (2012), no. 1, 40–55.

[21] Z. Cui and X. Cai, *Using social cognitive optimization algorithm to solve nonlinear equations*, 9th IEEE Int. Conf. Cogn. Inform. (ICCI'10), IEEE, 2010, pp. 199–203.

[22] C. Dai, W. Chen, Y. Song, and Y. Zhu, *Seeker optimization algorithm: A novel stochastic search algorithm for global numerical optimization*, J. Syst. Eng. Electron. **21** (2010), no. 2, 300–311.

[23] J. Derrac, S. García, D. Molina, and F. Herrera, *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms*, Swarm Evol. Comput. **1** (2011), no. 1, 3–18.

[24] G. Dhiman and V. Kumar, *Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications*, Adv. Eng. Softw. **114** (2017), 48–70.

[25] G. Dhiman and V. Kumar, *Emperor penguin optimizer: A bio-inspired algorithm for engineering problems*, Knowl.-Based Syst. **159** (2018), 20–50.

[26] T.T. Dhivyaprabha, P. Subashini, and M. Krishnaveni, *Synergistic fibroblast optimization: A novel nature-inspired computing algorithm*, Front. Inf. Technol. Electron. Eng. **19** (2018), no. 7, 815–833.

[27] J.G. Digalakis and K.G. Margaritis, *On benchmarking functions for genetic algorithms*, Int. J. Comput. Math. **77** (2001), no. 4, 481–506.

[28] M. Dorigo, M. Birattari, and T. Stutzle, *Ant colony optimization*, IEEE Comput. Intell. Mag. **1** (2006), no. 4, 28–39.

[29] J. Dreo, J.-P. Aumasson, W. Tfaili, and P. Siarry, *Adaptive learning search, a new tool to help comprehending metaheuristics*, Int. J. Artif. Intell. Tools **16** (2007), no. 03, 483–505.

[30] H. Du, X. Wu, and J. Zhuang, *Small-world optimization algorithm for function optimization*, Int. Conf. Nat. Comput., Springer, 2006, pp. 264–273.

[31] M.A. Eita and M.M. Fahmy, *Group counseling optimization*, Appl. Soft Comput. **22** (2014), 585–604.

[32] O.K. Erol and I. Eksin, *A new optimization method: Big bang–big crunch*, Adv. Eng. Softw. **37** (2006), no. 6, 106–111.

[33] E. Fadakar and M. Ebrahimi, *A new metaheuristic football game inspired algorithm*, in 2016 1st Conf. Swarm Intell. Evol. Comput. (CSIEC), IEEE, 2016, pp. 6–11.

[34] T.A. Feo and M.G.C. Resende, *A probabilistic heuristic for a computationally difficult set covering problem*, Oper. Res. Lett. **8** (1989), no. 2, 67–71.

[35] T.A. Feo and M.G.C. Resende, *Greedy randomized adaptive search procedures*, J. Global Optim. **6** (1995), 109–133.

[36] M. Ferry, *F3EA–A targeting paradigm for contemporary warfare*, Aust. Army J. **10** (2013), no. 1, 49–64.

[37] A.H. Gandomi and A.H. Alavi, *Krill herd: A new bio-inspired optimization algorithm*, Commun. Nonlinear Sci. Numer. Simul. **17** (2012), no. 12, 4831–4845.

[38] H. Ghasemian, F. Ghasemian, and H. Vahdat-Nejad, *Human urbanization algorithm: A novel metaheuristic approach*, Math. Comput. Simul. **178** (2020), 1–15.

[39] N. Ghorbani and E. Babaei, *Exchange market algorithm*, Appl. Soft Comput. **19** (2014), 177–187.

[40] F. Glover, *Tabu search methods in artificial intelligence and operations research*, ORSA Artif. Intell. **1** (1987), no. 2, 6.

[41] N. Hansen, S.D. Müller, and P. Koumoutsakos, *Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)*, Evol. Comput. **11** (2003), no. 1, 1–18.

[42] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, *Emperor Penguins Colony: A new meta-heuristic algorithm for optimization*, Evol. Intell. **12** (2019), no. 2, 211–226.

[43] A. Hatamlou, *Black hole: A new heuristic optimization approach for data clustering*, Inform. Sci. **222** (2013), 175–184.

[44] J.H. Holland, *Genetic algorithms*, Sci. Am. **267** (1992), no. 1, 66–73.

[45] K. Hussain, M.N. Mohd Salleh, S. Cheng, and Y. Shi, *Metaheuristic research: A comprehensive survey*, Artif. Intell. Rev. **52** (2019), no. 4, 2191–2233.

[46] M. Jain, V. Singh, and A. Rani, *A novel nature-inspired algorithm for optimization: Squirrel search algorithm*, Swarm Evol. Comput. **44** (2019), 148–175.

[47] V. Jayaraman and A. Ross, *A simulated annealing methodology to distribution network design and management*, Eur. J. Oper. Res. **144** (2003), no. 3, 629–645.

