ISSN 2007-9737

Multi-Objective Evolutionary Algorithm based on Decomposition with Adaptive Adjustment of Control Parameters to Solve the Bi-Objective Internet Shopping Optimization Problem (MOEA/D-AACPBIShOP)

Miguel A. García-Morales[1], José A. Brambila-Hernández[1],
Héctor J. Fraire-Huacuja[1,*], Juan Frausto-Solis[1],
Laura Cruz-Reyes[1], Claudia G. Gómez-Santillan[1],
Juan M. Carpio-Valadez[2]

[1] National Technological Institute of Mexico,
Technological Institute of Madero City,
Mexico

[2] National Technological Institute of Mexico,
Technological Institute of Leon,
Mexico

{soporteclusterlanti, hector.fraire2014, juan.frausto, cruzreyeslaura}@gmail.com,
alfredo.brambila@outlook.com, claudia.gs@cdmadero.tecnm.mx,
juanmartin.carpio@leon.tecnm.mx

Abstract. The main contribution of this paper is the implementation of a multi-objective evolutionary algorithm based on decomposition with adaptive adjustment of control parameters applied to the bi-objective problem of Internet shopping (MOEA/D-AACPBIShOP). For this variant of the IShOP, the minimization of the cost and shipping time of the shopping list is considered. The proposed MOEA/D-AACPBIShOP algorithm produces an approximate Pareto set on a total of nine of instances with real-world data classified as small, medium, and large. The instances are obtained using the Web Scraping technique, extracting some information attributes of technological products from the Amazon site. This optimization problem is a very little studied variant of the Internet Shopping Problem (IShOP). The proposed algorithm is compared with two multi-objective algorithms: A Non-dominated Sorting Genetic Algorithm II (NSGA-II) and the basic MOEA/D version. The results demonstrate that the three algorithms studied have a similar statistical performance with respect to the quality of the solutions they provide. To make a comparison, these algorithms are evaluated using three metrics: Hypervolume, Generalized Dispersion, and Inverted Generational Distance. On the other hand, the Wilcoxon and Friedman non-parametric tests validate the obtained results with a 5% significance level.

Keywords. Multi-objective, approximate Pareto front, evolutionary algorithm, web scraping, bi-objective.

1 Introduction

The Internet allows efficient communication throughout the world [10]. The Internet has revolutionized the way business is carried out due to the incorporation of commercial marketing, sales, and customer service tools [10].

Due to the great the importance of the Internet in organizations, E-commerce is one of the main contributors of large companies [1]. On the other hand, the Internet allows communication from multiple digital devices such as sensors, cameras, smart cities, among others [2, 3, 4]. Nowadays, this scenario is known as “The Internet Shopping Problem”.

It is a classic scenario of electronic commerce due to the multiple benefits that users obtain by buying or acquiring goods or services through the Internet [5]. Online shopping makes it easier for people to access a wide variety of products and services offered by companies without having restrictions on time, place, or space [1].

In one of the most relevant works in the state-of-the-art field, the authors propose an innovative solution for the basic case of the Internet shopping problem with shipping costs.

Table 1. Notation table [10]

Variable/Parameter	Description
M	Group of stores
N	Group of products
I	Array solution
m	Number of stores, $\|M\|$
n	Number of products, $\|N\|$
j	Store indicator
i	Product indicator
N_i	Container of products available in a store j
f_j	Shipping cost of all products in the store j
p_{ij}	Cost of product i in store j
d_{ij}	Delivery time of a product i in store j
x_{ij}	Binary variable that indicates wheter producto i is purchased in store j
y_j	Binary variable indicating wheter to add the sipping cost of store j

This method consists of a memetic algorithm (MAIShOP) that incorporates standard instances, solution generation through the first-best heuristic, and a local search based on a heuristic that selects the lowest cost of each product in all stores [6].

Morales et al. [7] review the developed models, the implemented solution methods, and the instances used to analyze the performance of the algorithms described in the state-of-the-art.

Finally, it can be identified that one of the variants little investigated is the one that involves more than one optimization objective, in which the total cost of the purchase and the delivery time of products are considered.

Some Internet purchases require optimizing the total purchase cost, including the shipping cost and delivery time of different online stores [1]. Typically, users want to find the store with the lowest total cost and the shortest delivery time [1].

These decisions allow us to minimize the effort and maximize the benefit of the shopping list [10]. Chung [8] proposes a new Internet shopping optimization model that includes two objectives (total cost and delivery time) in which he incorporates for the first time a multi-objective optimization model.

Chaerani et al. [9] establishes the similarity between the model developed by Chung and the maximum flow problem with circular demand (MFP-CD) because it matches the multiples sources with respect to the multiple stores.

Chung's bi-objective model incorporates the decision variable on delivery time. Chaerani et al. [9] modifies this decision variable into an adjustable robust counterpart (ARC) method. Chaerani et al. [1] propose the Benders decomposition method to solve the Adjustable Robust Count Party Problem adapted to "the Internet Shopping Problem (ARC-ISOP)".

