IJIO
(*International Journal of Industrial Optimization*)
ISSN 2723-3022 (online) | 2714-6006 (printed)
VOL 3, No. 1, 47-67
https://doi.org/10.12928/ijio.v3i1.5862

An estimation of distribution algorithm for combinatorial optimization problems

Ricardo Pérez-Rodríguez*
CONACYT – UAQ, Autonomous University of Queretaro, Mexico
*Corresponding Author: dr.ricardo.perez.rodriguez@gmail.com

ARTICLE INFO

Article history
Received: March 18, 2022
Revised: April 5, 2022
Accepted: April 14, 2022

Keywords
Estimation of distribution algorithm;
Mallows model;
Moth-flame algorithm;
Job shop scheduling problem;
Vehicle routing problem with time windows.

ABSTRACT

This paper considers solving more than one combinatorial problem considered some of the most difficult to solve in the combinatorial optimization field, such as the job shop scheduling problem (JSSP), the vehicle routing problem with time windows (VRPTW), and the quay crane scheduling problem (QCSP). A hybrid metaheuristic algorithm that integrates the Mallows model and the Moth-flame algorithm solves these problems. Through an exponential function, the Mallows model emulates the solution space distribution for the problems; meanwhile, the Moth-flame algorithm is in charge of determining how to produce the offspring by a geometric function that helps identify the new solutions. The proposed metaheuristic, called HEDAMMF (Hybrid Estimation of Distribution Algorithm with Mallows model and Moth-Flame algorithm), improves the performance of recent algorithms. Although knowing the algebra of permutations is required to understand the proposed metaheuristic, utilizing the HEDAMMF is justified because certain problems are fixed differently under different circumstances. These problems do not share the same objective function (fitness) and/or the same constraints. Therefore, it is not possible to use a single model problem. The aforementioned approach is able to outperform recent algorithms under different metrics for these three combinatorial problems. Finally, it is possible to conclude that the hybrid metaheuristics have a better performance, or equal in effectiveness than recent algorithms.

INTRODUCTION

Metaheuristics have been used to solve scale optimization problems in different engineering and science fields. One example about the effectiveness of such metaheuristics can be found in [1], for the integration between process planning and job shop scheduling. In [1] the communication concept between species is implemented for the aforementioned issue. Rossi and Dini [2] studied an applied case that was a flexible job shop scheduling with routing flexibility and setup times. As the proposed metaheuristic, an ant colony optimization-based approach is considered. Recent publications are found in [3]. The authors use a metaheuristic based on a genetic algorithm with Pareto-front. The aforementioned scheme works in a mail-order pharmacy system. Yue et al. [4] utilized a bee colony scheme with Pareto-front to tackle the single machine group-scheduling problem with sequence-dependent setup times. Huang et al. [5] developed an improved discrete particle swarm optimization method for the flexible job shop scheduling problem. Xu et al. [6] proposed a bat algorithm for solving the dual flexible job shop scheduling problem. Moreover, the application of metaheuristics is very remarkable. The main objective is to identify a better performance against other schemes. Prins [7] outperformed the drawbacks in vehicle routing problems. His proposal occupies a genetic algorithm and local search. Gao et al. [8] employed a genetic algorithm, too, but combined with a variable neighborhood descent method for flexible job shop scheduling problems. Zeng and Yang [9] hybridized a genetic algorithm with a neural network, enhancing the container-loading processes. A simulation validation technique gives support to their proposal. Lee et al. [10] also employed a genetic algorithm, it is hybridized with a minimum cost flow network model to improve the service activity in transport. A comparison was executed to show that their proposed scheme outperforms a neighborhood search algorithm. Schittekat et al. [11] proposed another metaheuristic to solve the school bus routing problem with bus stop selection. A parameter-free greedy randomized adaptive search procedure combined with a variable neighborhood descent method. Li and Gao [12] proposed a genetic algorithm and tabu search to solve flexible job shop scheduling open problems. Several recent studies [13-15] had other important contributions in the field. With this revision, it is clear that a wide set of metaheuristics has been effectively used for tackling diverse combinatorial problems.

Another branch of metaheuristics is the estimation of distribution algorithms (EDAs). Efficient EDAs can be found in [16] to solve permutation flow shop scheduling problems. Chen et al. [17] contributed to constructing EDAs for single machine scheduling problems. Pan and Ruiz [18] utilizes an EDA to solve lot-streaming flow shop problems with setup times. Based on the previous papers, EDAs have effectively tackled diverse combinatorial problems.

As with any metaheuristic, the EDA performance requires identifying each individual's fitness by generation from a set of them. Nevertheless, in the EDAs, a probability distribution is built based on population to generate new offspring. In addition, the EDA performance requires identifying the best probability distribution that fits with respect to the studied problem. Finding the best probability distribution is the aim of any EDA. Normally, the distribution is estimated by employing the statistical information of the population. Based on papers such as [19-20], a good estimated distribution helps to enhance the EDA performance. A way to identify a good distribution is by employing relations among independent variables. The proposal is to find high-order estimations between them. The interactions between variables help to build complex probability models. However, complex models are not necessarily the best option because it might be difficult to understand. In addition, the most important EDAs drawbacks, such as lack of diversity of the solutions, and poor exploitation ability [19], are also difficult to handle. To tackle the aforementioned drawbacks, hybridization plays an important role in many EDAs´papers.

Based on the results shown below, the hybridization between EDAs and other techniques improves EDA performance. Such hybridization should be considered in the design of any EDA. The hybridization also helps to tackle the aforementioned drawbacks of any EDA. in addition, the hybridization also might help to avoid the use of complex models. Finally, with the hybridization, any EDA could be more understandable.

This research aims to show how metaheuristics help tackle any EDA's drawbacks. The metaheuristics should consider some well-defined math expression because it permits reproducing the proposed scheme, helps comprehend how the distribution works, and how the offspring are obtained. This study aims to use new metaheuristics, mainly how the proposed scheme explicitly executes the search process. The proposal for doing it is to identify what is most useful, i.e., using a hybridized EDA, or the recent algorithms.

More than one type of combinatorial problem is considered in this article to understand the role hybridization plays in EDAs. The job shop scheduling problem (JSSP), the vehicle routing

 10.12928/ijio.v3i1.5862

problem with time windows (VRPTW), and the quay crane scheduling problem (QCSP) are solved with the proposed hybrid EDA. These discrete combinatorial optimization problems were selected due to their complex nature. These problems are considered the most difficult to solve in the combinatorial optimization field. The number of iterations exponentially increases based on the size of the problem, i.e., $O(n!)$. Due to these NP-hard problems [21], it is necessary to develop new methods of dealing with them. In addition, the discrete combinatorial optimization problems mentioned are suitable to use a permutation-based representation in the solutions. Furthermore, this study considers more than one discrete combinatorial problem to generate solid conclusions about the proposed metaheuristic.

1. Complex distribution models

A main challenge in creating new EDAs using complex distribution models is how to estimate high-order interactions between variables. The Univariate Marginal Distribution Algorithm (UMDA) proposed by [22] is an EDA example if there is no relation between variables. The Mutual Information Maximization for Input Clustering (MIMIC) detailed by [23] has good performance if there exists a relation between pair of variables. If we create new EDAs with high-order interactions between variables, new restrictions need to be satisfied, such as more population members. It could do difficult its understanding and/or computing. The Combining Optimizers with Mutual Information Trees (COMIT) published by [24], and the Bayesian Optimization Algorithm (BOA) proposed by [25] fall in this category.

EDAs with high-order relations among variables continue improving. A mix between complex distribution models can be found in the literature. In [26], three complex models are used to tackle the flexible job shop scheduling problem. Ceberio et al. [27] showed how the EDA complexity increases when higher-order interactions between variables are used.