[48] A.H. Kashan, *League championship algorithm: A new algorithm for numerical function optimization*, in 2009 Int. Conf. Soft Comput. Pattern Recognit., IEEE, 2009, pp. 43–48.

[49] A.H. Kashan, R. Tavakkoli-Moghaddam, and M. Gen, *Find-Fix-Finish-Exploit-Analyze (F3EA) meta-heuristic algorithm: An effective algorithm with new evolutionary operators for global optimization*, Comput. Ind. Eng. **128** (2019), 192–218.

[50] A. Kaveh and A. Dadras, *A novel meta-heuristic optimization algorithm: Thermal exchange optimization*, Adv. Eng. Softw. **110** (2017), 69–84.

[51] A. Kaveh and M. Khayatazad, *A new meta-heuristic method: Ray optimization*, Comput. Struct. **112** (2012), 283–294.

[52] A. Kaveh and S. Talatahari, *A novel heuristic optimization method: Charged system search*, Acta Mech. **213** (2010), no. 3-4, 267–289.

[53] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proc. ICNN'95-Int. Conf. Neural Netw., IEEE, 1995, vol. 4, pp. 1942–1948.

[54] S. Kirkpatrick, C.D. Gelatt Jr, and M.P. Vecchi, *Optimization by simulated annealing*, Science **220** (1983), no. 4598, 671–680.

[55] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, vol. 1, 1992.

[56] A.J. Kulkarni, I.P. Durugkar, and M. Kumar, *Cohort intelligence: A self supervised learning behavior*, IEEE Int. Conf. Syst. Man Cybern., IEEE, 2013, pp. 1396–1400.

[57] K.S. Lee and Z.W. Geem, *A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice*, Comput. Methods Appl. Mech. Engrg. **194** (2005), no. 36-38, 3902–3933.

[58] H.R. Lourenço, O.C. Martin, and T. Stützle, *Iterated local search*, Handbook of Metaheuristics, Springer, 2003,

pp. 320–353.

[59] X. Lu and Y. Zhou, *A novel global convergence algorithm: bee collecting pollen algorithm*, Int. Conf. Intell. Comput., Springer, 2008, pp. 518–525.

[60] W. Lv, C. He, D. Li, S. Cheng, S. Luo, and X. Zhang, *Election campaign optimization algorithm*, Procedia Comput. Sci. **1** (2010), no. 1, 1377–1386.

[61] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, and S.M. Mirjalili, *Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems*, Adv. Eng. Softw. **114** (2017), 163–191.

[62] S. Mirjalili, S.M. Mirjalili, and A. Lewis, *Grey wolf optimizer*, Adv. Eng. Softw. **69** (2014), 46–61.

[63] N. Mladenovic, *A variable neighborhood algorithm-a new metaheuristic for combinatorial optimization*, Papers presented at Optimization Days, 1995.

[64] N. Mladenović, M. Dražić, V. Kovačevic-Vujčić, and M. Čangalović, *General variable neighborhood search for the continuous optimization*, Eur. J. Oper. Res. **191** (2008), no. 3, 753–770.

[65] F.F. Moghaddam, R.F. Moghaddam, and M. Cheriet, *Curved Space Optimization: A Random Search Based on General Relativity Theory*, arXiv preprint arXiv:1208.2214, 2012.

[66] R. Moghdani and K. Salimifard, *Volleyball premier league algorithm*, Appl. Soft Comput. **64** (2018), 161–185.

[67] A.W. Mohamed, A.A. Hadi, and A.K. Mohamed, *Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm*, Int. J. Mach. Learn. Cybern. (2019), 1–29.

[68] M. Molga and C. Smutnicki, *Test functions for optimization needs*, Test Funct. Optim. Needs **101** (2005), 48.

[69] N. Moosavian and B.K. Roodsari, *Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks*, Swarm Evol. Comput. **17** (2014), 14–24.

[70] A. Mucherino and O. Seref, *Monkey search: a novel metaheuristic search for global optimization*, AIP Conf. Proc., Amer. Inst. Phys., vol. 953, 2007, pp. 162–173.

[71] O. Olorunda and A.P. Engelbrecht, *Measuring exploration/exploitation in particle swarms using swarm diversity*, IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.), IEEE, 2008, pp. 1128–1134.

[72] W.-T. Pan, *A new fruit fly optimization algorithm: taking the financial distress model as an example*, Knowl.-Based Syst. **26** (2012), 69–74.

[73] P.C. Pinto, T.A. Runkler, and J.M.C. Sousa, *Wasp swarm algorithm for dynamic MAX-SAT problems*, Int. Conf. Adapt. Nat. Comput. Algorithms, Springer, 2007 pp. 350–357.

[74] K.V. Price, N.H. Awad, M.Z. Ali, and P.N. Suganthan, *Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization*, Tech. Rep., Nanyang Technol. Univ., Singapore, 2018.

[75] F. Ramezani and S. Lotfi, *Social-based algorithm (SBA)*, Appl. Soft Comput. **13** (2013), no. 5, 2837–2856.