García-Morales et al. [10] propose a "MOEA/D algorithm to solve the bi-objective Internet shopping optimization problem (MOEA/D-BIShOP)"; this algorithm presents a basic MOEA/D version and has a clear superiority in two of the three metrics that were evaluated concerning the results of the state-of-the-art.

This research work proposes the implementation of a multi-objective evolutionary algorithm based on decomposition with adaptive adjustment of control parameters as a solution method to "the Bi-objective problem of Internet Shopping (MOEA/D-AACPBIShOP)".

In the computational feasibility tests, nine instances generated using the Web Scraping technique with data from technological products extracted from Amazon were used [10].

1.1 Definition of the Problem

This model is first proposed by Chung [8] to solve "the bi-objective Internet shopping optimization problem". "In this problem, a customer wants to buy a set of n products N online, which can be purchased in a set of m available stores M.

Algorithm 1. NSGA-II/BIShOP Algorithm

Input: cs: chromosome size, nt: number of targets, mni: maximum number of iterations, ps: population size, pc: crossing percentage, nci: number of crossed individuals, pm: mutation percentage, nmi: number of mutated individuals, m: number of stores, n: number of products, p_{ij}: price of each product, f_j: shipping cost, d_{ij}: delivery time.

Output: Pop

```
Initialize parameters: chromosome size cs, number of targets nt, maximum number of iterations mni, population size ps, crossover percentage pc, number of crossed individuals nci, mutation percentage pm, number of mutated individuals nmi.
Pop ← initial_population (ps)
    F ← non_dominated_sort (Pop)
    Pop ← crowding_distance (Pop, F)
    Pop ← sort_by_crowding_distance_and_Front (Pop, F)
    PopC ← selectionBinaryTournament (Pop)
    while non stop criterion do
        PopC ← crossover (PopC)
        PopM ← mutation (PopC)
        Pop ← merge_list (Pop, PopC, PopM)
        F ← non_dominated_sort (Pop)
        Pop ← crowding_distance (Pop, F)
        Pop ← sort_by_crowding_distance_and_Front (Pop, F)
        Pop ← truncate_list (Pop, ps)
        F ← non_dominated_sort (Pop)
        Pop ← crowding_distance (Pop, F)
        Pop ← sort_by_crowding_distance_and_Front (Pop, F)
    end while
return Pop
```

Now, the set $\mathrm{N_i}$ contains the products available in store i, each product $j \in N_i$ has a cost of p_{ij}, a shipping cost f_j, and a delivery time d_{ij}. The shipping cost is charged if one or more products are purchased in the store i.

The Bi-objective Internet Shopping Optimization Problem (BIShOP) consists of minimizing the total cost of purchasing all products N, considering the cost-plus shipping costs, and minimizing the delivery time" [10]. Table 1 describes the parameters and variables used in the model.

The model presents the optimization of two objectives: one is the purchase cost, and the other is the delivery time limitation. The first objective seeks to minimize the purchase cost; the second objective seeks to minimize the delivery time of the products (see Equation 1):

$$\text{Min} \sum_i \sum_j p_{ij} x_{ij} + \sum_j f_j y_j \,, \qquad (1)$$

$$\text{Min Max}_{i,j(d_{ij} x_{ij})},$$

s.t.

$$\sum_j x_{ij} = 1, \forall i = 1, \dots, n,$$

$$\sum_i x_{ij} \le n y_j, j = 1, \dots, m,$$

$$x_{ij} = 0/1, y_j = 0/1,$$

where m represents the number of stores, n the number of products, $\sum_j x_{ij} = 1$ is a limitation that indicates that the items to be purchased must be chosen only from available stores. $\sum_i x_{ij} \le n y_j$ is a constraint that implies that a standard shipping cost will be applied every time a purchase is made in the store, regardless of the products selected, and $x_{ij} = 0/1, y_j = 0/1$ indicates that decision variables can only take binary values.

2 General Structure of Multi Objective Algorithms Applied to BIShOP

This section provides a detailed explanation of the essential components that form the multi-objective optimization algorithms utilized in "BIShOP". To represent each solution in the population, these algorithms employ a vector representation, which is an I vector of N length. This I vector includes all the stores from where the products can be purchased. Equation 1 shows in detail how the calculation of the objective functions is carried out.

2.1 Crossover Operator

This operator randomly selects two solutions called parent_1 and parent_2 [11]. The solution child_1 is generated by taking the initial half of parent_1 and joining it with the second half of parent_2. Later, to form child_2, the initial half of parent_2 is joined with the second half of parent_1 [12]. Subsequently, a random number is generated; if this generated value is less than 0.5, the crossover operator selects child_1; otherwise, it takes child_2 to advance to the mutation process.

The crossover operator uses $\lfloor N/2 \rfloor$ or $\lceil N/2 \rceil$ as the crossover point. In the case of the MOEA/D algorithm, the crossover operator selects only one of the generated children and randomly decides which one will continue with the mutation process [10]”. The NSGA-II algorithm allows both offspring generated during the crossover process to advance to the mutation process.

2.2 Mutation Operator

The mutation process of the MOEA/D algorithm takes the candidate solution selected by the crossover operator. It immediately positions itself on the first element of the solution and generates a random number; if this random value is less μ, the current element of the solution is replaced by a random value in the online stores range $[1, m]$ [10].