2. The hybridization approach

Wide and diverse evolutionary algorithms have been published considering hybridization features at the core of their structure. Studies of [28] face the quay crane-scheduling problem using a hybrid genetic algorithm. Then, Figliozzi [29] tackle the VRP with Soft Time Windows (VRPSTW), and Hard Time Windows (VRPHTW) using an algorithm-based on route construction and improvement. Next, Kamkar et al. [30] solved the VRPTW by a cellular Genetic Algorithm (cGA). In addition, Kaveshgar et al. [31] faced the quay crane-scheduling problem by a hybrid genetic algorithm with heuristics. Then, Chung et al. [32] also faced the quay crane-scheduling problem through a hybrid genetic algorithm. Next, Tas et al. [33] tackle a vehicle routing problem with soft time windows and stochastic travel times through a tabu search (TS) method. In addition, Chung and Chan [34] face the quay crane-scheduling problem using the hybridization between genetic algorithm and fuzzy logic. Next, Vidal et al. [35] solve large-scale vehicle routing problems by the hybrid genetic algorithm (HGA). Then, Garg [36] optimize an industrial system with uncertain data, using a hybrid genetic algorithm coupled with a gravitational search algorithm. Next, Phanden and Jain [37] solved the flexible jobshop-scheduling problem with process plan flexibility, by a simulation-based genetic algorithm approach. Then, Li and Gao [12] tackled the flexible job shop scheduling problem, showing a hybrid genetic algorithm and tabu search. Then, Garg [38] solve the constrained optimization problems through a particle-based swarm method and a genetic algorithm. In addition, Xu et al. [6] resolved the flexible job shop scheduling problem with process plan flexibility through a bat algorithm, and Garg [39] addressed the constraint nonlinear optimization problems with mixed variables by a hybrid gravitational search algorithm coupled with a genetic algorithm.

Based on the previous examples, an improvement feature is included in the hybrid EDAs. The most popular techniques to enhance the EDA are heuristics or metaheuristics. Peña [40] proposed a genetic algorithm and EDAs. The goal is to take benefit from both schemes. The proposed hybrid EDA is used for resolving synthetic optimizations problems and two real-world cases. Then, Zhang et al. [41] used a metaheuristic too, by an EDA with a 2-opt local search, to tackle the quadratic assignment problem. In addition, Liu et al. [42] studed the permutation flowshop scheduling problem through a metaheuristic too, an EDA, and the particle swarm optimization method. Then, Wang et al. [19] faced out the flexible job shop scheduling problem through a metaheuristic too, an EDA, and a local search strategy to improve the EDA performance. Next, Fang et al. [43] also proposed to solve the stochastic resource-constrained project-scheduling problem with a metaheuristic by an EDA and a permutation-based local search. Then, Wang et al. [44] also presented a metaheuristic to tackle the distributed permutation flowshop scheduling problems under machine breakdown, by and EDA, and a fuzzy logic technique.

3. Ranking models

For the combinatorial optimization problems, it is suitable to develop new EDAs by already defined distribution models for rankings. Such as the proposed EDAs by [45] for the flow shop scheduling problem, [46] for the school bus routing problem with bus stop selection, [47] for the flexible job shop scheduling problem with process plan flexibility, and [48] for the vehicle routing problem with time windows. These investigations define distribution model for rankings, i.e., the Mallows model.

4. Research´s gap

Based on the previous review, there exists a gap to enhance the EDA performance. The metaheuristics are competitive approaches to improve any EDA. Therefore, this research continues in this direction showing that hybrid metaheuristics have a better performance or equal in effectiveness than the recent algorithms.

The hybridization employed in this investigation is based on the Mallows model and the Moth-flame technique [49]. The exponential models sparingly have been used to enhance the EDA performance. The available papers are scarce in the literature. Ceberio et al. [45] introduced an EDA based on the Mallows model for the flow shop scheduling problem. The Mallows model helps build the distribution probability through permutations as solutions.

The representation of the solutions in this research is rankings or permutations as a first step. It is a suitable representation for the problems above, i.e., the JSSP, the VRPTW, and the QCSP. Then, by the algebra of permutations, a distance metric is necessary by computing between the solutions as a second step. An exponential distribution is built with the previous distance metric as a third step. An input parameter, i.e., the distance between each of the solutions and a central solution should be computed. The Mallows model [50] assigns a probability to each solution that decays exponentially with respect to its distance to the central solution. Figure 1 details an example of it.

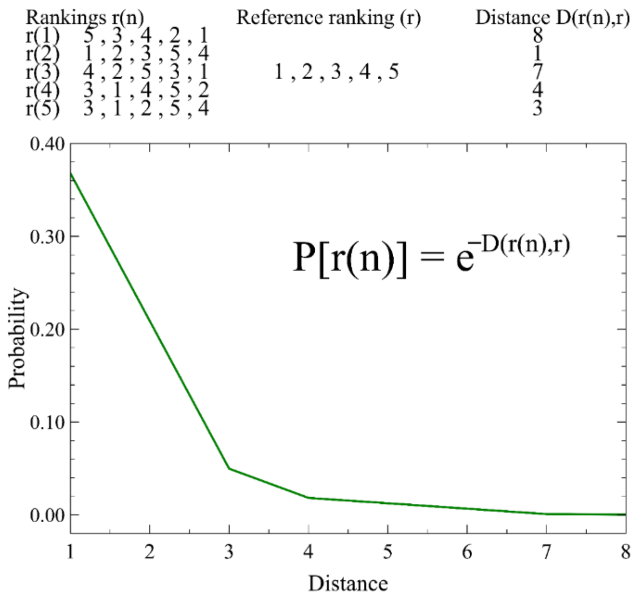

Figure 1. The Mallows model

 10.12928/ijio.v3i1.5862

The aforementioned distance should be factorized as the fourth step. Then, the aforementioned factorization is obtained by the generalized Mallows model (GMD), proposed by [51] and [52]. The factorization result contains $n-1$ elements, where n is the size of the rankings. Table 1 presents details the factorization process.

Table 1. Factorization process

Solutions	Central solution	Distance	Factorization result
3, 1, 4, 2, 5	1, 2, 3, 4, 5	3	[2 , 0 , 1 , 0]
5, 3, 2, 4, 1		8	[4 , 2 , 1 , 1]

The Moth-flame algorithm uses as input parameter the previous factorization in the fifth step. Here is the hybridization of this research between the EDA and the Moth-flame scheme. It helps to improve the EDA to reach the performance of recent algorithms. The Moth-flame scheme uses moths to execute the search process for the new solutions. It is necessary to know where the moths are located in the solution space. Therefore, the aforementioned factorization serves as the location coordinates for each moth. Figure 2 shows the relation previously explained.

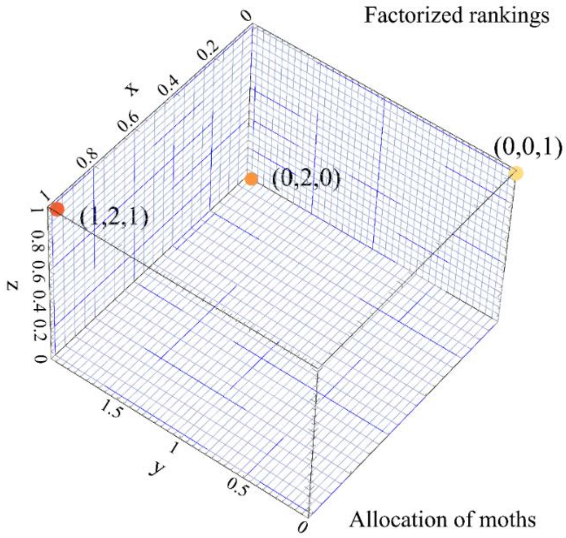

Figure 2. Moths´ location coordinates

The moths represent agents moving in the search space, whereas flames are the best position of moths obtained so far [49]. The moths move themselves to new locations by spiral way around flames. The distinctions between this study and past research are

- Through an exponential function, the Mallows model emulates the solution space distribution for the problems; meanwhile, the Moth-flame algorithm is in charge of determining how to produce the offspring.
- The previous research only uses the Mallows model directly, i.e., both the search and the offspring processes are obtained from the ranking model
- The previous contributions combine different methods to enhance the performance of the algorithm. That is, there is a main technique, and the other helps to tackle the drawbacks of the previous one. However, in this research, hybridization is the key feature to improving the performance of the proposed scheme. That is, the moth-flame technique not only helps to the ranking model, but also it contributes to finding the best solutions.