[76] R. Rao, *Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems*, Int. J. Ind. Eng. Comput. **7** (2016), no. 1, 19–34.

[77] R.V. Rao, V.J. Savsani, and D.P. Vakharia, *Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems*, Comput.-Aided Des. **43** (2011), no. 3, 303–315.

[78] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, *GSA: A gravitational search algorithm*, Inform. Sci. **179** (2009), no. 13, 2232–2248.

[79] T. Ray and K.-M. Liew, *Society and civilization: An optimization algorithm based on the simulation of social behavior*, IEEE Trans. Evol. Comput. **7** (2003), no. 4, 386–396.

[80] I. Rechenberg, *Evolution Strategy*, Computational Intelligence: Imitating Life, J.M. Zurada, R.J. Marks II, and C. Goldberg, Eds., IEEE Press, Piscataway, NJ, 1994, pp. 147–169.

[81] R.G. Reynolds, *An introduction to cultural algorithms*, Proc. 3rd Annu. Conf. Evol. Program., World Scientific, 1994, pp. 131–139.

[82] M. Roth, *Termite: A Swarm Intelligent Routing Algorithm for Mobile Wireless Ad-Hoc Networks*, Master's Thesis, Univ. Hamburg, Hamburg, Germany, 2005.

[83] H. Salimi, *Stochastic fractal search: A powerful metaheuristic algorithm*, Knowl.-Based Syst. **75** (2015), 1–18.

[84] P. Savsani and V. Savsani, *Passing vehicle search (PVS): A novel metaheuristic algorithm*, Appl. Math. Model. **40** (2016), no. 5-6, 3951–3978.

[85] H. Shah-Hosseini, *Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation*, Int. J. Comput. Sci. Eng. **6** (2011), no. 1-2, 132–140.

[86] Y. Shi, *Brain storm optimization algorithm*, Int. Conf. Swarm Intell., Springer, 2011, pp. 303–309.

[87] Y. Shiqin, J. Jianjun, and Y. Guangxing, *A dolphin partner optimization*, in 2009 WRI Global Congr. Intell. Syst., IEEE, vol. 1, pp. 124–128, 2009.

[88] D. Simon, *Biogeography-based optimization*, IEEE Trans. Evol. Comput. **12** (2008), no. 6, 702–713.

[89] R. Storn and K. Price, *Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces*, J. Global Optim. **11** (1997), no. 4, 341–359.

[90] M.H. Sulaiman, Z. Mustaffa, M.M. Saari, and H. Daniyal, *Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems*, Eng. Appl. Artif. Intell. **87** (2020), 103330.

[91] A. Tabari and A. Ahmad, *A new optimization method: Electro-Search algorithm*, Comput. Chem. Eng. **103** (2017), 1–11.

[92] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.

[93] A. Tharwat and T. Gabel, *Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm*, Neural Comput. Appl. **32** (2020), 6925–6938.

[94] C. Voudouris and E. Tsang, *Partial constraint satisfaction problems and guided local search*, Proc., Practical Application of Constraint Technology (PACT'96), London, 1996, pp. 337–356.

[95] A.I. Wagan and M.M. Shaikh, *A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem*, Appl. Soft Comput. **90** (2020), 106176.

[96] B. Webster and P.J. Bernhard, *A local search optimization algorithm based on natural principles of gravitation*, Report, Melbourne, FL. Florida Inst. Technol., 2003.

[97] D.H. Wolpert and W.G. Macready, *No free lunch theorems for optimization*, IEEE Trans. Evol. Comput. **1** (1997), no. 1, 67–82.

[98] Y. Xu, Z. Cui, and J. Zeng, *Social emotional optimization algorithm for nonlinear constrained optimization problems*, in Swarm, Evol., and Memetic Comput.: First Int. Conf. Swarm, Evol., and Memetic Comput., SEMCCO 2010, Chennai, India, December 16-18, 2010. Proc. 1, Springer, 2010, pp. 583–590.

[99] X.-S. Yang, *Firefly algorithm, stochastic test functions and design optimisation*, Int. J. Bio-Inspired Comput. **2** (2010), no. 2, 78–84.

[100] X.-S. Yang, *A new metaheuristic bat-inspired algorithm*, Nature Inspired Cooperative Strategies for Optimization, Springer, 1994, pp. 65–74.

[101] X.-S. Yang, *Test problems in optimization*, arXiv preprint arXiv:1008.0549, 2010.

[102] X.-S. Yang and S. Deb, *Cuckoo search via Lévy flights*, World Congr. Nat. Biol. Inspired Comput. (NaBIC), IEEE, pp. 210–214, 2009.

[103] X. Yao, Y. Liu, and G. Lin, *Evolutionary programming made faster*, IEEE Trans. Evol. Comput. **3** (1999), no. 2, 82–102.

[104] L.M.Y. Zhang, C. Dahlmann, and Y. Zhang, *Human-inspired algorithms for continuous function optimization*, IEEE Int. Conf. Intell. Comput. Intell. Syst., IEEE, 2009, vol. 1, pp. 318–321.