This process continues until all elements of the current solution have been examined”. The mutation process of the NSGA-II algorithm goes through all the elements of the vector and searches in which store that product has the lowest cost. This search ends when all stores in all products have been reviewed.

2.3 The Non-Dominated Sorting Genetic Algorithm II to Solve the IShOP Bi-Objective Problem (NSGA-II/BIShOP)

NSGA-II is a multi-objective optimization algorithm proposed as an improvement of NSGA [13], it uses the structure of genetic algorithms and is based on these principles: the best individuals never disappear from the population and during the selection if two non-dominated solutions are found, the most diverse one is preferred.

Algorithm 1 describes the general structure of the NSGA-II algorithm applied to the BIShOP problem. The algorithm in step 1 starts by defining the parameters such as chromosome size cs, number of targets nt, maximum number of iterations mni, population size ps, crossover percentage pc, number of crossed individuals nci, mutation percentage pm and number of mutated individuals nmi.

In step 2, a Pop population is created randomly. From steps 3 to 5, the population is ordered according to the levels of non-dominance (ordering of the Pareto fronts: F_1, F_2,…). Each solution is assigned a fitness function according to its level of non-dominance (1 is the best level) and it is understood that this function must be decreased during the process. In step 6, binary tournament selection is applied, and a new population called PopC is obtained.

The population obtained in the previous step is used in the crossover operator and is updated in step 8. In step 9, the mutation operator is applied and a new population of PopM descendants is obtained. In step 10, the three populations (Pop, PopC and PopM) are joined. From steps 11 to 13, a ranking is assigned to each individual in the fronts and the crowding distance is obtained, subsequently they are ordered, first by fronts from lowest to highest and then by crowding distance from highest to lowest. In step 14 the list of elements is truncated to leave only the best individuals, and which fits the initial ps. From steps 15 to 17, the previous process is applied again, only to the population that was obtained in the previous steps. Finally, in step 19 the NSGA-II algorithm returns the front with the best individuals obtained in the entire process.

Algorithm 1 NSGA-II/BIShOP Algorithm

Input: cs: chromosome size, nt: number of targets, mni: maximum number of iterations, ps: population size, pc: crossing percentage, nci: number of crossed individuals, pm: mutation percentage, nmi: number of mutated individuals, m: number of stores, n: number of products, p_{ij}: price of each product, f_j: shipping cost, d_{ij}: delivery time.

Output: Pop

1: Initialize parameters: chromosome size $\boldsymbol{cs}$, number of targets $\boldsymbol{nt}$, maximum number of iterations $\boldsymbol{mni}$, population size $\boldsymbol{ps}$, crossover percentage $\boldsymbol{pc}$, number of crossed individuals $\boldsymbol{nci}$, mutation percentage $\boldsymbol{pm}$, number of mutated individuals $\boldsymbol{nmi}$.

2: $Pop \leftarrow initial\_population\ (ps)$

3: $F \leftarrow non\_dominated\_sort\ (Pop)$

4: $Pop \leftarrow crowding\_distance\ (Pop, F)$

5: $Pop \leftarrow sort\_by\_crowding\_distance\_and\_Front\ (Pop, F)$

6: $PopC \leftarrow selectionBinaryTournament\ (Pop)$

7: $\boldsymbol{while}$ $non\ stop\ criterion$ $\boldsymbol{do}$

8: $PopC \leftarrow crossover\ (PopC)$

9: $PopM \leftarrow mutation\ (PopC)$

10: $Pop \leftarrow merge\_list\ (Pop, PopC, PopM)$

ISSN 2007-9737

$F \leftarrow non\_dominated\_sort\ (Pop)$
$Pop \leftarrow crowding\_distance\ (Pop, F)$
$Pop \leftarrow sort\_by\_crowding\_distance\_and\_Front\ (Pop, F)$
$Pop \leftarrow truncate\_list\ (Pop, ps)$
$F \leftarrow non\_dominated\_sort\ (Pop)$
$Pop \leftarrow crowding\_distance\ (Pop, F)$
$Pop \leftarrow sort\_by\_crowding\_distance\_and\_Front\ (Pop, F)$
$\boldsymbol{end\ while}$
$\boldsymbol{return}\ Pop$

2.4 The Multi-Objective Evolutionary Algorithm based on Decomposition with Adaptive Adjustment of Control Parameters to Solve the IShOP Bi-Objective Problem (MOEA/D AACPBIShOP)

The multi-objective evolutionary algorithm based on decomposition (MOEA/D) was developed by Zhang and Li [14, 15, 16] and serves as a reliable and robust alternative for working with MOPs. Initially it makes a distribution of a set of weight vectors (λ) within the objective functional space.

Subsequently it creates a matrix of T closets vectors considering the Euclidean distance between the vectors, thus generating neighborhoods [17]". The basic version of the MOEA/D algorithm uses the Tchebycheff decomposition shown in Eq. 6:

$$\min g^{te}\left(x \middle| \lambda^j, z^*\right) = \max_{1 \le i \le m} \frac{1}{\lambda_i^j} |f_i(x) - z_i^*|, \quad (2)$$

The MOEA/D-AACPBIShOP algorithm is represented in Algorithm 2. In steps 1 to 4, λ reference vectors are established, neighborhoods are created using the T nearest neighbor vectors as criteria, and the ideal Z point is calculated.