Finally, the objective in this research is to show a different metaheuristic to improve the EDA performance. The main contributions are:

- To manage the hybridization concept between the Mallows model and the Moth-flame algorithm for combinatorial problems to tackle the EDA drawbacks.
- To offer all the steps to be reproducible the proposed metaheuristic.

- To use a reference ranking model to enhance the EDA to find better solutions for the combinatorial problems.

RESEARCH METHODOLOGY

This research methodology will elaborate the HEDAMMF approach. The model of the JSSP can be found in [53]. The model of the VRPTW is available in [54]. The model of the QCSP is detailed in [55].

1. Initial population

As in other EDAs, the representation of the solutions is done by a permutation. This representation works for the three sets of issues above. In JSSP, a solution representation is shown by processing the sequence of operations on the available machines and the assignment of operations on the machines. Therefore, we use one vector to represent the operations sequence and another vector to depict the machine assignment.

Tabel 2. Operatings sequence

JSSP Vector 1 – Operations sequence					
Ranking	1st	2nd	3rd	4th	5th
Operations	3	1	4	2	5

Tabel 3. Machine assignment

JSSP Vector 2 – Machine assignment					
Ranking	1st	2nd	3rd	4th	5th
Machine	2	2	1	1	3

In VRPTW, a common representation is expressed by a vector that contains a sequence of all the customers to visit.

Tabel 4. VRPTW vector

VRPTW vector					
Ranking	1st	2nd	3rd	4th	5th
Vertices	3	1	4	2	5

In QCSP, a solution representation is expressed by the processing sequence of containers on the available quay cranes and the assignment of containers on the quay cranes. Two vectors do it, i.e., the containers sequence vector and the quay crane assignment vector. Both vectors are represented as the previous JSSP vectors. In this research, the initial population has 1000 individuals. The amount of individuals is a fixed parameter.

2. Fitness

The fitness for the JSSP is computed by the processing time required to complete all the operations. The fitness function formulation used to describe the processing time for all operations is found in [53]. The total distance computes the fitness for the VRPTW traveled that is required to attend all the customers. The fitness for the QCSP is calculated by the average waiting time required to give service to all the tasks.

3. JSSP machine distribution model

Once we have the machine's assignment vectors from the initial population, we apply the UMDA algorithm to get new offspring, i.e., through the marginal distribution that it is built with all the machine assignment vectors. Each value, for each position in the machine vector, depicts a specific selected machine for a specific operation. The cumulative probability, from marginal distribution, is used to sample the offspring. The process continues until the offspring vector is completed.

 10.12928/ijio.v3i1.5862

4. QCSP quay crane distribution model

Here the procedure is the same as the previous, i.e., once we have the quay crane assignment vectors from the initial population, again we apply the UMDA algorithm to get new offspring, i.e., through the marginal distribution that it is built with all the quay crane assignment vectors. Each value, in each quay crane assignment vector, represents a specific quay crane selected for a specific task. The cumulative probability, from marginal distribution, is used to sample the offspring. The process continues until the offspring vector is completed. The Figure 3 shows the process to sample the offspring.

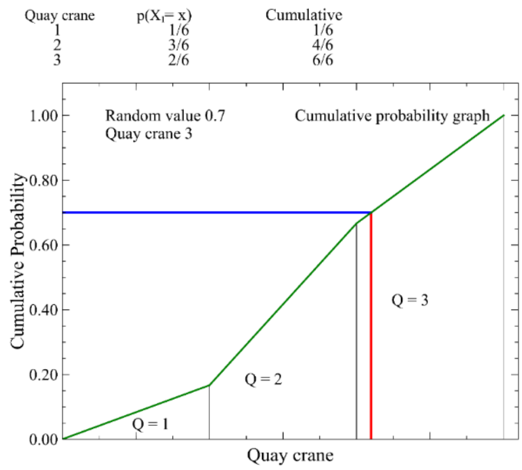

Figure 3. An example to obtain quay crane assignment vectors through the marginal distribution

5. The Mallows model

The Mallows model [50] is considered as the VRPTW distribution model. In addition, the Mallows model is addressed as the distribution model for the operations sequence vectors in the JSSP and the distribution model for the task sequence vectors in the QCSP.

5.1. Central solution

To compute the central solution, the first parameter of the Mallows model, is done with all individuals by [56] procedure. Figure 4 details a didactical example to obtain the central solution.

Solution vectors

4	2	5	3	1
1	4	3	2	5
3	2	1	4	5
5	3	4	1	2
2	5	4	3	1
4	5	2	1	3
4	1	3	5	2

Mean vector

3.1, 4.8, 2.2, 1.8, 4.2

Central ranking

3, 5, 2, 1, 4

Borda algorithm

Figure 4. A didactical example to obtain a central solution

5.2. Factorization computing

Once we have the central solution, it is possible to obtain the product between the central solution, and each vector. The product is computed by the procedure defined by [51] and [52]. Figure 5 shows a didactical example through the algebra of permutations.

Vector 1 - π

2 3 1 4

Central ranking - σ

4 3 1 2

Inverse of the Vector 1 - $\pi^{(-1)}$

3 1 2 4

Composition - $\sigma\pi^{(-1)}$

2 3 4 1

Figure 5. A didactical example to obtain the product between solutions

Once we have the aforementioned product, it factorizes in $n-1$ items. Figure 6 shows the procedure.

Composition

(2, 3, 4, 1)

Factorization...

1st element

1, because there is one value smaller than number 2 on its right

2nd element

1, because there is one value smaller than number 3 on its right

3rd element

1, because there is one value smaller than number 4 on its right

Output: Factorized vector

(1, 1, 1)

Figure 6. Factorization

6. The Moth-flame phase

6.1. The moths in their feasible space

Transfer the information of the aforementioned factorization process obtains the coordinates where a moth is allocated. As a didactical example, if we have a factorized vector as (1,1,1), the moth is allocated in the coordinate (1,1,1). Figure 2 provides a didactical example of it.

6.2. Moths fitness

As in the previous step, we transfer the fitness of the solution vector as the moth fitness. If we are talking to the fitness for the JSSP, it is computed by the processing time required to complete all the operations. Then, if we are talking to the fitness for the VRPTW, it is computed by the total distance traveled required to attend all the customers. Next, if we are talking to the fitness for the QCSP, it is calculated by the average waiting time required to give service to all the tasks.

 10.12928/ijio.v3i1.5862

6.3. Moths sorting

A descending order with respect to the fitness is done in the moths population. It is a process done by the bubble method.

6.4. Flames amount computing

Mirjalili [49] published a mathematical equation to compute

$$number\ of\ flames = round\left(N - l * \frac{N-1}{T}\right) \quad (1)$$

where N is the maximum number of flames, l is the current generation, and T indicates the maximum number of generations.

6.5. Flames setting

Transfer the information of the aforementioned moths sorting step obtains the coordinates where a flame is allocated. Table 5 shows a didactical example.

Table 5. Flames allocation example

Flames	Flame Coordinates	Sorted Moths	Moth Coordinates
1	(0,0,1)	1	(0,0,1)*
2	(3,1,0)	2	(3,1,0)*
3	(2,1,2)	3	(2,1,2)*
		4	(0,1,0)
		5	(1,1,1)
		6	(3,0,0)
		7	(2,2,0)
		8	(2,3,0)
		9	(1,0,0)

* the best moths

6.6. Flame fitness

We transfer the fitness of the moth as the flame fitness. At the beginning of the process, there is no difference between flame and moth fitness. However, a descending order with respect to the fitness is done in the flames population in each generation. It is necessary to establish which flames will be adequate for the orientation of the moths in the progress of the algorithm.

