


Article

Attracting Potential Customers in E-Commerce Environments: A Comparative Study of Metaheuristic Algorithms

Reza Yazdani ¹, Mohammad Javad Taghipourian ^{1,*}, Mohammad Mahdi Pourpasha ² 
and Seyed Shamseddin Hosseini ³

¹ Department of Business Management, Chalous Branch, Islamic Azad University, Chalous 46615/397, Iran; rezayazdani@iauc.ac.ir

² Department of Mathematics, Chalous Branch, Islamic Azad University, Chalous 46615/397, Iran; pourpasha@iauc.ac.ir

³ Faculty of Economics, Allameh Tabataba'i University, Tehran 15134, Iran; sh-hosseini@srbiau.ac.ir

* Correspondence: mj.pourian@iauc.ac.ir

Abstract: Internet technology has provided an indescribable new way for businesses to attract new customers, track their behaviour, customise services, products, and advertising. Internet technology and the new trend of online shopping have resulted in the establishment of numerous websites to sell products on a daily basis. Products compete to be displayed on the limited pages of a website in online shopping because it has a significant impact on sales. Website designers carefully select which products to display on a page in order to influence the customers' purchasing decisions. However, concerns regarding appropriate decision making have not been fully addressed. As a result, this study conducts a comprehensive comparative analysis of the performance of ten different metaheuristics. The ant lion optimiser (ALO), Dragonfly algorithm (DA), Grasshopper optimisation algorithm (GOA), Harris hawks optimisation (HHO), Moth-flame optimisation algorithm (MFO), Multi-verse optimiser (MVO), sine cosine algorithm (SCA), Salp Swarm Algorithm (SSA), The whale optimisation algorithm (WOA), and Grey wolf optimiser (GWO) are some of the recent algorithms that were chosen for this study. The results show that the MFO outperforms the other methods in all sizes. MFO has an average normalised objective function of 81%, while ALO has a normalised objective function of 77%. In contrast, HHO has the worst performance of 16%. The study's findings add new theoretical and practical insights to the growing body of knowledge about e-commerce environments and have implications for planners, policymakers, and managers, particularly in companies where an unplanned advertisement wastes the budget.

Keywords: e-commerce; metaheuristics; optimisation; operation research; customers



Citation: Yazdani, R.; Taghipourian, M.J.; Pourpasha, M.M.; Hosseini, S.S. Attracting Potential Customers in E-Commerce Environments: A Comparative Study of Metaheuristic Algorithms. *Processes* **2022**, *10*, 369. <https://doi.org/10.3390/pr10020369>

Academic Editors: Bor-Yih Yu, Vincentius Surya Kurnia Adi, Chien-Yuan Su and Felicia Januarlia Novita

Received: 30 December 2021

Accepted: 8 February 2022

Published: 14 February 2022

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



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

1. Introduction

Companies need to devise a strategy for attracting new customers and generating revenue in order to achieve long-term success [1,2]. In today's competitive environment, the Internet enables businesses to keep track of their customers' points of search and behaviour on a real-time basis [3,4]. In this manner, the information that is collected may be used to provide input for the customisation of products, services, prices, and improvements in the method of communication [5,6]. Data mining (DM), optimisation techniques, and a hybrid of methodologies are often used in the study of online consumer contact as the primary problem in electronic customer relationship management (e-CRM) [7–9]. The launch and the selling of goods on the Internet and online contact with consumers require a thorough understanding of the needs and desires of the target market [10,11]. The progressively accumulated knowledge of consumer wants aids a business in improving the electronic presentation of products and services in order to ensure that they meet or exceed the customer's expectations [12]. Given the ease with which customers may obtain the same information supplied by many manufacturers, effective online communication for sales

promotion is critical for a successful advertisement effort [13]. Effective online communication of information for sales promotion in a virtual environment and the quality of the information that is provided and conveyed to customers is key in their decision making.

Following the collection of the information gathered from previous experiences in online customer contact, the information may be evaluated and used to enhance the company's marketing strategy and to create a long-term connection with its consumers [14,15]. Companies keep track of their customers' previous behaviour and changes in order to respond discreetly to their expectations in the future [16]. The information recorded in a company or other databases is afterwards utilised to forecast future customer behaviour based on the information contained in the database [17].

These days, a wide range of tools and techniques from different fields are being employed in e-commerce [2,18–20]. Data mining involves employing machine learning and statistical techniques to large amounts of raw data in order to uncover exciting patterns [21–23]. The current research indicates that this technique is the most widely used technique for detecting trends, patterns (or habits), and associations in consumer purchasing behaviour, as well as for clustering and categorising customers' preferences, and finally for forecasting future purchases through regression analysis, sequence discovery, or visualisation [24,25]. The customer interests and preferences are clustered and classified using data mining methods, which are also used to assist sales marketing and market segmentation [26]. Among the elements in transaction databases or other data, repositories are intriguing correlations, common patterns, connections, and casual structures that may be extracted. When it comes to clustering and categorising consumer interests, the information on the goods wanted by each customer is gathered and analysed. It may pertain to all of the client's previous purchases or to the particular characteristics that the consumer prefers in relation to a specific group of goods.

Given the importance of having a reliable approach to assisting decision makers in e-commerce environments in absorbing more potential customers through appropriate advertising strategies, "The primary aim of this research is to conduct a comparative study of metaheuristic algorithms in order to assess their potential to assist decision makers in dealing with the challenges that businesses are currently facing in selecting more appropriate options for attracting more customers while operating with a limited budget." The specific research objectives of this research are as follows:

- (1) Investigating the existing studies in e-commerce environments, particularly those that addressed the website design for attracting customers in e-commerce environments using decision-making techniques.
- (2) Evaluating the accuracy and robustness of the metaheuristic algorithms as an optimisation tool in the website design process.

The following are the study's research questions:

- (1) What are the limitations of the existing approaches for an effective website design for attracting customers in e-commerce environments?
- (2) How are the robustness and accuracy of metaheuristic algorithms in finding appropriate decisions for website design to attract potential customers in e-commerce environments?

This research uses information about anticipated consumer behaviour to create web pages that classify clients into similar groups based on their past behaviour. Because of the high level of complexity of this issue, it seems that there is no way to avoid using metaheuristic algorithms to find optimum or near-optimal solutions to the problem under consideration. Advanced computational techniques, especially metaheuristic methods, are becoming more popular among academics at present [27–32]. Metaheuristics have been proven to successfully solve a broad range of complex problems in an acceptable amount of time [33–35]. They can provide the desired solutions in a fair amount of time [36]. Although these techniques have been extensively used in a broad range of study fields, no one algorithm can obtain the optimum solution for all problems. As a result, the search for innovative and efficient optimisation methods continues to be an open problem [37]. In

recent years, the area of metaheuristics has seen rapid growth, with many metaheuristic algorithms being created to date [38,39]. For comprehensive reviews about metaheuristic algorithms, readers are referred to [40–43]. In the present research, a comprehensive comparative study of metaheuristic algorithms among ten new metaheuristics, including the following: The ant lion optimiser (ALO) [44], Dragonfly algorithm (DA) [45], Grasshopper optimisation algorithm (GOA) [46], Harris hawks optimisation (HHO) [47], Moth-flame optimisation algorithm (MFO) [48], Multi-verse optimiser (MVO) [49], sine cosine algorithm (SCA) [50], Salp Swarm Algorithm (SSA) [51], The whale optimisation algorithm (WOA) [52], and Grey wolf optimiser (GWO) [53], are conducted.

The remainder of this article is organised as follows: Section 2 contains a review of the literature. Section 3 explains the problem under investigation, and Section 4 provides optimisation algorithms to address the research problem. Section 5 assesses the performance of the algorithms. Section 6 of the article summarises the most important findings and makes recommendations for future research.

2. Literature Review

In this section, available studies in this research area are reviewed. In one of the first studies in this research area, Kleinberg, et al. [54] propose a sampling-based algorithm by simply itemising and assessing all of the possible divisions of a selected group of customers. They demonstrated that catalogue segmentation could, at times, be an NP-complete problem, though with only two catalogues. Steinbach, et al. [55] argue that in the generic hierarchical clustering, in many cases, the nearest neighbours of a document belong to different classes. As a solution to the inefficiency of the collective sampling-based algorithm, they introduce three algorithms based on the K-means clustering approach [55]. Xu, et al. [56] developed an approximation algorithm based on semidefinite programming with a performance guarantee of $1/2$ for any catalogue size of r and a value greater than $1/2$ for a catalogue size of at least $m/3$, where m is the number of available products.

Kleinberg, et al. [57] show that a sampling-based enumeration algorithm is an inefficient approach to actual problem sizes. Alternatively, they developed two heuristic algorithms (ICC and DCC) and one hybrid algorithm (HCC). In the Indirect Catalogue Creation (ICC), similar customers are grouped together, and then the best catalogue is derived for each subgroup (segment). The second algorithm, called direct catalogue creation, simultaneously tries to identify both a catalogue and its associated customer segment. Finally, the third algorithm, called hybrid catalogue creation, solves the problem by combining elements of the earlier two algorithms [55].