The main loop runs through all individuals within the population. In step 7, two parents are chosen. These are taken from the neighborhoods created in $B(i)$. $B(i)$ is traversed, and two parents are chosen randomly; then, the crossover and mutation operators are applied to generate a single child. In the final part of the algorithm, the Z value is updated again.

The aggregation values of the two are calculated using λ reference vectors; likewise, the aggregation value of the child y^i is replaced with a simple criterion: if the child y^i has an aggregation value less than one of the parents, it is replaced; otherwise, the parent remains, and the population is not modified".

Algorithm 2 MOEA/D-AACPBIShOP Algorithm

Input: MOP – Bi-objective IShOP Problem, Pop – Population, $nPop$ – Population size, $fileSize$ – File size, Stopping criterion, N – the number of subproblems considered in MOEA/D-AACPBIShOP, A uniform distribution of N weight vectors: $\lambda^1, \dots, \lambda^N$, T – the number of weight vectors in the neighborhoods of each weight vector

Output: EP

Functions: $FRRMAB()$: obtains an index of an action to perform, $executeAction(actionindex)$: executes an action according to an index, $SlidingWindow\ (actionindex, improvement[i])$: Sliding window that stores the index of action to be performed and the improvement in cost, $UpgradeRewards(SlidingWindow)$: Updates the sliding window rewards.

1: $EP = \emptyset$
2: Compute the Euclidean distances between any two weight vectors and then compute the weight vectors T closets to each weight vector.
3: $\boldsymbol{for}\ i \leftarrow 1\ \boldsymbol{to}\ N\ \boldsymbol{do}$
4: $B(i) = \{i_1, \dots, i_T\}$ where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are T nearest weight vectors λ^i
5: $\boldsymbol{end\ for}$
6: Generate initial population $x^1, \dots, x^N$ randomly.
7: $FV^i = F(x^i)$
8: Initialize $z = (z_1, \dots, z_m)^N$ for the bi-objective IShOP
9: $\boldsymbol{while}$ stopping criterion not met $\boldsymbol{do}$
10: $indexaction = FRRMAB()$
11: $executeAction(indexaction)$
12: $\boldsymbol{for}\ i \leftarrow 1\ \boldsymbol{to}\ N\ \boldsymbol{do}$
13: Randomly select two indices *k*, *l* from *B*(*i*), and generate a new solution y from x^k and x^l using genetic operators

Apply a problem-specific repair/improvement heuristic on *y* to produce *y′*
    $\boldsymbol{for}\ j \leftarrow 1\ \boldsymbol{to}\ m\ \boldsymbol{do}$
        $\boldsymbol{if}\ z_j < f_j(y')\ \boldsymbol{then}$
            $z_j = f_j(y')$
        $\boldsymbol{end\ if}$
    $\boldsymbol{end\ for}$
    $\boldsymbol{foreach}$ index $j \in B(i)$ $\boldsymbol{do}$
        $\boldsymbol{if}\ g^{te}(y'|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)\ \boldsymbol{then}$
            $x^j = y'$
            $FV^j = F(y')$
        $\boldsymbol{end\ if}$
    $\boldsymbol{end\ foreach}$
    Add to $SlidingWindow(actionindex, improvement[i])$
  $Upgrade\_Rewards(SlidingWindow)$
  $\boldsymbol{end\ for}$
  remove from EP all solutions dominated by *F*(*y'*)
  insert *F*(*y'*) in EP if there are no solutions in EP that dominate *F*(*y'*)
$\boldsymbol{end\ while}$

In this research work, the modified version of the adaptive operator selection method is used to achieve adaptive adjustment of control parameters. Using the Fitness-Rate-Rank-Based Multi-armed Bandit Adaptive (FRRMAB) method [18]. The FRRMAB method avoids this problem using fitness improvement rates (FIR). The formula for calculating these rates is shown in Equation 3:

$$FIR_{i,t} = \frac{pf_{i,t} - cf_{i,t}}{pf_{i,t}}, \tag{3}$$

where $pf_{i,t}$ is the fitness value of the parent, and $cf_{i,t}$ is the fitness value of the children. The reward (Reward_i) of the actions is calculated by adding the FIR values of each action within the sliding window, they are ordered in descending order and classified, using the rank (Rank_i) for each action i [18]. In the end, only the best stocks are selected, considering the decay factor $D \in [0,1]$. Rewards Reward_i are transformed using Equation 4:

$$\text{Decay}_i = D^{\text{Rank}_i} \times \text{Reward}_i. \tag{4}$$

To assign credits to action i, use Equation 5:

$$FRR_{i,t} = \frac{\text{Decay}_i}{\sum_{j=1}^{K} \text{Decay}_j}. \tag{5}$$

The lower the D decay value, the more likely it is to influence the stock's upside. The credit allocation process is represented in Algorithm 3 [17].