6.7. Moth-flame assignment

Every moth is assigned to only one flame. Table 6 presents the details a didactical example

Table 6. Assignment moth-flame

Flame	Moths
1	1
1	2
1	3
2	4
2	5
2	6
3	7
3	8
3	9

6.8. Moth movement

Every moth finds a new allocation through [49]. The function is below

$$Spiral\ function = D_i * e^{bt} * \cos(2\pi t) + F_j \qquad (2)$$

where b is a constant that defines the spiral shape, t is a random value from -1 to 1, D_i indicates the distance from the i-th moth and the j-th flame. Figure 7 illustrates the application of the expression above

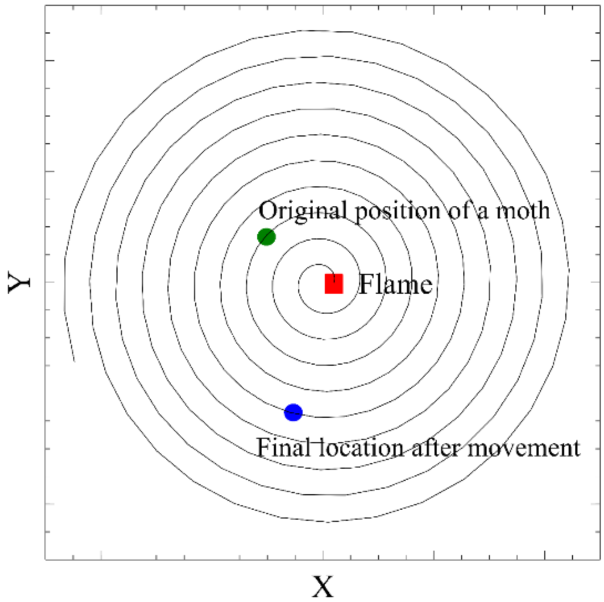

Figure 7. An example of the use of the spiral function

7. Offspring (genotype)

The allocations previously obtained are used as the factorization. The new positions of each moth represent the factorization above previously detailed. As an example, the coordinate (2, 0, 1) could represent an offspring (genotype).

8. Offspring (phenotype)

Meilă's [57] algorithm is used to obtain a new offspring by the genotype vectors and by means of the algebra of permutations. As an example, if the genotype is (2, 0, 1), then with the Meilă procedure, the phenotype is (2, 4, 1, 3).

9. Replacement

A binary tournament is done between parents and offspring to create a new population. The offspring must be previously evaluated to do the replacement, i.e., the offspring's fitness must be computed. As an example, if the parent 1 is compared against the offspring 1, and the parent 1 has less fitness value with respect to the offspring 1, then the parent 1 is selected to be part of the new population because we are looking for the smallest fitness values for the JSSP, VRPTW, and QCSP. We go back to step (section) 3, during 100 generations as a fixed parameter. A didactical illustration is shown in Figure 8 to depict the global process.

RESULTS AND DISCUSSION

Three sets of instances are occupied in this section to realize the key point about the hybrid proposed EDA. The instances used for the JSSP are datasets such as the [58] instances; the [59] instances; the [60] instances; the [61] instances; the [62] instances; and, the [63] instances. For the VRPTW, we use the dataset proposed by [64]. For the QCSP, the [65] instances served as input data. In addition, for the comparison, some JSSP methods are used, such as the [37] method, the [6] approach, the [12] scheme, and the [66] procedure.

 10.12928/ijio.v3i1.5862

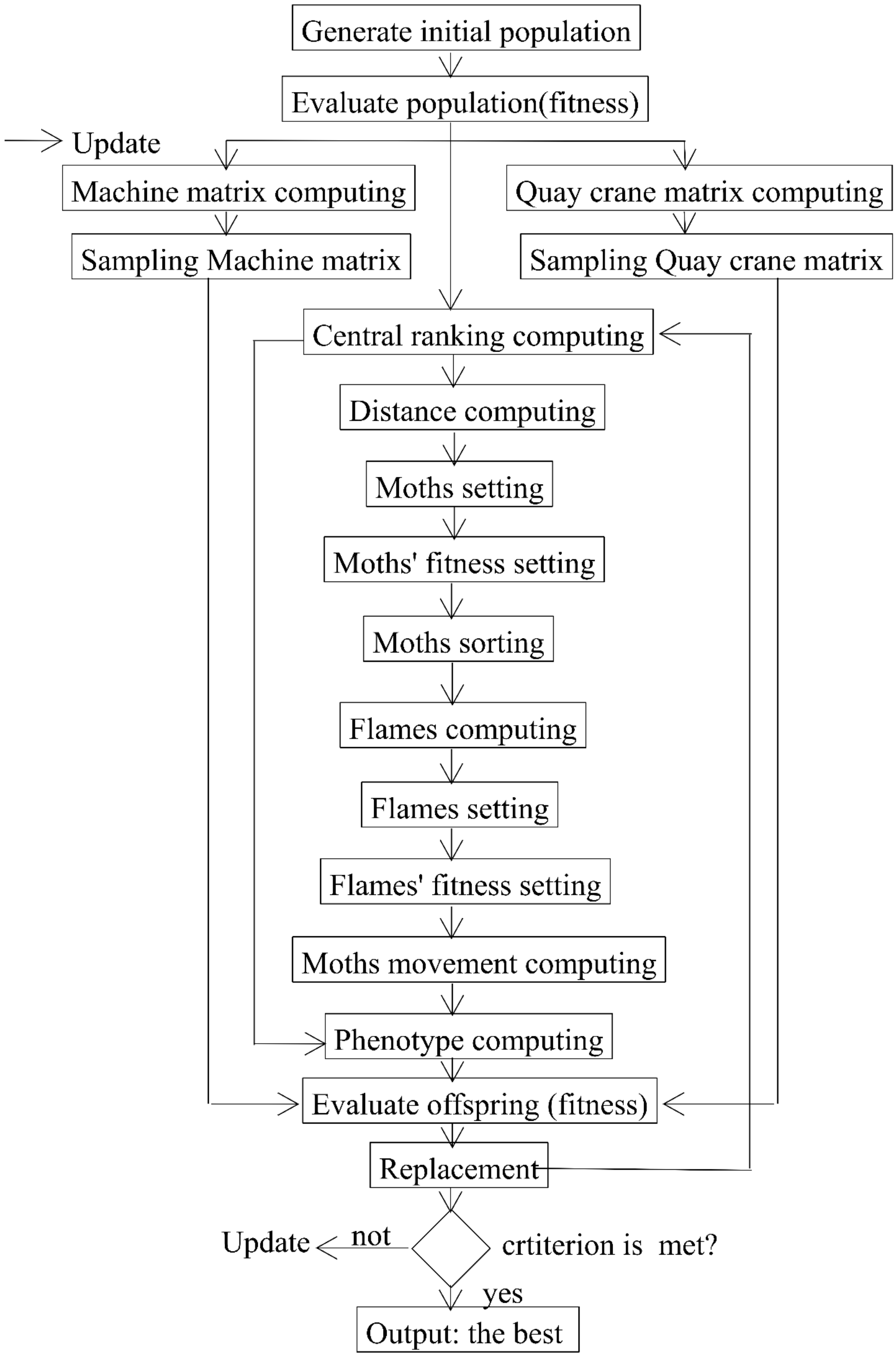

Figure 8. The HEDAMMF general procedure

The algorithms considered in the comparison for the VRPTW are the [33] method, the [35] approach, the [29] scheme, and the [30] procedure. The algorithms considered in the comparison for the QCSP are the [31] method, the [32] scheme, the [28] procedure, and the [34] approach. In addition, an EDA that works with the GMD process as a probability model is a participant in the comparison. The EDA above is labeled as ‘Mallows’ in the corresponding charts and it is considered a hybrid EDA. Finally, the HEDAMMF is as the new metaheuristic.