Ester, et al. [58] investigated an alternative problem formulation that they call customer-oriented catalogue segmentation, where the overall utility was measured by the number of customers that had at least a specified minimum interest in the t items in the catalogues and found that the use of the new algorithms significantly enhanced the utility of the catalogues compared to the classical catalogue segmentation algorithms. However, the underlying concept in this study was, in fact, a reproduction of the minimum support in association rule mining, which was first proposed by Agrawal and Srikant [59].

Amiri [60] proposed a two-algorithm model to capture the customer-oriented catalogue segmentation problem. The first one, the Greedy Out algorithm (GO), constructs the catalogues one at a time. Each catalogue is constructed by initially including all of the products and then removing the undesirable products one by one from the catalogue in a greedy fashion so that the number of uncovered customers is minimised. The construction of the second algorithm, Association-Based (AB) catalogue, which likewise builds the catalogues one at a time, is inspired by association rule mining. In the grouping of products in one catalogue, it tries to maximise the association between the products, which is defined as the customer interest relationship. He demonstrated the superior performance of the Greedy Out algorithm relative to both the AB algorithm and Randomised Best Product Fit (RBPF) proposed by Ester, Ge, Jin, and Hu [58]. In another study, a self-adaptive genetic algorithm was proposed by Mahdavi, et al. [61] for designing customer-oriented catalogues in an

e-CRM environment. Namvar, et al. [62] proposed a customer segmentation method based on using a customer lifetime value (LTV) model and a recency, frequency, and monetary (RFM) model, as well as demographic parameters with the aid of data mining tools. First, various combinations of RFM and demographic variables were utilised for clustering in this approach. Second, the optimal clustering was selected using LTV. Finally, in order to create consumer profiles, each section was compared to the other segments in terms of various characteristics. The technique was applied to a dataset from a food chain retailer.

Yousefpoor and Olfat [63] examined the possibilities of current markets using an analytical hierarchy approach. They assumed that items are then represented in the markets via the use of an online catalogue. The customers browse the online catalogues and choose items. They proposed a mathematical approach to optimise the anticipated profit while taking into account the length of viewing. Hsu, et al. [64] proposed a model for a mobile-oriented catalogue (MOC) segmentation problem to improve consumer attractiveness for mobile applications in m-commerce. They utilise query-based learning (QBL) to create MOCs with the goal of attracting the highest number of consumers with the fewest amount of MOCs. Makinde, et al. [65] developed an integrated model for B2B CRM that improves decision making by combining data mining techniques and a genetic algorithm (GA). The approach divides the consumers into the following two groups: repeat customers and shop-and-go customers. For customer classification, a modified data mining—C5.0 was employed, and a GA was utilised to optimise the rules produced by the decision tree algorithm.

3. Problem Statement

Increasing profit margins is a major and essential challenge for commercial organisations to address in today's competitive markets. The approach of selling more products necessitates presenting the products to a greater number of potential customers. When it comes to drawing in more customers, introducing products via Internet websites is an effective technique that results in a larger profit than was anticipated. Because of technological advancements, we can collect a great deal of information on our clients. Companies are utilising a variety of presentation techniques to introduce their goods to consumers.

One of these approaches is to advertise through web pages, which are already being utilised in EC. Every client examines digital catalogues in order to learn about the many features of a company's current product offerings. The various categories of products are presented in a hierarchical structure on the first page of every catalogue, starting with the most important category of the goods. Following that, more information is given under the relevant headings. At the same time, any customers who are interested in a particular product are encouraged to learn more about it by clicking on the table.

By collecting customers' transaction data, their preferences may be obtained. Consider that each catalogue has n products. We assume that a catalogue covers a customer if he or she is interested in at least t items within it. The objective of the problem is to maximise the profit by increasing the number of customers covered by all of these catalogues. Let $C = \{c_1, c_2, c_3, \dots\}$ represent the set of all of the customers, and $P = \{p_1, p_2, p_3, \dots\}$ represent the set of products in the database. $L = \{l_1, l_2, l_3, \dots\}$ is the layer set, and there are numerous screens. In the l -th layer, $S_l = \{s_{l,1}, s_{l,2}, s_{l,3}, \dots\}$. The s -th screen of the l -th layer is made up of catalogues and is denoted by $K_{l,s} = \{k_{l,s,1}, k_{l,s,2}, k_{l,s,3}, \dots\}$. Each catalogue $k_{l,s,k}$ has n items.

Customers will not be bothered to browse through several screens for a product in the MOC segmentation problem, therefore, the first catalogue is given greater importance. As a result, we assign the top weight to the catalogue of the first layer and the first screen. The weights are given to the following catalogues in decreasing order. In order to equalise the screen weight in each layer, we utilise the biggest screen size from all of the layers. The weight of the final screen of the first layer in this design may be less than the weight of the first screen of the second layer. This is a fair technique since swiping the screen five times is more time consuming than touching to reach the second tier. Our object function was

created on the basis of a commission. The compensation given to the m-commerce platform provider is referred to as commissions. As a result, the supplier of the m-commerce platform may optimise the MOCs to maximise the income. For the e-commerce platform, we define a commission as screen commission and layer commission. In order to formulate the problem, sets, indices, parameters, and decision variables described below are employed.

C	Set of customers, $C = \{c_1, c_2, c_3, \dots\}$
P	Set of websites, $P = \{p_1, p_2, p_3, \dots\}$
W	Set of websites, $W = \{w_1, w_2, w_3, \dots\}$
S_w	Set of screens in website w , $S_w = \{s_{w,1}, s_{w,2}, s_{w,3}, \dots\}$
ξ	Size of screen
A	Set of potential size for advertising, $A = \{a_1, a_2, a_3, \dots\}$
J_{cpsa}	Interest of customer c to product p if it is advertised in screen s and at size a
T	Minimum customer interest threshold
M	A big number
B	Available budget
Q_{pwsa}	Cost of advertising product p in website w in screen s and at size a .
x_{pwsa}	Binary variable that equals 1 if product p is advertised in website w in screen s and at size a , else 0,
y_{cw}	Binary variable that equals 1 if customer c is covered by website w , else 0,
φ_c	Binary variable that equals 1 if customer c is covered, else 0,

Based on the above-mentioned assumptions and descriptions, we developed the following mathematical model:

$$\text{Max } Z = \sum_{c \in C} \varphi_c \quad (1)$$

Subject to

$$\sum_{p \in P} \sum_{w \in W} \sum_{s \in S_w} \sum_{a \in A} (x_{pwsa} \cdot Q_{pwsa}) \leq B \quad (2)$$

$$\sum_{s \in S_w} \sum_{a \in A} x_{pwsa} \leq 1, \quad \forall p \in P, w \in W \quad (3)$$

$$\sum_{p \in P} \sum_{a \in A} (x_{pwsa} \cdot S_a) \leq \xi, \quad \forall s \in S_w, w \in W \quad (4)$$

$$\left(\sum_{p \in P} \sum_{s \in S_w} \sum_{a \in A} (J_{cpsa} \times x_{pwsa}) - T \right) \geq M(-1 + y_{cw}), \quad \forall w \in W, c \in C \quad (5)$$

$$\sum_{w \in W} y_{cw} \geq \varphi_c, \quad \forall c \in C \quad (6)$$

$$x_{pwsa}, \varphi_c, y_{cw} \in \{0, 1\}, \quad \forall s \in S_w, a \in A, w \in W, c \in C, p \in P \quad (7)$$

Equation (1) maximises the number of potential customers absorbed by the advertising strategy. Equation (2) controls the available budget for advertising. Equation (3) guarantees that each product cannot be advertised on each website more than one time. Equation (4) ensures that the sum of advertisements on each page cannot exceed the maximum space that is available on that page. Equations (5) and (6) control covering, or not covering, a customer by products advertised on different websites. Equation (7) satisfies that decision variables are integer.

4. Review of Metaheuristic Algorithms

The complexity of many real-world problems necessitates rapid computation methods to find near-optimal solutions [66–70]. As a result, decision makers face a significant challenge in identifying an appropriate solution approach. It has been proven that metaheuristics can find a suitable solution with less computation effort [39,71]. “A metaheuristic is an algorithm designed to solve approximately a wide range of hard optimisation problems without having to deeply adapt to each problem” [72]. In recent years, researchers

have focused considerable attention on finding near-optimal solutions to various real-world optimisation problems using metaheuristic methods [73–75]. As a result, numerous metaheuristic algorithms are being developed and applied in various fields [76,77]. The concept of metaheuristics is explained first in the following subsections. Following that, various classifications of metaheuristic algorithms are discussed. Finally, for this study, some metaheuristic algorithms that have recently been presented in the literature are chosen.

4.1. Concepts of Metaheuristic Algorithms

Meta (from the Greek meta—μετά) is a prefix used in English, and in metaheuristic indicates that an algorithm is “higher level” heuristic [78]. Some of the characteristics of metaheuristic algorithms [79] include the following: (1) seeking a near-optimal solution rather than specifically trying to find the exact optimal solution, (2) having no rigorous proof of convergence to the optimal solution, and (3) being computationally faster than exhaustive search. These iterative methods frequently employ stochastic operations in their search process to modify one or more initial candidate solutions (usually generated by the random sampling of the search space) [80,81].