Algorithm 3 Assignment of credits

Initialize each reward $Reward_i = 0$
Initialize $n_i = 0$;
$\boldsymbol{for}\ i \leftarrow 1\ \boldsymbol{to}$ SlidingWindow.length $\boldsymbol{do}$
    $action =$ SlidingWindow.GetIndexaction(i)
    $FIR =$ SlidingWindow.GetFIR(i)
    $Reward_{action} = Reward_{action} + FIR$
    $n_{action}++$
$\boldsymbol{endfor}$
Rank $Reward_i$ in descending order and set $Rank_i$ t be the rank value of action i
$\boldsymbol{for}\ action \leftarrow \boldsymbol{to}\ K\ \boldsymbol{do}$
    $Decay_{action} = D^{Rank_{action}} \times Reward_{action}$
$\boldsymbol{endfor}$
    $DecaySum = \sum_{action=1}^{K} Decay_{action}$
$\boldsymbol{for}\ action \leftarrow \boldsymbol{to}\ K\ \boldsymbol{do}$
    $FRR_{action} = Decay_{action}/DecaySum$
$\boldsymbol{endfor}$

Bandit-based action selection chooses a stock considering the credits assigned to it and using the FRR values as a quality indicator [18], this process is shown in Algorithm 4.

Algorithm 4 Bandit-based action selection

if there are actions that have not been selected **then**
    $\text{action}_t =$ randomly select a security from the action pool
else

$$\text{action}_t = \text{argmax}_{i=1,\ldots,K}\left(FRR_{i,t} + C \times \sqrt{\frac{2 \times \ln\left(\sum_{j=1}^{K}\{n_{j,t}\}\right)}{n_{i,t}}}\right)$$

end if

Algorithm 5 contains the various actions that are executed when the variable $actionindex$ is evaluated and said action determines the increase

ISSN 2007-9737

or decrease in the value of the parameters that are adjusted adaptively.

Algorithm 5 executeAction

Input: $actionindex$: value of the stock selected by the FRRMAB method.

```
Switch(actionindex)
Case 1:
    pc = pc + 0.0001;
    pm = pm + 0.0001;
    sigma = sigma + 0.0001;
break;
Case 2:
    pc = pc − 0.0001;
    pm = pm − 0.0001;
    sigma = sigma − 0.0001;
break;
Case 3:
    pc = pc + 0.0001;
break;
Case 4:
    pm = pm + 0.0001;
break;
Case 5:
    sigma = sigma + 0.0001;
break;
Case 6:
    pc = pc − 0.0001;
break;
Case 7:
    pm = pm − 0.0001;
break;
Case 8:
    sigma = sigma − 0.0001;
break;
Case 9:
    pm = pm + 0.0001;
    sigma = sigma + 0.0001;
break;
Case 10:
    pc = pc + 0.0001;
    pm = pm + 0.0001;
break;
Case 11:
    sigma = pc + 0.0001;
    sigma = sigma + 0.0001;
break;
Case 12:
    pm = pm − 0.0001;
    sigma = sigma − 0.0001;
break;
Case 13:
    pc = pc − 0.0001;
    pm = pm − 0.0001;
break;
Case 14:
    pc = pc − 0.0001;
    sigma = sigma − 0.0001;
break;
```

3 Computational Experiments

The names of instances determine their size, m is the number of stores, and n is the number of products. For the experimental test, three sets of real-world instances of different sizes were used, and each subset contains 30 instances, as can be seen in Table 2 [10]".

The designs are obtained from Web Scraping of multiple technological products (USB flash, Modem, RAM) that were carried out on Amazon's e-commerce website. In this process, approximately 8002 records containing product names, prices, suppliers, delivery time, and shipping costs were obtained [10].

Fig. 1 shows the process of building the instances from real-world data described below: collect product and store information from the Amazon.com page.

Build an application in the Python language that allows us to explore within the search engine and obtain information using the Web Scraping technique, using various keywords such as laptop, headphones, and speakers, among others.

With a depth of 10 pages for each, the Beautiful Soup Python library is used to process the information [10]. A first version of the instances has been generated, and its construction is carried out by taking the products obtained with a defined price range and the stores are obtained.

Shipping times are defined arbitrarily (randomly) with values between 1 and 5 days. For the shipping cost, four arbitrary values are used, which are assigned randomly. These values are 88, 99, 120, and 140. The types of instances generated are shown in Table 2 [10]".

ISSN 2007-9737

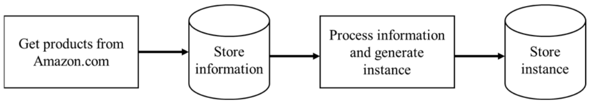

Fig. 1. General instance generation process [10]

Table 2. Definition of instances

Small	Medium	Large
3n20m	5n240m	50n400m
4n20m	5n400m	100n240m
5n20m	50n240m	100n400m

Table 3. Parameter configuration of multi-objective algorithms

Variable	MOEA/D-BIShOP	MOEA/D-AACPBIShOP	NSGA-II/BIShOP
pop	100	100	100
pc	0.5	*0.5	0.5
pm	0.01	*0.01	0.01
maxIter	100	100	100
μ	--	0.02	0.02

*Initial values before adaptive adjustment

3.1 Configuration of the Parameters

The configuration parameters of the proposed MOEA/D-AACPBIShOP algorithm is shown below pop $= 100$, pc $= 0.5$, pm $= 0.01$, maxIter $= 100$, and $\mu = 0.02$.