Three of the most important metrics are computed to evaluate the performance between the algorithms. The mean absolute error (MAE) is as follows.

$$MAE(c_i) = |c_i - c^+| \tag{3}$$

c_i is the fitness obtained in the i-th replication, and c^+ is the best fitness available for each instance used in this study. Then, the mean square error (MSE) is presented in Eq. 4 and the relative percentage increase (RPI) is shown in Eq. 5.

$$MSE(c_i) = (c_i - c^+)^2 \tag{4}$$

$$RPI(c_i) = (c_i - c^+)/c^+ \tag{5}$$

Figure 9 depicts the performance of the JSSP by Eq. (3). According to the results, the algorithm labeled as ‘Mallows’ obtain better results than the recent algorithms, such as the HEDAMMF scheme. Then, Figure 10 presents the details the algorithm's output for the JSSP with the Eq. (4). The aforementioned figure depicts similar results to the previous Figure 9. The HEDAMMF outperforms the other algorithms for the JSSP.

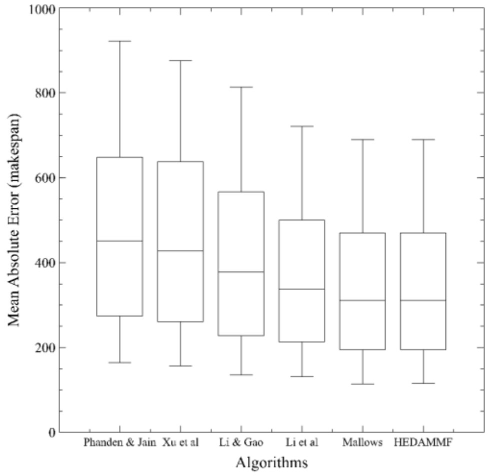

Figure 9. JSSP results by MAE

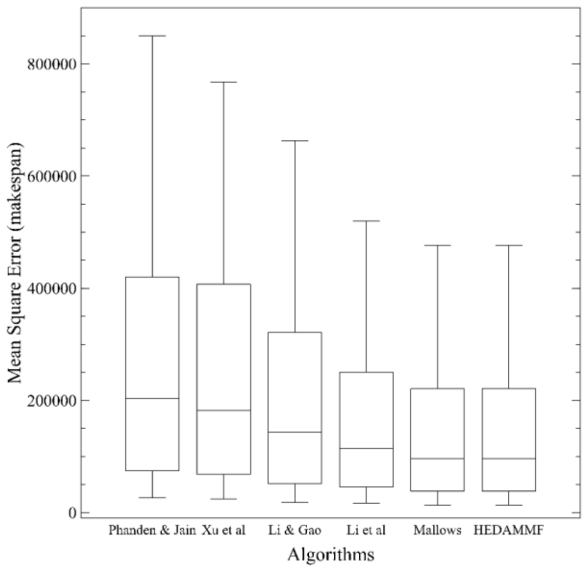

Figure 10. JSSP results by MSE

10.12928/ijio.v3i1.5862

Figure 11 shows the performance for the JSSP by Eq. (5). The algorithm labeled' Mallows' obtains better results than the recent ones. Furthermore, the algorithm 'Mallows' obtain similar results to the [66] algorithm and the HEDAMMF is the best proposal.

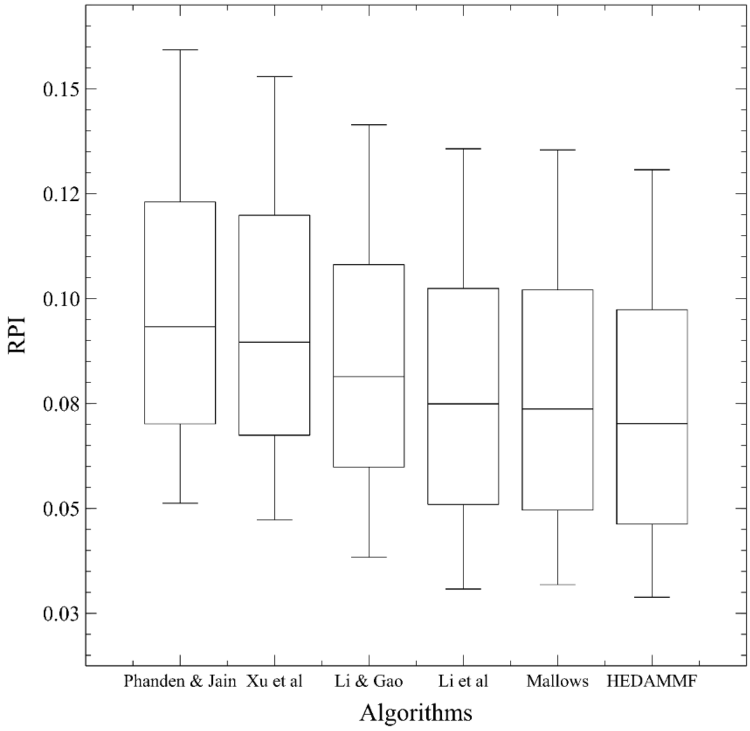

Figure 11. JSSP results by RPI

Figure 12 details the Dunnett test; there is a statistically significant difference between all the recent algorithms and the HEDAMMF scheme. The HEDAMMF scheme outperforms all the current algorithms for the JSSP. Based on the results, the Moth-flame scheme helps to get better performance for the HEDAMMF. Then, Figure 13 depicts the performance for the VRPTW by Eq. (3). According to the results, the HEDAMMF outperforms to the other algorithms. In addition, Figure 14 details the performance of the VRPTW by Eq. (4). The behavior is similar to Figure 13. The HEDAMMF outperforms to the other algorithms.