Diversification and intensification are two standard features seen across all metaheuristics [82]. Diversification refers to how well the algorithm diversifies the solutions in the search space. The intensification gives the metaheuristic a local search behaviour and searches the current best solutions in order to reach the best candidates [83]. The majority of metaheuristic algorithms follow common steps. Therefore, they can be stated as a general framework, as shown in Algorithm 1 [84], as follows:

Algorithm 1. A Generic Metaheuristic Framework.

```

Set counter for iterations  $it = 0$ .
Randomly generate  $N$  solutions in the search space ( $SP$ ).
Evaluating the generated solutions.
Find the best solution  $x^*$ 
While stop condition met Do
    Selecting a predefined number of solutions ( $P_t$ ) from the solution pool ( $SP$ ) using selection strategies
    Generate  $\lambda$  new solutions from  $P_t$ .
    Selecting a predefined number of solutions ( $R_t$ ) from the solution pool ( $SP$ ) using a replacement plan
    Updating solutions in  $SP$ 
    Evaluate the fitness of members in  $SP$ 
    Find the best solutions in  $SP$  and update  $x^*$ 
End while
Present the best-obtained solution as  $x^*$ 

```

4.2. Classification of Metaheuristic Techniques

There are various ways to categorise metaheuristic algorithms [85]. One of the common ways to categorise metaheuristic algorithms is based on the number of solutions in the algorithm. Single-solution metaheuristics apply generation and replacement procedures to a single solution. In contrast, population-based metaheuristics use more than one initial solution and apply generation and replacement procedures to a set of solutions. Categorising metaheuristic algorithms into two groups of swarm intelligence algorithms and evolutionary algorithms (EAs) is another common type of metaheuristic classification [86].

Evolutionary algorithms optimise a problem based on the evolutionary principle of survival of the fittest. An EA begins with a randomly generated population of individuals (solutions). At each generation, the EA modifies the key characteristics of the current population in order to form a new population that will be selected based on the natural selection principle. Swarm intelligence algorithms are inspired by the collective behaviour of a group of animals or insects when searching for food. At each iteration, the swarm intelligence algorithm constructs the solutions based on the historical information attained

from previous generations. The other most important ways of classifying metaheuristics are as follows [85,87]:

1. *Deterministic versus stochastic methods*: Deterministic methods follow a definite trajectory from the random initial solution(s). Therefore, they are sometimes referred to as trajectory methods. Stochastic methods (discontinuous methods) allow probabilistic jumps from the current solution(s) to the next.
2. *Greedy versus non-greedy methods*: Greedy algorithms usually search in the neighbourhood of the current solution and immediately move to a better solution when it is found. This behaviour often leads to a local optimum. Non-greedy methods either hold out for some iterations before updating the solution(s) or have a mechanism to backtrack from a local optimum. However, for convex problems, greedy behaviour is the optimum strategy.
3. *Memory usage versus memoryless methods*: Memory-based methods record past solutions and their trajectories and use them to direct search.
4. *One versus various neighbourhood methods*: Some metaheuristics, such as simulated annealing and tabu search, only allow a limited set of moves from the current solution. However, many metaheuristics employ operators and parameters to allow multiple neighbourhoods. For example, particle swarm optimisation achieves this through various swarm topologies.
5. *Dynamic versus static objective function*: Metaheuristics that update the objective function depending on the current search requirements are classified as dynamic. Other metaheuristics use their operators to control search.

4.3. Metaheuristic Algorithms Used in This Study

In this study, a comprehensive analysis of the performance of the metaheuristics on this problem was conducted. Therefore, ten different metaheuristics were selected, including The ant lion optimiser (ALO) [44], Dragonfly algorithm (DA) [45], Grasshopper optimisation algorithm (GOA) [46], Harris hawks optimisation (HHO) [47], Moth-flame optimisation algorithm (MFO) [48], Multi-verse optimiser (MVO) [49], sine cosine algorithm (SCA) [50], Salp Swarm Algorithm (SSA) [51], The whale optimisation algorithm (WOA) [52], and Grey wolf optimiser (GWO) [53]. These algorithms and their developers are listed in Table 1.

Table 1. A list of metaheuristic algorithms used in this study.

The Name of the Algorithm [Ref.]	ACR	Author(s)	Year
Ant Lion Optimiser [44]	ALO	S Mirjalili	2015
Dragonfly Algorithm [45]	DA	S Mirjalili	2016
Grasshopper Optimisation Algorithm [46]	GOA	S Saremi, S Mirjalili, A Lewis	2017
Harris Hawks Optimisation [47]	HHO	AA Heidari, S Mirjalili, H Faris, I Aljarah, M Mafarja, H Chen	2019
Moth-Flame Optimisation Algorithm [48]	MFO	S Mirjalili	2015
Multi-Verse Optimiser [49]	MVO	S Mirjalili, SM Mirjalili, A Hatamlou	2016
Sine Cosine Algorithm [50]	SCA	S Mirjalili	2016
Salp Swarm Algorithm [51]	SSA	S Mirjalili, AH Gandomi, SZ Mirjalili, S Saremi, H Faris, SM Mirjalili	2017
Whale Optimisation Algorithm [52]	WOA	S Mirjalili, A Lewis	2016
Grey Wolf Optimiser [53]	GWO	S Mirjalili, SM Mirjalili, A Lewis	2014

- *Ant Lion Optimiser (ALO) [44]* is a recent metaheuristic that mathematically models the interaction of ants and antlions in nature. An optimisation algorithm has been developed to solve the optimisation problems that take into account random ant walk, building traps, entrapment of ants in traps, catching prey, and re-building traps [88]. The pseudocode of ALO is presented in Algorithm 2.

Algorithm 2. Ant Lion Optimiser (ALO).

Initialisation of population $X_i(i=1, 2, \dots, N)$, set total number of ants and antlions

Initialisation of generation $\xi = 1$;

Calculate the fitness of population.

Select the elite antlion E.

while generation $\xi \leq \xi_{\max}$

 for each Ant_i

 Select an antlion \mathcal{A} based on roulette wheel

 Update c and d using: $c^\xi = \frac{c^\xi}{1}$ and $d^\xi = \frac{d^\xi}{1}$

 Create a random walk and normalise it using:

$X(t) = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \dots, \text{cumsum}(2r(t_{\xi_{\max}}) - 1)]$

$X_i^\xi = \frac{(X_i^\xi - a_i) \times (d_i - c_i^\xi)}{(d_i^\xi - a_i)} + c_i$

 Update the location of Ant_i using $Ant_i^t = \frac{R_A^t + R_E^t}{2}$

 end

 Update the fitness value of all agents

 Merge, sort, and send the best ants to the next generation

 Update the elite ant if an antlion is better than the elite agent

$\xi = \xi + 1$

end

Return E

- **Dragonfly Algorithm (DA)** [45] is primarily inspired by static and dynamic swarming behaviours. These two swarming behaviours are very similar to the two main phases of metaheuristic optimisation, which are exploration and exploitation. The main goal of the exploration phase is for the dragonflies to form subswarms and fly over different areas in a static swarm. The dragonflies in the static swarm, on the other hand, fly in larger swarms and in one direction, which is advantageous during the exploitation phase. The pseudocode of DA is presented in Algorithm 3.

Algorithm 3. Dragonfly Algorithm (DA).

Initialisation of population $X_i(i=1, 2, \dots, N)$

Initialisation of generation $\xi = 1$;

Calculate the fitness of population.

Select the elite antlion E.

while generation $\xi \leq \xi_{\max}$

 Calculate the objective values of all dragonflies

 Update the food source and enemy

 Update $w, s, a, c, f,$ and e

 Calculate S, A, C, F, and E using:

$S_i = - \sum_{j=1}^N X - X_j, A_i = \frac{\sum_{j=1}^N V_j}{N}, C_i = \frac{\sum_{j=1}^N X_j}{N} - X, F_i = X^+ - X, E_i = X^- + X$

 Update neighbouring radius

if a dragonfly has at least one neighbouring dragonfly

 Update velocity vector using $\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t$

 Update position vector using $X_{t+1} = X_t + \Delta X_{t+1}$

else

 Update position vector using $X_{t+1} = X_t + Levy(d) \times X_t$

end

Check the boundaries of variables

 Increment generation $\xi = \xi + 1$

end

- Grasshopper Optimisation Algorithm (GOA) was proposed by Saremi, Mirjalili and Lewis [46], which is a metaheuristic optimisation method and is inspired by the group behaviour of grasshoppers. This method imitates and simulates the behaviour of grasshoppers in nature and their group movement towards food sources [65]. The capability of GOA in tackling complex problems has been proven by numerous studies [89–91]. The pseudocode of MFO is presented in Algorithm 4.

Algorithm 4. Grasshopper Optimisation Algorithm (GOA).