The above configuration was determined based on related works found in the state-of-the-art. Modifying the values of the parameters can affect the behavior of the algorithm and, therefore, the quality of the solutions.

The size of the population is important because it affects the diversity and convergence of the algorithm. A small population can lead to loss of performance, diversity, and early convergence.

An inadequate number of generations can cause the algorithm to converge prematurely or have excessive resource consumption, and incorrect use of the crossover and mutation operators can lead to deadlocks or inefficient explorations of the solution space and the size of the neighborhood because it determines the number of neighboring solutions to explore contributes to the quality of the generated solutions.

In the computational experiments, the 30 non-dominated fronts were obtained from each of the three sets of instances for each subset; subsequently, non-parametric tests were applied, and the $p-value$ was obtained to determine if there were significant differences in favor of the implemented algorithm".

Table 3 shows the parameters used for each algorithm used. The algorithms were implemented in the Java language.

3.2 Results

Tables 4, 6, and 8 organize the experimental results by metric. Friedman and Wilcoxon non-parametric tests were used with a significance level of 5%.

The first column of each table corresponds to the evaluated instance name. The second column corresponds to the reference algorithm results (MOEA/D-BIShOP). The third column contains the results of the proposed MOEA/D-AACPBIShOP and the fourth the results of the NSGA-II algorithm.

In the table, the symbol ▲ represents the statistical significance in favor of the reference algorithm, the symbol ▼ indicates that there is significant statistical difference in favor of the comparison algorithm (current column), and the symbol == means that the algorithms being compared have the same statistical performance.

The cells marked in dark gray represent the winning algorithm in a given problem and the front, and second places are marked in light gray.

3.1.1 Hypervolume

"The hypervolume (HV) calculates the volume of the objective space weakly dominated by an approximation set [17]. The first column in Table 4 represents the reference algorithm". As can be seen, in the hypervolume metric, the NSGA-II/BIShOP algorithm is better in five of the nine problems compared to the reference algorithm and compared to the MOEA/D-AACPBIShOP Algorithm it has a similar performance.

3.1.1.1 Friedman Test

"The p-value calculated with the Friedman test is 0.12110333239233029, so with a level of statistical

Table 4. Results HV (median and IQR values)

Problem	MOEA/D-BIShOP	MOEA/D-AACPBIShOP	NSGA-II/BIShOP
3n20m	0.00e+00 3.33e-01	3.33e-01 3.33e-01 ==	1.00e+00 3.33e-16▼
4n20m	0.00e+00 2.50e-01	0.00e+00 2.50e-01 ==	1.00e+00 2.50e-01▼
5n20m	0.00e+00 3.24e-01	0.00e+00 3.33e-01 ==	1.00e+00 3.33e-16▼
5n240m	0.00e+00 3.33e-01	0.00e+00 3.33e-01 ==	1.00e+00 3.33e-16▼
5n400m	0.00e+00 0.00e+00	0.00e+00 3.33e-01 ==	1.00e+00 3.33e-16▼
50n240m	0.00e+00 0.00e+00	0.00e+00 0.00e+00 ==	0.00e+00 0.00e+00 ==
50n400m	0.00e+00 0.00e+00	0.00e+00 0.00e+00 ==	0.00e+00 0.00e+00 ==
100n240m	1.00e+00 0.00e+00	1.00e+00 0.00e+00 ==	0.00e+00 0.00e+00 ==
100n400m	1.00e+00 0.00e+00	0.00e+00 0.00e+00 ==	1.00e+00 0.00e+00 ==

Table 5. Average ranks for HV

Algorithm	AVG Rank
NSGA-II/BIShOP	1.44
MOEA/D-AACPBIShOP	2.22
MOEA/D-BIShOP	2.33

Table 6. Results GS (Median and IQR values)

Problem	MOEA/D-BIShOP	MOEA/D-AACPBIShOP	NSGA-II/BIShOP
3n20m	4.86e-01 1.02e-01	4.83e-01 6.74e-02 ▼	4.66e-01 5.52e-02 ▼
4n20m	4.15e-01 6.55e-02	4.15e-01 6.64e-02 ==	4.05e-01 5.43e-02 ▼
5n20m	4.86e-01 1.28e-01	4.98e-01 9.84e-02 ▲	4.66e-01 5.52e-02 ▼
5n240m	4.94e-01 9.88e-02	4.89e-01 7.74e-02 ▼	4.76e-01 8.29e-02 ▼
5n400m	4.15e-01 8.92e-02	4.11e-01 7.67e-02 ▼	4.07e-01 6.41e-02 ▼
50n240m	0.00e+00 0.00e+00	0.00e+00 0.00e+00 ==	0.00e+00 0.00e+00 ==
50n400m	0.00e+00 0.00e+00	0.00e+00 0.00e+00 ==	0.00e+00 0.00e+00 ==
100n240m	0.00e+00 0.00e+00	0.00e+00 0.00e+00 ==	0.00e+00 0.00e+00 ==
100n400m	0.00e+00 0.00e+00	0.00e+00 0.00e+00 ==	0.00e+00 0.00e+00 ==

significance of 5%, it is significant. Table 5 below shows the average ranks per algorithm obtained with the Friedman test”. The Friedman test suggests that no algorithm differs significantly.