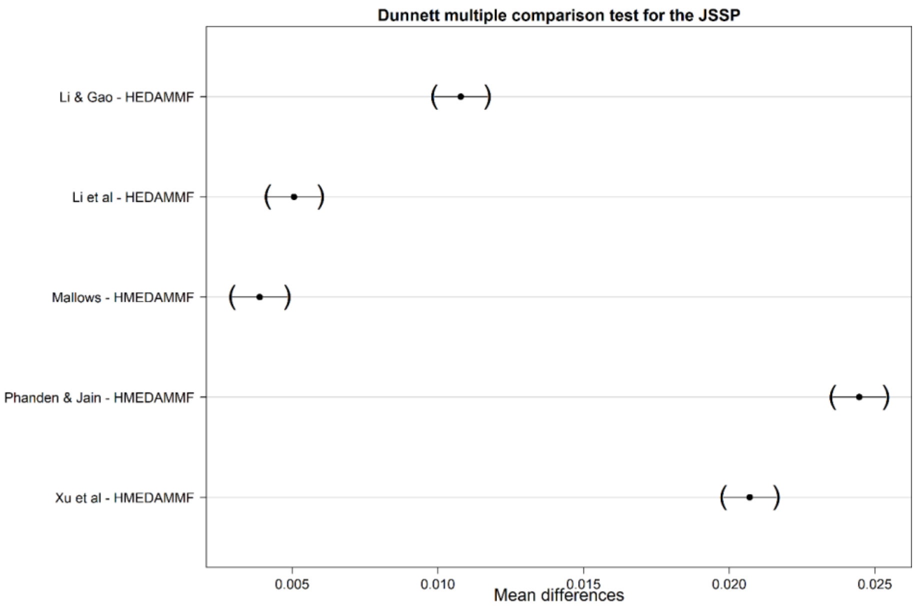

Figure 12. Dunnett test for the JSSP

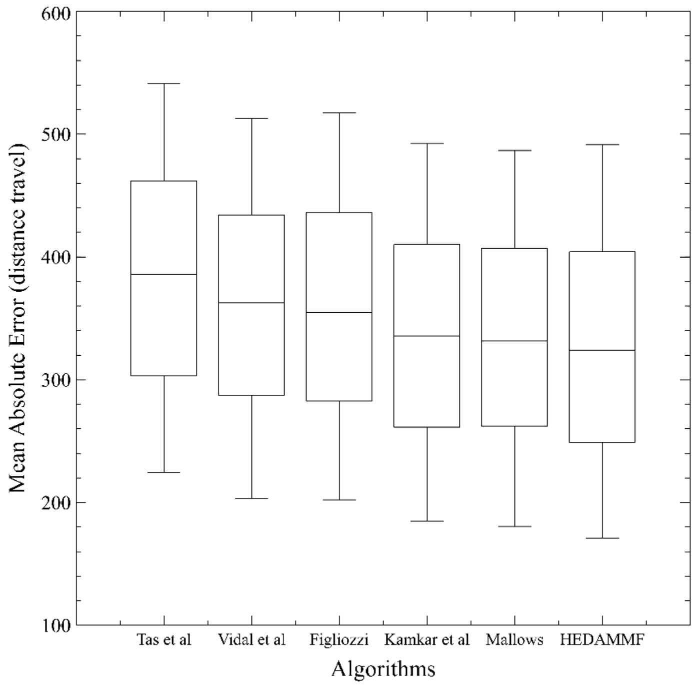

Figure 13. VRPTW results by MAE

Figure 14 shows the performance for the VRPTW by Eq. (5). The hybrid algorithm 'Mallows' outperforms almost all the recent algorithms. The algorithm 'Mallows' obtains practically similar results to the [30] algorithm; meanwhile, the scheme HEDAMMF is the best.

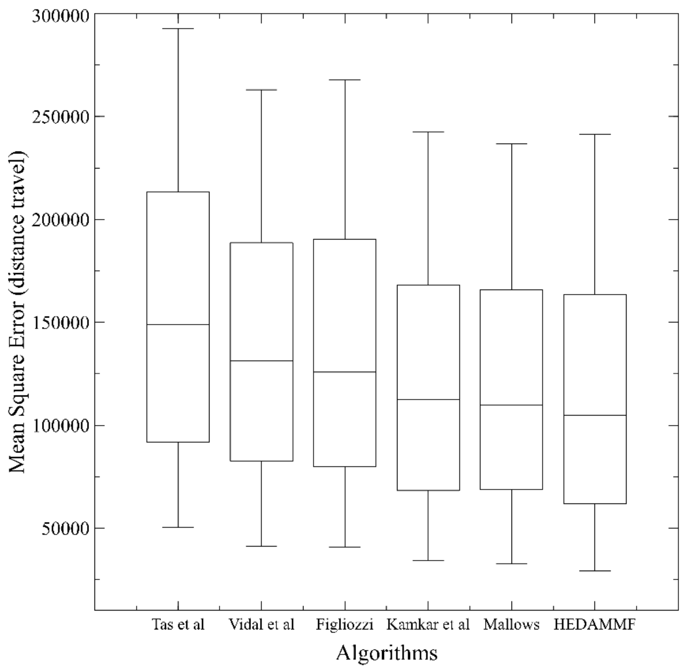

Figure 14. VRPTW results by MSE.

Figure 15 and Figure 16 present the RPI and Dunnett test; there is a statistically significant difference between all the recent algorithms and the HEDAMMF scheme. The HEDAMMF scheme outperforms all the recent algorithms for the VRPTW. The HEDAMMF is capable of finding the best results through the hybridization between the Mallows model, and the Moth-

10.12928/ijio.v3i1.5862

flame method. Figure 17 defines the performance for the QCSP by the Eq. (3). It shows the HEDAMMF outperforms to the other algorithms. In addition, Figure 18 details the performance of the QCSP by Eq. (4) and Figure 19 present the QCSP by Eq. (5). The HEDAMMF shows the best performance. Next, Figure 20 defines the performance of the QCSP by Eq. (5). The HEDAMMF approach is outstanding.