Initialisation of population $X_i (i=1, 2, \dots, H)$, the boundaries, $\xi_{max}, K_{max}, K_{min}$
 Initialisation of generation $\xi = 1$;
while generation $\xi \leq \xi_{max}$
 Update the value of K by using the $K = K_{max} - \xi * \frac{K_{max} - K_{min}}{\xi_{max}}$
 For each position
 Normalised the distance between grasshoppers.
 Update the current position of the individual by:

$$X_i^k = K \left(\sum_{j=1, j \neq i}^H K \frac{ub_k - lb_k}{2} S(|x_j^k - x_i^k|) \frac{x_j - x_i}{d_{ij}} \right) + T_k$$

 The current search agent will be brought back if it is outside the boundaries.
 If a better solution becomes available, A is updated.
 end
 $\xi = \xi + 1$
end
Return \mathcal{A}

- Harris hawks optimisation (HHO) [47] is a swarm-based optimisation method. HHO's main concept is to mimic the action and reaction of a hawk's team collaboration hunting in nature and prey escaping to discover solutions to the single-objective problem. The pseudocode of HHO is presented in Algorithm 5.

Algorithm 5. Harris hawks optimisation (HHO).

The initialisation of population $X_i, i = 1, \dots, N$;
 Initialisation of generation $\xi = 1$;
while generation $\xi \leq \xi_{max}$
 Calculate the fitness values of hawks
 for each hawk X_i
 Update the initial energy \mathcal{E}_0 and jump strength \mathcal{J}
 Update the \mathcal{E} using $\mathcal{E} = 2\mathcal{E}_0(1 - t/\mathcal{T})$
 if ($|\mathcal{E}| \geq 1$)
 Update the location vector using #
 else
 if ($r \geq 0.5$ and $|\mathcal{E}| \geq 0.5$) then
 Update the \mathcal{E} using $X(t+1) = \Delta X(t) - \mathcal{E} |J X_{rabbit}(t) - X(t)|$
 elseif ($r \geq 0.5$ and $|\mathcal{E}| < 0.5$) then
 Update the \mathcal{E} using $X(t+1) = X_{rabbit}(t) - \mathcal{E} |\Delta X(t)|$
 elseif ($r < 0.5$ and $|\mathcal{E}| \geq 0.5$) then
 Update the \mathcal{E} using $X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases}$
 elseif ($r < 0.5$ and $|\mathcal{E}| < 0.5$) then
 Update the \mathcal{E} using $X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases}$
 end
 end
 end
 $\xi = \xi + 1$
end
Return X_{rabbit}

- Moth-flame optimisation algorithm (MFO) [48] assumes that the candidate solutions are moths and that the variables of the problem are the positions of the moths in space. Both moths and flames are viable solutions. The distinction is in how we treat and update them in each iteration. The moths are actual search agents that move around the search space, whereas the flames are the best position of the moths that have so far been obtained. In other words, the flames can be thought of as flags or pins that the moths drop while searching the search space. As a result, each moth searches for a flag (flame) and updates it if a better solution is discovered. A moth's best solution is never lost with this mechanism. The pseudocode of MFO is presented in Algorithm 6.

Algorithm 6. Moth-flame optimisation algorithm (MFO).

```

The initialisation of Moth population  $\mathcal{M}$ ;
Calculate the fitness of the  $\mathcal{M} - \mathcal{OM}$ 
while generation  $\xi \leq \xi_{\max}$ 
  if  $\xi == 1$  then
     $\mathcal{F} \leftarrow \text{sorted}(\mathcal{M})$ ;
     $\mathcal{OF} \leftarrow \text{sorted}(\mathcal{OM})$ ;
  else
     $\mathcal{F} \leftarrow \text{sorted}(\mathcal{F}_{t-1}, \mathcal{M}_t)$ ;
     $\mathcal{OF} \leftarrow \text{sorted}(\mathcal{OF}_{t-1}, \mathcal{OM}_t)$ ;
  end
  for  $i$ -th
    Calculate D for the corresponding Moth and Flame;
    Update Moth ( $i$ )
  end
   $\xi = \xi + 1$ 
end
 $\mathcal{F} \leftarrow \text{sorted}(\mathcal{F}_{t-1}, \mathcal{M}_t)$ ;
Return the best solution

```

- Multi-verse optimiser (MVO) [49] is inspired by three cosmological concepts, including white holes, black holes, and wormholes. These three concepts' mathematical models are created to perform exploration, exploitation, and local search, in that order. The pseudocode of MVO is presented in Algorithm 7.

Algorithm 7. Multi-verse optimiser (MVO).

```

The initialisation of population  $X_i, i = 1, \dots, N$  ;
while generation  $\xi \leq \xi_{\max}$ 
  Calculate the fitness of universes
  for each universe $_i$ 
    Update WEP and TDR
    for each object $_j$ 
       $\varphi_1 = \text{rand}(0,1)$ 
      if  $\varphi_1 < \text{NI}(U_i)$ 
        White_hole_index = Roulette_wheel_selection(-NI)
         $U(\text{Black\_hole\_index}, j) = \text{SU}(\text{White\_hole\_index}, j)$ 
      end
       $\varphi_2 = \text{rand}(0,1)$ 
      if  $\varphi_2 < \text{WEP}$ 
         $\varphi_3 = \text{rand}(0,1)$ 
         $\varphi_4 = \text{rand}(0,1)$ 
        if  $\varphi_3 < 0.5$ 
           $U(i, j) = \text{Best}_{\text{universe}(j)} + \text{TDR} \times ((\text{UB}(j) - \text{LB}(j)) \times \varphi_4 + \text{LB}(j))$ ;
        else
           $U(i, j) = \text{Best}_{\text{universe}(j)} - \text{TDR} \times ((\text{UB}(j) - \text{LB}(j)) \times \varphi_4 + \text{LB}(j))$ ;
        end
      end
    end
  end
   $\xi = \xi + 1$ 
end
Return the best universe

```

- **Sine Cosine Algorithm (SCA)** [50] is an optimisation technique used to solve optimisation problems. Using a mathematical model based on sine and cosine functions, the SCA generates multiple initial random candidate solutions and requires them to fluctuate outwards or towards the best solution. Several random and adaptive variables are also integrated into this algorithm in order to emphasise exploration and exploitation of the search space at various optimisation milestones. The pseudocode of SCA is presented in Algorithm 8.

Algorithm 8. Sine Cosine Algorithm (SCA).

The initialisation of population $X_i, i = 1, \dots, N$;
 Initialisation of generation $\xi = 1$;
while generation $\xi \leq \xi_{max}$
 Calculate the fitness of each solution
 Update the best solution obtained so far
 Update r_1, r_2, r_3, r_4
 Update the position of the search agents using

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 \geq 0.5 \end{cases}$$

 $\xi = \xi + 1$
end
 Return the best solution

- **Salp Swarm Algorithm (SSA)** [51] is a population-based optimisation method. The SSA's behaviour can be demonstrated by computing it with the salp chain in search of optimal food sources (i.e., the target of this swarm is a food source in the search space called F) [92]. The individuals (i.e., salps) are classified as leaders or followers in SSA based on their position in the chain. The chain begins with a leader, and the followers follow to guide their movements. The pseudocode of SSA is presented in Algorithm 9.

Algorithm 9. Salp Swarm Algorithm (SSA).

The initialisation of population $X_i, i = 1, \dots, N$;
 Calculate the fitness of each salp using the objective function.
 Initialisation of generation $\xi = 1$;
 X^* is the best solution.
 Update r_1 using $r_1 = 2e^{-\left(-\frac{4a}{A}\right)^2}$
while generation $\xi \leq \xi_{max}$
 for each salp
 if $m == 1$
 Update the position of the leading salp using $z_n^1 = \begin{cases} P_n + r_1((u_n - l_n)r_2 + l_n) & r_3 \geq 0 \\ P_n - r_1((u_n - l_n)r_2 + l_n) & r_3 < 0 \end{cases}$
 end if
 Update the position of the follower salp using $z_n^m = \frac{1}{2}(z_n^m + z_n^{m-1})$
 end
 Calculate the fitness of salp using the objective function.
 Update X^*
 end
 $\xi = \xi + 1$
end
 Return the best solution

- Whale Optimisation Algorithm (WOA) [52] is a swarm-based metaheuristic algorithm that is based on the bubble-net hunting manoeuvre technique of humpback whales. This algorithm includes three operators that simulate the humpback whale's search for prey, encircling prey, and bubble-net foraging behaviour. The pseudocode of WOA is presented in Algorithm 10.

Algorithm 10. Whale Optimisation Algorithm (WOA).