The above shows that the algorithm obtains better approximate Pareto fronts for all the evaluated instances.

3.1.2 Generalized Spread

“Generalized Spread (GS) evaluates the degree of dispersion and uniformity of the solutions identified. In Table 6, the first column is the reference algorithm”.

As can be seen, in the generalized spread metric, the reference algorithm is statistically better in one of nine problems compared to the MOEA/D-AACPBIShOP Algorithm and compared to the NSGA-II/BIShOP it has a lower performance.

3.1.2.1 Friedman Test

“The p-value calculated with the Friedman test is 0.09697196786440554, so with a level of statistical significance of 5%, it is significant. Table 7 below shows the average ranks per algorithm obtained with the Friedman test”. The Friedman test suggests that no algorithm differs significantly. Therefore, the approximate Pareto fronts obtained in the three algorithms have similar performance.

ISSN 2007-9737

Table 7. Average ranks for GS

Algorithm	AVG Rank
NSGA-II/BIShOP	1.44
MOEA/D-AACPBIShOP	2.11
MOEA/D-BIShOP	2.44

Table 8. Results IGD (median and IQR values)

Problem	MOEA/D-BIShOP	MOEA/D-AACPBIShOP	NSGA-II/BIShOP
3n20m	6.94e-01 1.93e-01	6.94e-01 2.20e-01 ==	6.65e-01 2.57e-01▼
4n20m	1.58e+00 8.64e-01	1.59e+00 8.74e-01 ▲	1.53e+00 9.49e-01▼
5n20m	9.53e-01 6.41e-01	9.60e-01 6.45e-01 ▲	8.92e-01 7.43e-01▼
5n240m	1.87e+00 1.25e+00	1.89e+00 1.16e+00 ▲	1.83e+00 1.30e+00▼
5n400m	1.77e+00 1.23e+00	1.79e+00 1.12e+00 ▲	1.73e+00 1.39e+00▼
50n240m	1.34e+154 0.00e+00	1.34e+154 0.00e+00 ==	1.34e+154 0.00e+00 ==
50n400m	1.34e+154 0.00e+00	1.34e+154 0.00e+00 ==	1.34e+154 0.00e+00 ==
100n240m	1.34e+154 0.00e+00	1.34e+154 0.00e+00 ==	1.34e+154 0.00e+00 ==
100n400m	1.34e+154 0.00e+00	1.34e+154 0.00e+00 ==	1.34e+154 0.00e+00 ==

Table 9. Average ranks for IGD

Algorithm	AVG Rank
NSGA-II/BIShOP	2
MOEA/D-AACPBIShOP	2
MOEA/D-BIShOP	2

3.1.3 Inverted Generational Distance

"The inverted generation distance (IGD) gives the average distance between any point in the reference set and its nearest point in the approximation set [18]. In Table 8, the second column is considered as the reference algorithm".

As can be seen, in the generalized spread metric, the reference algorithm is statistically better in four of nine problems compared to the MOEA/D-AACPBIShOP Algorithm and compared to the NSGA-II/BIShOP it has a lower performance.

3.1.3.1 Friedman Test

"The p-value calculated with the Friedman test is 1.0, so with a level of statistical significance of 5%, it is not significant. Table 9 below shows the average ranks per algorithm obtained with the Friedman test". The Friedman test suggests that no algorithm differs significantly. Therefore, the inverted generation distance metric indicates that the three algorithms find the best solution in fewer iterations.

4 Conclusions and Future Work

Finally, with the results obtained, it is observed that the three proposed multi-objective algorithms have a statistically similar performance in three evaluated metrics, which suggests that the algorithms have good dispersion in the solutions and a similar convergence.

Therefore, it is assumed that by using other genetic operators and including new elements the performance of these new BIShOP solution methods can be improved.

This paper proposes future work to explore and develop genetic operators. They would also be very useful in online stores, Internet search engines, and other complex problems similar to BIShOP.

These tools allow Internet searches to be carried out considering more than one attribute at a time and allow more than one solution to be chosen that can provide great benefits to users and companies.

Acknowledgments

The authors thank CONAHCyT, Mexico, for the support provided through scholarships for postgraduate studies and the National System of Researchers.