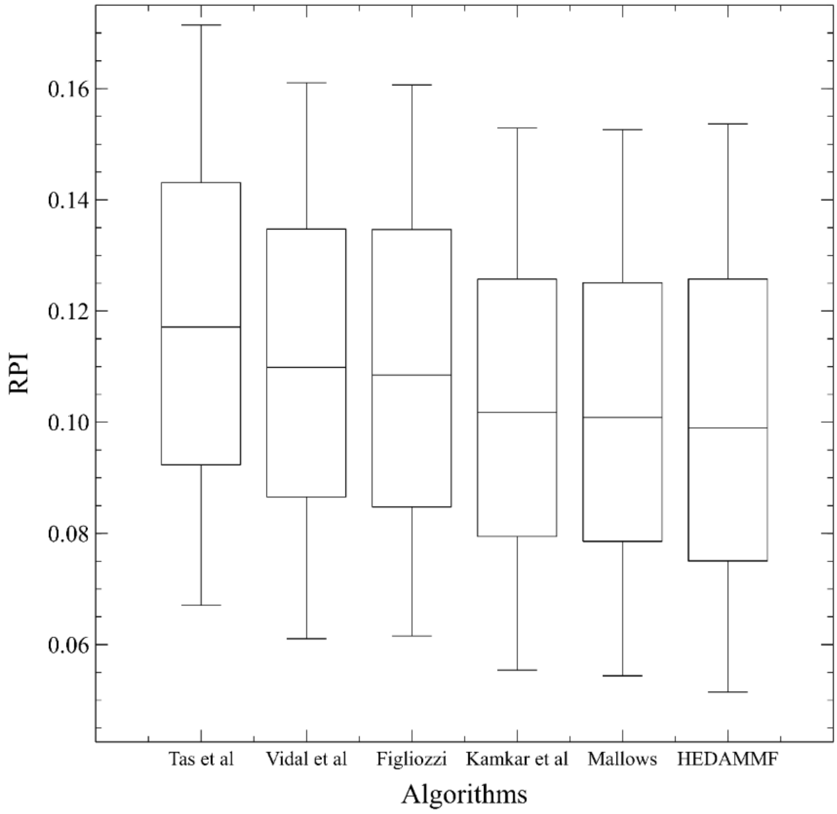

Figure 15. VRPTW results by RPI

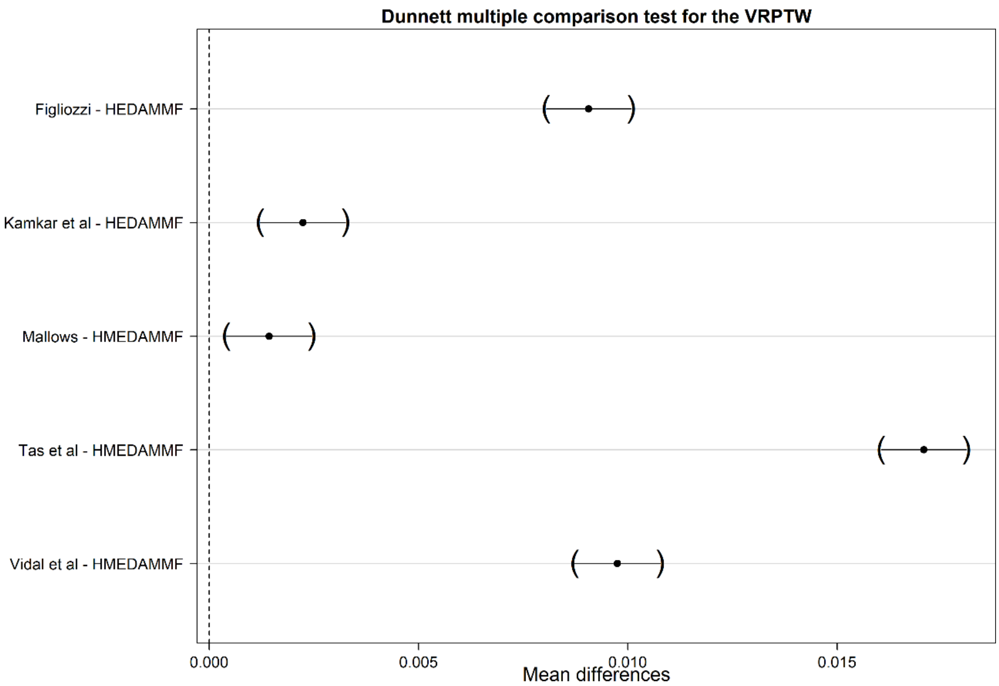

Figure 16. Dunnett test for the VRPTW

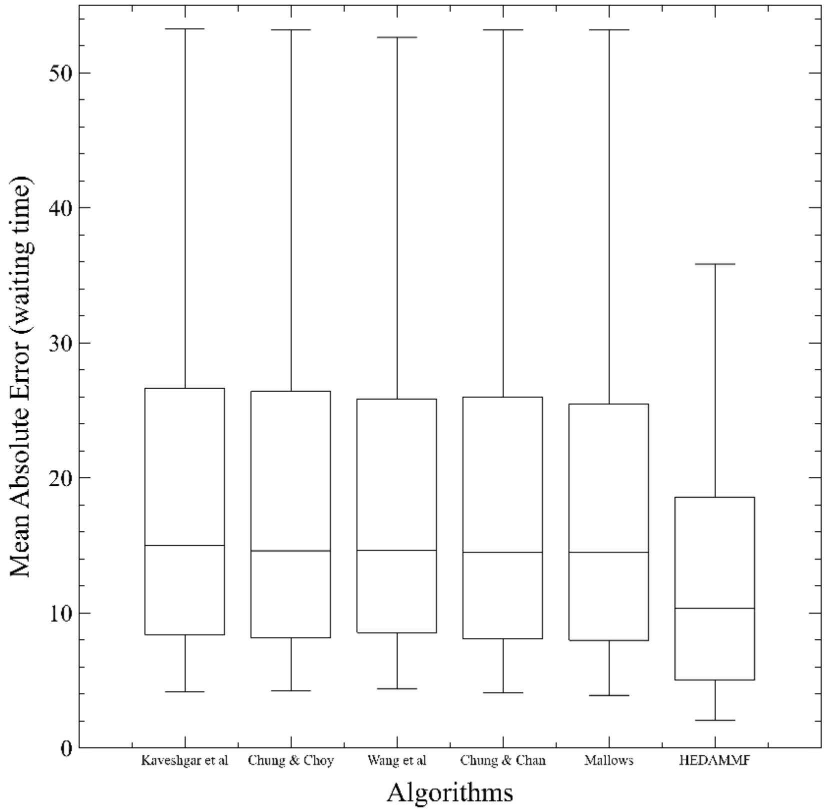

Figure 17. QCSP results by MAE

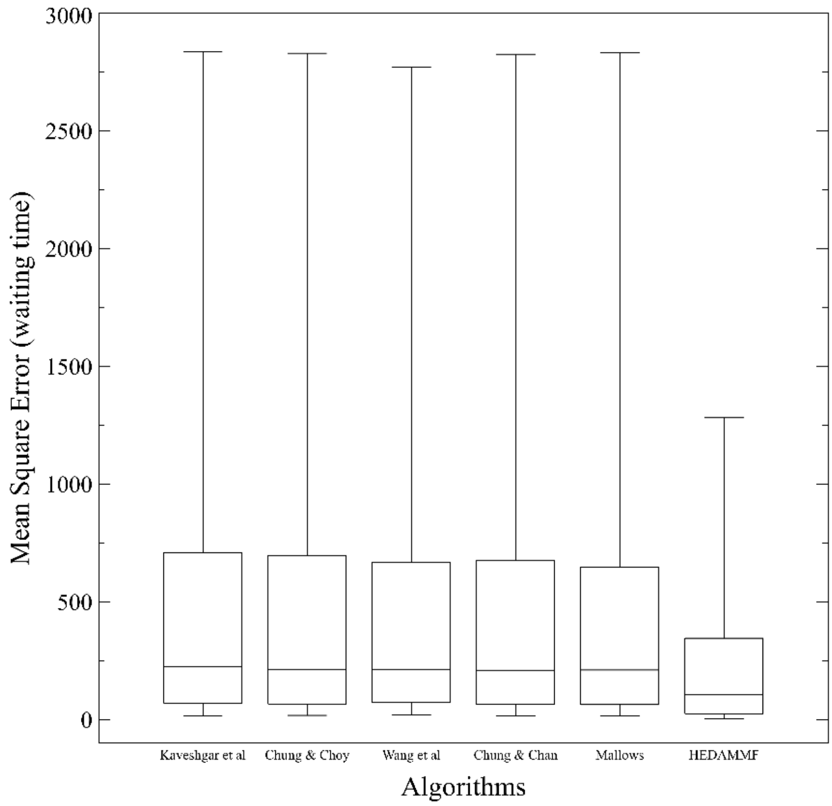

Figure 18. QCSP results by MSE

10.12928/ijio.v3i1.5862

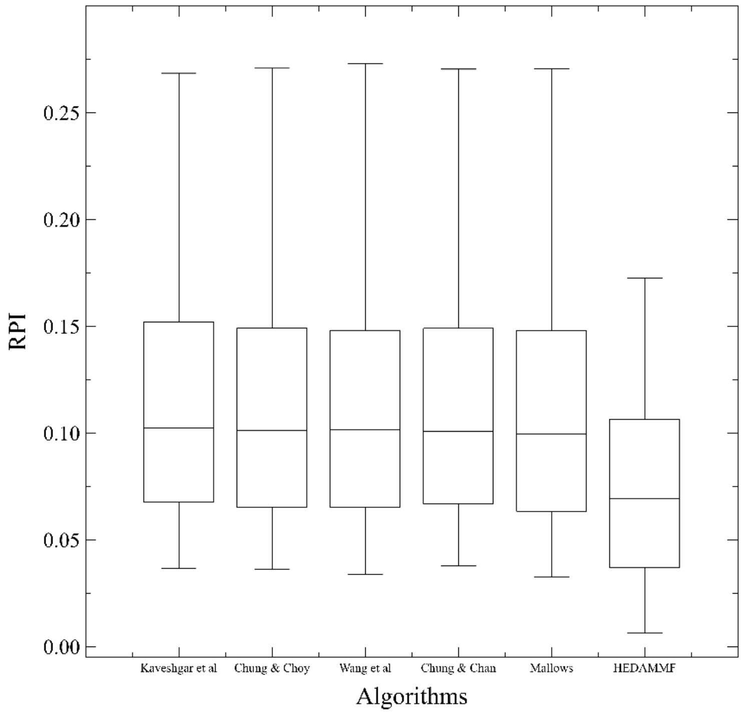

Figure 19. QCSP results by RPI

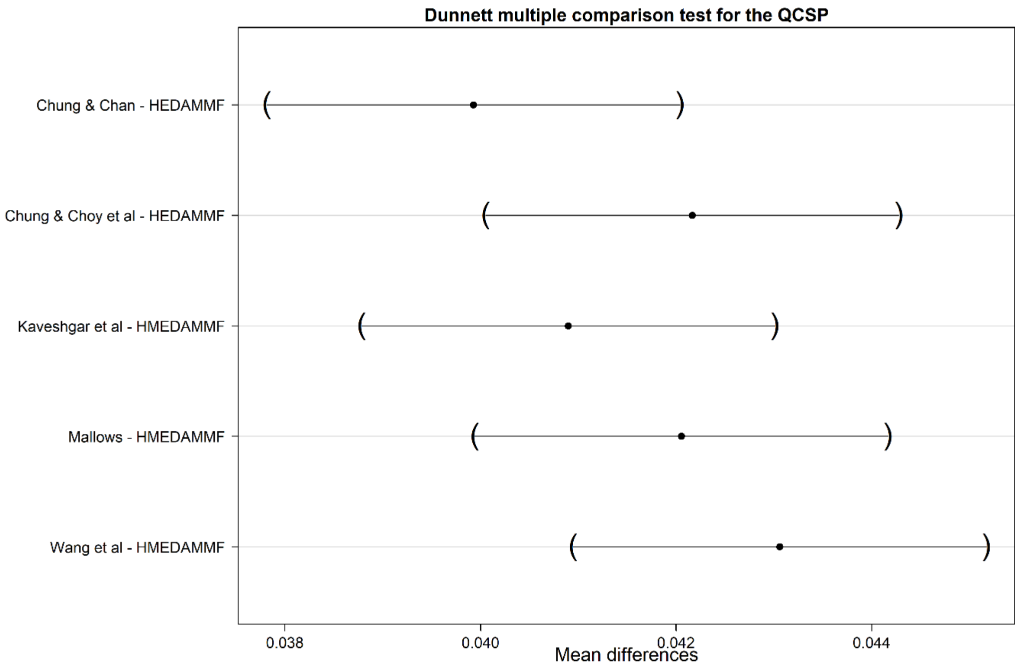

Figure 20. Dunnett test for the QCSP

The HEDAMMF is capable of tackling the EDAs drawbacks. No evolutionary operators are required to obtain new solutions. We only need to apply the Mallows model to find a suitable search process; meanwhile, the Moth-flame algorithm determines the allocation of the offspring.

As previously commented in the Introduction section, these problems are considered the most difficult to solve in the combinatorial optimization field. For this reason, it is necessary to develop new methods of dealing with them. To utilize the HEDAMMF is justified because certain problems are fixed differently under different circumstances. Finally, these problems do not share the same objective function (fitness) and/or the same constraints. Therefore, it is not possible to use a single model problem.

CONCLUSION

This paper considers solving more than one combinatorial problem such as the jobshop scheduling problem (JSSP), the vehicle routing problem with time windows (VRPTW) and, the quay crane scheduling problem (QCSP). These problems are considered some of the most difficult to solve in the combinatorial optimization field.

According to the previous results, the hybrid EDAs have a better behavior than the recent algorithms. We wish to conclude that estimations of the high-order interactions could be omitted if it is about constructing a better metaheuristic. The proposed metaheuristic is more competitive against other algorithms.

The predefined distribution models, such as the Mallows model, help improve the EDA scheme's performance. The hybridization concept between the Mallows model and the Moth-flame algorithm for combinatorial problems helps tackle the EDA drawbacks. The ranking model helps enhance the EDA to find better solutions for the combinatorial problems. The moth-flame technique not only helps the ranking model, also it contributes to finding the best solutions

Future research should consider other exponential models based on the results above. In addition, more combinatorial problems should be addressed using the proposed EDA, and other metaheuristics should be integrated to enhance the EDA results. Finally, real-world data should be considered to review the feasibility to implement the proposed EDA in final user platforms.

ACKNOWLEDGEMENT

Gratitude to all the reviewers for their comments on improving the manuscript. This research has not received external funding

REFERENCES

[1] Kim, Y., Park, K., & Ko, J. (2003). A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. Computers & Operations Research, 30, 1151-1171.

[2] Rossi, A., & Dini, G. (2007). Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. Robotics and Computer-Integrated Manufacturing, 23, 503-516.

[3] Dauod, H., Li, D., Yoon, S., & Srihari, K. (2016). Multi-objective optimization of the order scheduling problem in mail-order pharmacy automation systems. International Journal of Advanced Manufacturing Technology. 99, 73–83

[4] Yue, L., Guan, Z., Saif, U., Zhang, F., & Wang, H. (2016). Hybrid Pareto artificial bee colony algorithm for multi objective single machine group scheduling problem with sequence dependent setup times and learning effects. Springerplus, 5(1593).