The initialisation of population $X_i, i = 1, \dots, N$;
 Calculate the fitness of each search agent using the objective function.
 Initialisation of generation $\xi = 1$;
 X^* is the best solution.
while generation $\xi \leq \xi_{max}$
 for each agent
 Update a, A, C, l, p
 if $p < 0.5$
 if $|A| < 1$
 Update the position of the current search using $\vec{D} = |\vec{C} \cdot \vec{X}^*(\xi) - \vec{X}(\xi)|$
 else if $|A| \geq 1$
 Select a random search agent X_{rand}
 Update the position of the current search using $\vec{X}(\xi + 1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D}$
 end
 else if $p \geq 0.5$
 Update the position of the current search using $\vec{X}(\xi + 1) = \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(\xi)$
 end
 end
 The current search agent will be brought back if it is outside the boundaries.
 Calculate the fitness of each search agent using the objective function.
 Update X^*
 $\xi = \xi + 1$
end
 Return the best solution

- *Grey Wolf Optimiser (GWO)* [53] simulates the natural leadership hierarchy and hunting mechanism of grey wolves. For simulating the leadership hierarchy, the following four types of grey wolves are used: alpha, beta, delta, and omega. In addition, the following three main hunting steps are implemented in order to optimise performance: searching for prey, encircling prey, and attacking prey. The pseudocode of GWO is presented in Algorithm 11.

Algorithm 11. Grey Wolf Optimiser (GWO).

The initialisation of population $X_i, i = 1, \dots, N$;
 The initialisation of a, \mathcal{A} , and \mathcal{C} (\mathcal{A} and \mathcal{C} are coefficient vectors);
 Calculate the fitness of each search agent using the objective function.
 $X_\alpha =$ the best search agent
 $X_\beta =$ the second-best search agent
 $X_\gamma =$ the third best search agent
while generation $\xi \leq \xi_{max}$
 for each search agent
 Randomly initialise r_1 and r_2 (random vectors with values from [0,1]);
 Update the position of the current search agent by $\vec{X}(t + 1) = \vec{X}_1 + \vec{X}_2 + \vec{X}_3/3$
 end
 Update a, \mathcal{A} and \mathcal{C} ;
 Calculate the fitness of all search agent
 Update $X_\alpha, X_\beta, X_\gamma$
 $\xi = \xi + 1$;
end
 Return X_α

5. Computational Results

The main aim of this section is to assess the proposed solution method using test instances in multiple sizes. All of the algorithms were tested on a laptop with an Intel® Core i7 processor running at 2.9 GHz and 16.0 GB of RAM. The data set is described in Section 5.1. The different scenarios are considered to assess the proposed method from a practical perspective. Then, numerical examples and analysis are presented in Section 5.2. The mean of the normalised objective function (OF) is computed to assess the suggested algorithm's performance reliability. The following is the formula for calculating the normalised \mathcal{OF} value for the algorithm [37]:

$$\text{Normalized } \mathcal{OF} = \frac{\text{Algorithm}_{sol} - \text{Algorithm}_{Worst}}{\text{Algorithm}_{Best} - \text{Algorithm}_{Worst}} \quad (8)$$

where Algorithm_{sol} is the objective value obtained from each algorithm, and Algorithm_{Best} and Algorithm_{Worst} are the best and the worst solutions obtained among all of the methods, respectively. The highest normalised OF values are preferable. For an unbiased comparison, we have adjusted the parameters of the algorithms based on their primary sources.

5.1. Data for Numerical Tests

The problem instances used in the experiments are generated at random. Problem instances are defined by parameters, such as the number of customers, the number of products, the number of websites, and the number of screens. As a result, three different levels are considered for the number of customers, with the first type of instance having 200 customers, the second having 400, and the third having 1000 customers. Similarly, the following three different levels of product number are considered: 20, 40, and 100 products. Furthermore, it is assumed in these cases that there are three levels for the number of websites, 10, 20, and 50. While different levels of screen number can be considered in this study, only a single value is considered for this parameter. Table 2 contains parameter values for the various parameters.

Table 2. The problem set's parameter settings.

Control Parameter	Value
Number of customers	200, 400, 1000
Number of products	20, 40, 100
Number of websites	10, 20, 50
Number of screens	2

5.2. Discussion of the Performance of the Proposed Algorithm

We compared various optimisation algorithms. All of the algorithms have the same stopping condition and are terminated after a predetermined number of iterations to ensure a fair comparison. Each algorithm was run ten times for each instance, and the results obtained using Equation (8) were normalised. Table 2 shows the outcomes of the algorithm mentioned above. The results show that MFO outperforms the other metaheuristics used in this study. Except for 11 instances, MFO produced superior results when compared to the other methods. With a value of 77% for the normalised objective function, ALO has the best performance after MFO. This algorithm could produce the best results for 11 different problems, while for others could find an acceptable solution. MVO and SSA rank third and fourth in terms of performance, with 68% and 57% normalised objective function, respectively. In contrast, HHO performed the worst among the methods used in this study, with an average of 16% for the normalised objective function. SCA had the worst performance after HHO, with an average of 24% for the normalised objective function. Table 3 shows that almost all of the metaheuristic algorithms could maintain their performance in small size problems, whereas the difference in large instances is more sensible.

Table 3. The normalised objective function for each algorithm and each instance.

	Website	Screen	Customer	Product	ALO	DA	GOA	HHO	MFO	MVO	SCA	SSA	WOA	GWO
1	10	2	200	20	66%	41%	22%	15%	86%	70%	33%	54%	44%	29%
2	20	2	200	20	77%	58%	34%	19%	99%	80%	44%	72%	62%	33%
3	50	2	200	20	100%	100%	100%	60%	100%	100%	100%	100%	95%	100%
4	10	2	400	20	74%	45%	24%	17%	91%	67%	27%	53%	35%	30%
5	20	2	400	20	76%	55%	23%	11%	89%	79%	26%	62%	47%	24%
6	50	2	400	20	75%	95%	82%	35%	100%	97%	80%	100%	90%	82%
7	10	2	1000	20	79%	40%	11%	7%	84%	71%	16%	49%	26%	18%
8	20	2	1000	20	85%	47%	24%	13%	84%	67%	25%	59%	45%	27%
9	50	2	1000	20	77%	82%	57%	27%	100%	97%	44%	90%	77%	62%
10	10	2	200	40	93%	46%	19%	15%	83%	66%	23%	53%	40%	29%
11	20	2	200	40	66%	44%	18%	17%	75%	60%	22%	51%	40%	25%
12	50	2	200	40	72%	59%	23%	15%	85%	70%	15%	64%	44%	27%
13	10	2	400	40	85%	40%	11%	10%	68%	54%	19%	48%	34%	19%
14	20	2	400	40	92%	48%	16%	10%	79%	64%	22%	60%	41%	21%
15	50	2	400	40	71%	59%	20%	20%	77%	70%	16%	56%	43%	21%
16	10	2	1000	40	85%	45%	17%	13%	72%	68%	17%	47%	27%	23%
17	20	2	1000	40	84%	48%	16%	11%	72%	56%	11%	48%	30%	18%
18	50	2	1000	40	76%	51%	14%	14%	72%	66%	7%	48%	35%	16%
19	10	2	200	100	73%	47%	18%	10%	74%	54%	20%	49%	34%	23%
20	20	2	200	100	75%	50%	16%	13%	78%	63%	9%	51%	40%	17%
21	50	2	200	100	59%	57%	12%	7%	83%	52%	9%	45%	49%	15%
22	10	2	400	100	89%	39%	17%	13%	68%	60%	16%	43%	31%	23%
23	20	2	400	100	65%	55%	15%	10%	68%	59%	6%	41%	33%	17%
24	50	2	400	100	82%	58%	16%	9%	76%	59%	9%	52%	39%	18%
25	10	2	1000	100	65%	40%	10%	7%	60%	50%	8%	40%	26%	17%
26	20	2	1000	100	68%	52%	18%	19%	70%	57%	14%	50%	40%	22%
27	50	2	1000	100	72%	69%	20%	17%	82%	72%	14%	60%	55%	25%
					77%	54%	25%	16%	81%	68%	24%	57%	44%	29%

Note: The bold cells are the best results.

Additionally, in order to illustrate the metaheuristics’ overall performance for the problem under consideration in this study, Figure 1 displays the average of the normalised objective function. On the basis of this figure, it is clear that a considerable gap exists between certain metaheuristic algorithms. For example, there is a 65% difference in the average performance between MFO, the best algorithm in this study, and HHO, the worst algorithm in this study.

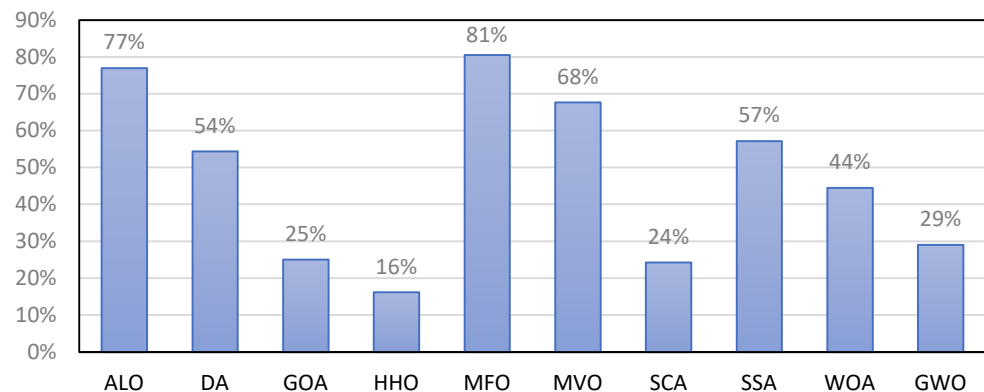


Figure 1. Average normalised objective functions for different algorithms.