References

1. **Chaerani, D., Rusyaman, E., Mahrudinda, Marcia, A., Fridayana, A. (2021).** Adjustable robust counterpart optimization model for internet shopping online problem. Journal of Physics: Conference Series, Vol. 1722, No. 1, pp. 012074. DOI: 10.1088/1742-6596/1722/1/012074.
2. **Zamir, M., Ali, N., Naseem, A., Frasteen, A. A., Zafar, B., Assam, M., Othman, M., Attia, E. (2022).** Face detection & recognition from images & videos based on CNN & raspberry pi. Computation, Vol. 10, No. 9, pp. 148. DOI: 10.3390/computation10090148.
3. **Afzal, K., Tariq, R., Aadil, F., Iqbal, Z., Ali, N., Sajid, M. (2021).** An optimized and efficient routing protocol application for IoV. Mathematical Problems in Engineering, Vol. 2021, pp. 1–32. DOI: 10.1155/2021/9977252.
4. **Malik, U. M., Javed, M. A., Zeadally, S., Islam, S. U. (2022).** Energy-efficient fog computing for 6g-enabled massive IoT: Recent trends and future opportunities. IEEE Internet of Things Journal, Vol. 9, No. 16, pp. 14572–14594. DOI: 10.1109/jiot.2021.3068056.
5. **Kumar, S. (2015).** Online shopping-a literature review. Proceedings of National Conference on Innovative Trends in Computer Science Engineering. pp. 129–131.
6. **Fraire-Huacuja, H. J., García-Morales, M. Á., López-Locés, M. C., Gómez-Santillán, C. G., Cruz-Reyes, L. Morales-Rodríguez, M. L. (2021).** Optimization of the internet shopping problem with shipping costs. Studies in Computational Intelligence, pp. 249–255. DOI: 10.1007/978-3-030-68776-2_14.
7. **García-Morales, M. Á., Fraire-Huacuja, H. J., Frausto-Solís, J., Cruz-Reyes, L., Gómez-Santillán, C. G. (2023).** A survey of models and solution methods for the internet shopping optimization problem. Studies In: Castillo, O., Melin, P. (eds) Fuzzy Logic and Neural Networks for Hybrid Intelligent System Design. Studies in Computational Intelligence, Springer, Cham., Vol. 1061, pp. 105–122. DOI: 10.1007/978-3-031-22042-5_6.
8. **Chung, J. B. (2017).** Internet shopping optimization problem with delivery constraints. Distribution Science Research, Vol. 15, No. 2, pp. 15–20.
9. **Chaerani, D., Rusyaman, E., Marcia, A., Fridayana, A. (2021).** Adjustable robust counterpart optimization model for internet shopping online problem. Journal of Physics: Conference Series, Vol. 1722, No. 1, pp. 012074. DOI: 10.1088/1742-6596/1722/1/012074.
10. **García-Morales, M. A., Brambila-Hernández, J. A., Fraire-Huacuja, H. J., Frausto-Solis, J., Cruz-Reyes, L., Gómez-Santillan, C. G., Valadez, J. M. C., Aguirre-Lam, M. A. (2024).** Multi-objective evolutionary algorithm based on decomposition to solve the bi-objective internet shopping optimization problem (MOEA/D-BIShOP). Lecture Notes in Computer Science, pp. 326–336. DOI: 10.1007/978-3-031-51940-6_24.
11. **Holland, J. H. (1975).** Adaptation in natural and artificial systems. University of Michigan Press, And Arbor, Michigan.
12. **Umbakar, A. J., Sheth, P. D. (2015).** Crossover operators in genetic algorithms: A review. ICTACT journal on soft computing, Vol. 06, No. 01, pp. 1083–1092. DOI: 10.21917/ijsc.2015.0150.
13. **Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000).** A fastelitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Lecture Notes in Computer Science, pp. 849–858. DOI: 10.1007/3-540-45356-3_83.
14. **Zhang, Q., Li, H. (2007).** MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, Vol. 11, No. 6, pp. 712–731. DOI: 10.1109/tevc.2007.892759.

ISSN 2007-9737

15. **Eiben, A., Smith, J. (2015).** Introduction to evolutionary computing. Natural Computing Series. Berlin, DOI: 10.1007/978-3-662-44874-8.
16. **Li, H., Zhang, Q. (2009).** Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 13, No. 2, pp. 284–302. DOI: 10.1109/tevc.2008.925798.
17. **García, C. (2020).** A cellular evolutionary algorithm to tackle constrained multiobjective optimization problems. Tesis de maestría, Instituto Nacional de Astrofísica, Óptica y Electrónica. Repositorio institucional del INAOE https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/2155/1/Mc_Thesis_Cosijopii.pdf.
18. **Brambila-Hernández, J. A., García-Morales, M. Á., Fraire-Huacuja, H. J., Angel, A. B. D., Villegas-Huerta, E., Carbajal-López, R. (2023).** Experimental evaluation of adaptive operators selection methods for the dynamic multiobjective evolutionary algorithm based on decomposition (DMOEA/D). In: Castillo, O., Melin, P. (eds) Hybrid Intelligent Systems Based on Extensions of Fuzzy Logic, Neural Networks and Metaheuristics. Studies in Computational Intelligence, Vol. 1096, pp. 307—330. DOI: 10.1007/978-3-031-28999-6_20.
19. **Jiang, S., Yang, S., Yao, X., Tan, K. C., Kaiser, M., Krasnogor, N. (2018).** Benchmark problems for CEC2018 competition on dynamic multiobjective optimisation. Newcastle University.

Article received on 18/01/2024; accepted on 08/05/2024.
Corresponding author is Héctor J. Fraire-Huacuja.