In addition, in order to demonstrate the stability of the employed metaheuristic algorithms, Table 4 demonstrates the best and worst solutions obtained by each algorithm for the various problems. According to Table 4, HHO obtained the majority of the worst solutions. On the other hand, ALO obtained all of the best solutions, except for three. In contrast, ALO performance came in second place based on the normalised objective function. It demonstrates that, while ALO was able to find the best solution, it was unable to maintain its performance across multiple repetitions. In Table 4, the red blocks represent the worst solutions obtained by all of the algorithms, while the green blocks represent the best solutions obtained.

Table 4. Best and worst solutions obtained by metaheuristics.

	ALO		DA		GOA		HHO		MFO		MVO		SCA		SSA		WOA		GWO	
	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best
1	38	49	37	47	35	38	32	37	48	53	44	51	37	41	39	46	34	49	37	39
2	45	55	46	54	45	49	43	47	54	55	49	55	46	52	50	54	47	53	46	48
3	55	55	55	55	55	55	53	55	55	55	55	55	55	55	55	55	54	55	55	55
4	74	95	67	83	65	71	59	69	87	95	79	87	66	72	73	86	68	78	66	75
5	88	103	89	100	85	88	81	87	100	105	96	105	84	90	92	101	86	97	85	89
6	102	105	104	105	104	105	101	105	105	105	104	105	103	105	105	105	104	105	104	105
7	192	219	159	203	147	157	144	156	201	217	183	210	150	167	164	192	151	181	151	168
8	235	249	207	231	198	212	189	209	232	244	212	239	200	210	218	230	206	223	202	209
9	247	254	248	254	247	252	242	249	254	254	253	254	245	249	251	254	248	253	248	250
10	72	82	58	68	49	54	44	52	71	82	64	76	50	57	60	72	49	68	53	61
11	77	106	73	92	70	73	64	77	89	102	77	97	71	76	78	95	76	86	72	77
12	104	124	109	118	100	104	96	107	116	124	104	121	99	103	109	120	103	112	101	107
13	132	154	100	124	87	95	83	100	125	139	114	131	88	108	105	136	93	117	92	101
14	172	185	138	161	130	136	123	135	166	180	151	175	128	146	142	174	140	161	130	141
15	192	233	208	215	187	195	181	195	218	225	200	223	185	197	200	219	195	211	189	195
16	279	327	229	263	203	212	182	216	270	307	257	302	196	218	223	278	201	236	210	226
17	391	435	330	371	293	318	283	309	375	407	343	408	285	318	337	381	314	350	303	329
18	458	543	476	506	440	455	433	459	502	521	483	528	431	445	469	503	457	488	441	459
19	59	100	62	81	54	64	49	59	82	94	66	83	54	64	63	83	62	72	59	63
20	92	129	91	109	84	90	80	89	110	124	92	124	79	91	98	118	92	103	85	90
21	133	162	134	158	129	133	127	133	150	164	138	159	127	136	137	152	140	149	130	136
22	131	176	115	129	98	114	88	104	142	155	124	154	98	111	117	138	106	128	105	115
23	158	236	184	206	151	170	147	162	191	218	190	204	146	159	169	194	163	188	158	165
24	254	306	269	289	241	252	238	254	283	297	270	288	238	251	261	290	254	281	244	261
25	234	388	250	313	227	248	218	264	300	348	273	340	225	251	251	310	247	279	239	261
26	365	516	412	452	355	381	338	407	447	484	395	478	333	377	395	456	384	423	365	383
27	582	695	628	674	570	593	554	588	654	684	630	680	559	596	617	678	612	659	578	600

Note: the red blocks represent the worst solutions obtained by all of the algorithms, while the green blocks represent the best solutions obtained.

6. Conclusions and Future Work

Because of advancements in Internet technology, businesses can now monitor the behaviour and performance of their customers. In order to regulate their electronic customer relationship management, online shopping websites use optimisation techniques to analyse and interact with their customers (e-CRM). A comprehensive metaheuristic analysis is proposed in this study to decide the items displayed on each page of a website. Based on the available information about consumer behaviour, the results showed that MFO performance is acceptable.

Furthermore, based on these findings, it is possible to conclude that metaheuristic algorithms can provide appealing opportunities for decision makers to obtain good answers to challenging e-CRM problems. Marketing managers will benefit from this useful tool as well, as it allows them to quickly search for important information based on consumer transaction data and modify their advertising strategies. As a result, they can develop marketing programmes that boost sales and profits in a short period of time. Based on this, the findings revealed that, in order to achieve the aforementioned objectives, many e-customer relationship management systems in businesses must create their own customer profiles that include a set of their most important information.

Furthermore, because individuals and businesses without access to the Internet and related technologies are unable to benefit from the electronic services provided, they may gradually lose competitiveness in global markets, which e-CRM in the context of information and communication technology aids in. Moreover, its emergence as the most effective tool for gaining a competitive advantage through customer attraction can be advantageous.

Companies that use e-commerce to brand themselves in a competitive environment while creating a new distribution channel in a virtual space, on the other hand, can relate to the customer more than they could before. As a result, it appears that many changes in the current problems are required for all businesses in order to successfully use modern e-

commerce settings, and some additional realistic assumptions, such as random parameters for page availability, may exist. To make the problem more realistic, using fuzzy parameters in the model can accurately depict real-world conditions. Finally, using other metaheuristic algorithms, or hybridising metaheuristic algorithms with deep learning methods, can be viewed as an exciting future research topic that should be pursued further.

Author Contributions: Conceptualization, R.Y. and M.J.T.; methodology, R.Y.; software, R.Y.; validation, R.Y.; formal analysis, R.Y.; investigation, R.Y.; resources, R.Y.; data curation, R.Y.; writing—original draft preparation, R.Y.; writing—review and editing, M.J.T., M.M.P. and S.S.H.; visualization, R.Y.; supervision, M.J.T., M.M.P. and S.S.H.; project administration, M.J.T., M.M.P. and S.S.H.; funding acquisition, R.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in the study is available from the authors and can be shared upon reasonable requests.

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

References

- Anastasei, B.; Dospinescu, N.; Dospinescu, O. Understanding the Adoption of Incentivized Word-of-Mouth in the Online Environment. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 992–1007. [\[CrossRef\]](#)
- Vieira, J.; Frade, R.; Ascenso, R.; Prates, I.; Martinho, F. Generation Z and Key-Factors on E-Commerce: A Study on the Portuguese Tourism Sector. *Adm. Sci.* **2020**, *10*, 103. [\[CrossRef\]](#)
- Shams, G.; Rehman, M.A.; Samad, S.; Oikarinen, E.-L. Exploring customer's mobile banking experiences and expectations among generations X, Y and Z. *J. Financ. Serv. Mark.* **2020**, *25*, 1–13. [\[CrossRef\]](#)
- Akhtar, M.-J.; Ahmad, Z.; Amin, R.; Almotiri, S.-H.; Ghamdi, M.-A.-A.; Aldabbas, H. An Efficient Mechanism for Product Data Extraction from E-Commerce Websites. *Comput. Mater. Contin.* **2020**, *65*, 2639–2663. [\[CrossRef\]](#)
- Shafiee, M.M.; Bazargan, N.A. Behavioral customer loyalty in online shopping: The role of e-service quality and e-recovery. *J. Theor. Appl. Electron. Commer. Res.* **2018**, *13*, 26–38. [\[CrossRef\]](#)
- Dospinescu, O.; Dospinescu, N.; Bostan, I. Determinants of e-commerce satisfaction: A comparative study between Romania and Moldova. *Kybernetes* **2022**, *51*, 1–17. [\[CrossRef\]](#)
- Bekamiri, H.; Ghasempour Ganji, S.F.; Simonetti, B.; Seno, S.A.H. A New Model to Identify the Reliability and Trust of Internet Banking Users Using Fuzzy Theory and Data-Mining. *Mathematics* **2021**, *9*, 916. [\[CrossRef\]](#)
- Shokouhyar, S.; Shokoohyar, S.; Raja, N.; Gupta, V. Promoting fashion customer relationship management dimensions based on customer tendency to outfit matching: Mining customer orientation and buying behaviour. *Int. J. Appl. Decis. Sci.* **2021**, *14*, 1–23. [\[CrossRef\]](#)
- Hassani, H.; Huang, X.; Silva, E. Digitalisation and big data mining in banking. *Big Data Cogn. Comput.* **2018**, *2*, 18. [\[CrossRef\]](#)
- Danyali, A.A. Factors influencing customers' change of behaviors from online banking to mobile banking in Tejarat Bank, Iran. *J. Organ. Chang. Manag.* **2018**, *31*, 1226–1233. [\[CrossRef\]](#)
- Hajiheydari, N.; Ashkani, M. Mobile application user behavior in the developing countries: A survey in Iran. *Inf. Syst.* **2018**, *77*, 22–33. [\[CrossRef\]](#)
- Hamidi, H.; Safareeyeh, M. A model to analyze the effect of mobile banking adoption on customer interaction and satisfaction: A case study of m-banking in Iran. *Telemat. Inform.* **2019**, *38*, 166–181. [\[CrossRef\]](#)
- Hosseini, S.M.; Paydar, M.M. Discount and advertisement in ecotourism supply chain. *Asia Pac. J. Tour. Res.* **2021**, *26*, 668–684. [\[CrossRef\]](#)
- Bilgihan, A.; Okumus, F.; Nusair, K.; Bujisic, M. Online experiences: Flow theory, measuring online customer experience in e-commerce and managerial implications for the lodging industry. *Inf. Technol. Tour.* **2014**, *14*, 49–71. [\[CrossRef\]](#)
- Svobodová, Z.; Rajchlová, J. Strategic Behavior of E-Commerce Businesses in Online Industry of Electronics from a Customer Perspective. *Adm. Sci.* **2020**, *10*, 78. [\[CrossRef\]](#)
- Chang, H.H.; Wang, H.W. The moderating effect of customer perceived value on online shopping behaviour. *Online Inf. Rev.* **2011**, *35*, 333–359. [\[CrossRef\]](#)
- Sheng, J. Being active in online communications: Firm responsiveness and customer engagement behaviour. *J. Interact. Mark.* **2019**, *46*, 40–51. [\[CrossRef\]](#)
- Buraga, S.-C.; Amariei, D.; Dospinescu, O. An OWL-Based Specification of Database Management Systems. *Comput. Mater. Contin.* **2022**, *70*, 5537–5550. [\[CrossRef\]](#)
- Dospinescu, O.; Dospinescu, N. Workaholism in IT: An Analysis of the Influence Factors. *Adm. Sci.* **2020**, *10*, 96. [\[CrossRef\]](#)

20. Dospinescu, O.; Anastasiei, B.; Dospinescu, N. Key Factors Determining the Expected Benefit of Customers When Using Bank Cards: An Analysis on Millennials and Generation Z in Romania. *Symmetry* **2019**, *11*, 1449. [[CrossRef](#)]
21. Junfang, L.; Shan, C. Design of Sino-Japanese cross border e-commerce platform based on FPGA and data mining. *Microprocess. Microsyst.* **2021**, *80*, 103360. [[CrossRef](#)]
22. Pan, H.; Zhou, H. Study on convolutional neural network and its application in data mining and sales forecasting for E-commerce. *Electron. Commer. Res.* **2020**, *20*, 297–320. [[CrossRef](#)]
23. Sharma, D.K.; Lohana, S.; Arora, S.; Dixit, A.; Tiwari, M.; Tiwari, T. E-Commerce product comparison portal for classification of customer data based on data mining. *Mater. Today Proc.* **2021**, *51 Pt 1*, 166–171. [[CrossRef](#)]
24. Zhong, Q.; Liang, S.; Cui, L.; Chan, H.K.; Qiu, Y. Using online reviews to explore consumer purchasing behaviour in different cultural settings. *Kybernetes* **2019**, *48*, 1242–1263. [[CrossRef](#)]
25. Van Nguyen, T.; Zhou, L.; Chong, A.Y.L.; Li, B.; Pu, X. Predicting customer demand for remanufactured products: A data-mining approach. *Eur. J. Oper. Res.* **2020**, *281*, 543–558. [[CrossRef](#)]
26. Zhang, H.; Rao, H.; Feng, J. Product innovation based on online review data mining: A case study of Huawei phones. *Electron. Commer. Res.* **2018**, *18*, 3–22. [[CrossRef](#)]
27. Morasaei, A.; Ghabussi, A.; Aghlmand, S.; Yazdani, M.; Baharom, S.; Assilzadeh, H. Simulation of steel–concrete composite floor system behavior at elevated temperatures via multi-hybrid metaheuristic framework. *Eng. Comput.* **2021**, 1–16. [[CrossRef](#)]
28. Yazdani, M.; Khalili, S.M.; Babagolzadeh, M.; Jolai, F. A single-machine scheduling problem with multiple unavailability constraints: A mathematical model and an enhanced variable neighborhood search approach. *J. Comput. Des. Eng.* **2017**, *4*, 46–59. [[CrossRef](#)]
29. Yazdani, M.; Ghodsi, R. Invasive weed optimization algorithm for minimizing total weighted earliness and tardiness penalties on a single machine under aging effect. *Int. Robot. Autom. J.* **2017**, *2*, 1–5. [[CrossRef](#)]
30. Azadeh, A.; Seif, J.; Sheikhalishahi, M.; Yazdani, M. An integrated support vector regression–imperialist competitive algorithm for reliability estimation of a shearing machine. *Int. J. Comput. Integr. Manuf.* **2016**, *29*, 16–24. [[CrossRef](#)]
31. Shahmansouri, A.A.; Yazdani, M.; Ghanbari, S.; Bengar, H.A.; Jafari, A.; Ghatte, H.F. Artificial neural network model to predict the compressive strength of eco-friendly geopolymer concrete incorporating silica fume and natural zeolite. *J. Clean. Prod.* **2021**, *279*, 123697. [[CrossRef](#)]
32. Mirmozaffari, M.; Yazdani, M.; Boskabadi, A.; Ahady Dolatsara, H.; Kabirifar, K.; Amiri Golilarz, N. A novel machine learning approach combined with optimization models for eco-efficiency evaluation. *Appl. Sci.* **2020**, *10*, 5210. [[CrossRef](#)]
33. Yazdani, M.; Mojtahedi, M.; Loosemore, M. Enhancing evacuation response to extreme weather disasters using public transportation systems: A novel simheuristic approach. *J. Comput. Des. Eng.* **2020**, *7*, 195–210. [[CrossRef](#)]
34. Yazdani, M.; Jolai, F.; Taleghani, M.; Yazdani, R. A modified imperialist competitive algorithm for a two-agent single-machine scheduling under periodic maintenance consideration. *Int. J. Oper. Res.* **2018**, *32*, 127–155. [[CrossRef](#)]
35. Yazdani, M.; Jolai, F. A genetic algorithm with modified crossover operator for a two-agent scheduling problem. *J. Syst. Manag.* **2013**, *1*, 1–13.
36. Yazdani, M.; Kabirifar, K.; Frimpong, B.E.; Shariati, M.; Mirmozaffari, M.; Boskabadi, A. Improving construction and demolition waste collection service in an urban area using a simheuristic approach: A case study in Sydney, Australia. *J. Clean. Prod.* **2021**, *280*, 124138. [[CrossRef](#)]
37. Yazdani, M.; Aleti, A.; Khalili, S.M.; Jolai, F. Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem. *Comput. Ind. Eng.* **2017**, *107*, 12–24. [[CrossRef](#)]
38. Yazdani, M.; Babagolzadeh, M.; Kazemitash, N.; Saberi, M. Reliability estimation using an integrated support vector regression—Variable neighborhood search model. *J. Ind. Inf. Integr.* **2019**, *15*, 103–110. [[CrossRef](#)]
39. Yazdani, M.; Khalili, S.M.; Jolai, F. A parallel machine scheduling problem with two-agent and tool change activities: An efficient hybrid metaheuristic algorithm. *Int. J. Comput. Integr. Manuf.* **2016**, *29*, 1075–1088. [[CrossRef](#)]
40. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [[CrossRef](#)]
41. Hussain, K.; Salleh, M.N.M.; Cheng, S.; Shi, Y. Metaheuristic research: A comprehensive survey. *Artif. Intell. Rev.* **2019**, *52*, 2191–2233. [[CrossRef](#)]
42. Halim, A.H.; Ismail, I.; Das, S. Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artif. Intell. Rev.* **2021**, *54*, 2323–2409. [[CrossRef](#)]
43. Abdel-Basset, M.; Abdel-Fatah, L.; Sangaiah, A.K. Metaheuristic algorithms: A comprehensive review. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications; Intelligent Data-Centric Systems*; Sangaiah, A.K., Sheng, M., Zhang, Z., Eds.; Academic Press: Cambridge, MA, USA, 2018; pp. 185–231.
44. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [[CrossRef](#)]
45. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [[CrossRef](#)]
46. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper Optimisation Algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [[CrossRef](#)]
47. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]

48. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [[CrossRef](#)]
49. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [[CrossRef](#)]
50. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
51. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [[CrossRef](#)]
52. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
53. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
54. Kleinberg, J.; Papadimitriou, C.; Raghavan, P. A microeconomic view of data mining. *Data Min. Knowl. Discov.* **1998**, *2*, 311–324. [[CrossRef](#)]
55. Steinbach, M.; Karypis, G.; Kumar, V. *Efficient Algorithms for Creating Product Catalogs*; Technical Report #00-057; Minnesota University, Minneapolis Department of Computer Science: Minneapolis, MN, USA, 2000.
56. Xu, D.; Ye, Y.; Zhang, J. Approximating the 2-catalog segmentation problem using semidefinite programming relaxations. *Optim. Methods Softw.* **2003**, *18*, 705–719.
57. Kleinberg, J.; Papadimitriou, C.; Raghavan, P. Segmentation problems. *JACM* **2004**, *51*, 263–280. [[CrossRef](#)]
58. Ester, M.; Ge, R.; Jin, W.; Hu, Z. A microeconomic data mining problem: Customer-oriented catalog segmentation. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 557–562.
59. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago de Chile, Chile, 12–15 September 1994; pp. 487–499.
60. Amiri, A. Customer-oriented catalog segmentation: Effective solution approaches. *Decis. Support Syst.* **2006**, *42*, 1860–1871. [[CrossRef](#)]
61. Mahdavi, I.; Movahednejad, M.; Adbesh, F. Designing customer-oriented catalogs in e-CRM using an effective self-adaptive genetic algorithm. *Expert Syst. Appl.* **2011**, *38*, 631–639. [[CrossRef](#)]
62. Namvar, M.; Khakabimamaghani, S.; Gholamian, M.R. An approach to optimised customer segmentation and profiling using RFM, LTV, and demographic features. *Int. J. Electron. Cust. Relatsh. Manag.* **2011**, *5*, 220–235. [[CrossRef](#)]
63. Yousefpoor, N.; Olfat, L. An integrated mathematical model to optimise expected profit in e-markets using web catalogues. *Int. J. Ind. Syst. Eng.* **2012**, *12*, 100–118. [[CrossRef](#)]
64. Hsu, H.; Chang, R.; Ho, J. Query-Based-Learning Genetic Algorithm to Construct Mobile-Oriented Catalogs in M-Commerce. *IEEE Access* **2017**, *5*, 7294–7304. [[CrossRef](#)]
65. Makinde, A.S.; Vincent, O.R.; Akinwale, A.T.; Oguntuase, A.; Acheme, I.D. An Improved Customer Relationship Management Model for Business-to-Business E-commerce Using Genetic-Based Data Mining Process. In Proceedings of the 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Lagos, Nigeria, 18–21 March 2020; pp. 1–7.
66. Sohani, A.; Naderi, S.; Torabi, F. Comprehensive comparative evaluation of different possible optimization scenarios for a polymer electrolyte membrane fuel cell. *Energy Convers. Manag.* **2019**, *191*, 247–260. [[CrossRef](#)]
67. Zhang, C.; Fathollahi-Fard, A.M.; Li, J.; Tian, G.; Zhang, T. Disassembly Sequence Planning for Intelligent Manufacturing Using Social Engineering Optimizer. *Symmetry* **2021**, *13*, 663. [[CrossRef](#)]
68. Pasha, J.; Dulebenets, M.A.; Fathollahi-Fard, A.M.; Tian, G.; Lau, Y.-y.; Singh, P.; Liang, B. An integrated optimization method for tactical-level planning in liner shipping with heterogeneous ship fleet and environmental considerations. *Adv. Eng. Inform.* **2021**, *48*, 101299. [[CrossRef](#)]
69. Khalili, S.M.; Babagolzadeh, M.; Yazdani, M.; Saberi, M.; Chang, E. A bi-objective model for relief supply location in post-disaster management. In Proceedings of the 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), Ostrava, Czech Republic, 7–9 September 2016; pp. 428–434.
70. Aghapour, A.H.; Yazdani, M.; Jolai, F.; Mojtahedi, M. Capacity planning and reconfiguration for disaster-resilient health infrastructure. *J. Build. Eng.* **2019**, *26*, 100853. [[CrossRef](#)]
71. Yazdani, M.; Jolai, F. Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm. *J. Comput. Des. Eng.* **2016**, *3*, 24–36. [[CrossRef](#)]
72. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [[CrossRef](#)]
73. Theophilus, O.; Dulebenets, M.A.; Pasha, J.; Lau, Y.-Y.; Fathollahi-Fard, A.M.; Mazaheri, A. Truck scheduling optimization at a cold-chain cross-docking terminal with product perishability considerations. *Comput. Ind. Eng.* **2021**, *156*, 107240. [[CrossRef](#)]
74. Yazdani, M.; Kabirifar, K.; Fathollahi-Fard, A.M.; Mojtahedi, M. Production scheduling of off-site prefabricated construction components considering sequence dependent due dates. *Environ. Sci. Pollut. Res.* **2021**, 1–17. [[CrossRef](#)]
75. Salari, S.A.-S.; Mahmoudi, H.; Aghsami, A.; Jolai, F.; Jolai, S.; Yazdani, M. Off-Site Construction Three-Echelon Supply Chain Management with Stochastic Constraints: A Modelling Approach. *Buildings* **2022**, *12*, 119. [[CrossRef](#)]
76. Safaeian, M.; Fathollahi-Fard, A.M.; Kabirifar, K.; Yazdani, M.; Shapouri, M. Selecting Appropriate Risk Response Strategies Considering Utility Function and Budget Constraints: A Case Study of a Construction Company in Iran. *Buildings* **2022**, *12*, 98. [[CrossRef](#)]

77. Mohammadnazari, Z.; Mousapour Mamoudan, M.; Alipour-Vaezi, M.; Aghsami, A.; Jolai, F.; Yazdani, M. Prioritizing Post-Disaster Reconstruction Projects Using an Integrated Multi-Criteria Decision-Making Approach: A Case Study. *Buildings* **2022**, *12*, 136. [[CrossRef](#)]
78. Lazar, A. Heuristic knowledge discovery for archaeological data using genetic algorithms and rough sets. In *Heuristic and Optimization for Knowledge Discovery*; IGI Global: Hershey, PA, USA, 2002; pp. 263–278.
79. Jana, N.D.; Das, S.; Sil, J. *A Metaheuristic Approach to Protein Structure Prediction*; Springer: Cham, Switzerland, 2018.
80. Fathollahi-Fard, A.M.; Ahmadi, A.; Karimi, B. Multi-Objective Optimization of Home Healthcare with Working-Time Balancing and Care Continuity. *Sustainability* **2021**, *13*, 12431. [[CrossRef](#)]
81. Fathollahi-Fard, A.M.; Ahmadi, A.; Karimi, B. Sustainable and Robust Home Healthcare Logistics: A Response to the COVID-19 Pandemic. *Symmetry* **2022**, *14*, 193. [[CrossRef](#)]
82. Goodarzian, F.; Abraham, A.; Fathollahi-Fard, A.M. A biobjective home health care logistics considering the working time and route balancing: A self-adaptive social engineering optimizer. *J. Comput. Des. Eng.* **2020**, *8*, 452–474. [[CrossRef](#)]
83. Gholizadeh, H.; Fazlollahtabar, H.; Fathollahi-Fard, A.M.; Dulebenets, M.A. Preventive maintenance for the flexible flowshop scheduling under uncertainty: A waste-to-energy system. *Environ. Sci. Pollut. Res.* **2021**. [[CrossRef](#)]
84. Sengupta, R.N.; Gupta, A.; Dutta, J. *Decision Sciences: Theory and Practice*; CRC Press: Boca Raton, FL, USA, 2016.
85. Talbi, E.-G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
86. Piotrowski, A.P.; Napiorkowski, M.J.; Napiorkowski, J.J.; Rowinski, P.M. Swarm Intelligence and Evolutionary Algorithms: Performance versus speed. *Inf. Sci.* **2017**, *384*, 34–85. [[CrossRef](#)]
87. Gogna, A.; Tayal, A. Metaheuristics: Review and application. *J. Exp. Theor. Artif. Intell.* **2013**, *25*, 503–526. [[CrossRef](#)]
88. Fathollahi-Fard, A.M.; Dulebenets, M.A.; Hajiaghaei-Keshteli, M.; Tavakkoli-Moghaddam, R.; Safaeian, M.; Mirzahosseini, H. Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design problem in the tire industry under uncertainty. *Adv. Eng. Inform.* **2021**, *50*, 101418. [[CrossRef](#)]
89. Dinh, P.-H. A novel approach based on Grasshopper optimization algorithm for medical image fusion. *Expert Syst. Appl.* **2021**, *171*, 114576. [[CrossRef](#)]
90. Lv, Z.; Peng, R. A novel periodic learning ontology matching model based on interactive grasshopper optimization algorithm. *Knowl.-Based Syst.* **2021**, *228*, 107239. [[CrossRef](#)]
91. Motlagh, S.Z.T.; Akbari Foroud, A. Power quality disturbances recognition using adaptive chirp mode pursuit and grasshopper optimized support vector machines. *Measurement* **2021**, *168*, 108461. [[CrossRef](#)]
92. Fathollahi-Fard, A.M.; Hajiaghaei-Keshteli, M.; Tavakkoli-Moghaddam, R.; Smith, N.R. Bi-level programming for home health care supply chain considering outsourcing. *J. Ind. Inf. Integr.* **2022**, *25*, 100246. [[CrossRef](#)]