[5] Huang, S., Tian, N., Wang, Y., & Ji, Z. (2016). Multi objective flexible job shop scheduling problem using modified discrete particle swarm optimization. Springerplus, 5(1432).

[6] Xu, H., Bao, Z., & Zhang, T. (2017). Solving dual flexible job-shop scheduling problem using a Bat Algorithm. Advances in Production Engineering & Management, 12(1), 5-16.

[7] Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31, 1985-2002.

[8] Gao, J., Sun, L., & Gen, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 35, 2892-2907.

 10.12928/ijio.v3i1.5862

[9] Zeng, Q., & Yang, Z. (2009). Integrating simulation and optimization to schedule loading operations in container terminals. Computers & Operations Research, 36, 1935-1944.

[10] Lee, L., Chew, E., Tan, K., & Wang, Y. (2010). Vehicle dispatching algorithms for container transshipment hubs. OR Spectrum, 32, 663-685.

[11] Schittekat, P., Kinable, J., Sörensen, K., Sevaux, M., & Spieksma, F. (2013). A metaheuristic for the school bus routing problem with bus stop selection. European Journal of Operational Research, 229, 518-528.

[12] Li, X., & Gao, L. (2016). An effective hybrid genetic algorithm and tabu search for flexible jobshop scheduling problem. International Journal of Production Economics, 174, 93-110.

[13] Raden, A.C.L., Mohd, N.A.R., Wan, M.F.W.M., & Chairul, S. (2019). Integrated Vendor–Buyer Lot-Sizing Model with Transportation and Quality Improvement Consideration under Just-in-Time Problem. Mathematics, 7(10), 944.

[14] Raden, A.C.L., Fairul, A.B.J., Chairul, S., Mohd, R.B.M., & Mohd, N.A.R. (2014). Incorporating Transportation Cost into Joint Economic Lot Size For Single Vendor-Buyer. Journal of Software, 9(5), 1313-1323.

[15] Leuveano, A.C., Bin Jafar, F.A., & Bin Muhamad, M.R. (2012). Development of genetic algorithm on multi-vendor integrated procurement-production system under shared transportation and just-in-time delivery system. 2nd International Conference on Uncertainty Reasoning and Knowledge Engineering, 2012, pp. 78-81.

[16] Jarboui, B., Eddaly, M., & Siarry, P. (2009). An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. Computers & Operations Research, 36, 2638-2646.

[17] Chen, S., Chen, M., Chang, P., Zhang, Q., & Chen, Y. (2010). Guidelines for developing effective Estimation of Distribution Algorithms in solving simple machine scheduling problems. Expert Systems with Applications, 37, 6441-6451.

[18] Pan, Q., & Ruiz, R. (2012). An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega, 40, 166-180.

[19] Wang, L., Wang, S., Xu, Y., Zhou, G., & Liu, M. (2012). A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. Computers & Industrial Engineering, 62, 917-926.

[20] Chen, Y., Chen , M., Chang, P., & Chen, S. (2012). Extended artificial chromosomes genetic algorithm for permutation flowshop scheduling problems. Computers & Industrial Engineering, 62, 536-545.

[21] González, T., & Sahni, S. (1978). Flowshop and jobshop schedules: complexity, and approximation. Operations Research, 26, 36–52.

[22] Mühlenbein, H. (1997). The equation for response to selection and its use for prediction. Evolutionary Computation, 5(3), 303-346.

[23] De Bonet, J., Isbell, C., & Viola, P. (1997). MIMIC: Finding Optima by Estimation Probability Densities. In Advances in Neural Information Processing Systems, (pp. 424–430).

[24] Baluja, S., & Davies, S. (1997). Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space. In D. Fisher (Ed.), Proceedings of the Fourteenth International Conference on Machine Learning, (pp. 30–38).

[25] Pelikan, M., Goldberg, D., & Cantú-Paz, E. (1998). Linkage problem, distribution estimation, and Bayesian networks. The University of Illinois, Genetic Algorithm Laboratory, Urbana-Champaign., Illinois.

[26] Pérez-Rodríguez, R., S, Jöns., Hernández-Aguirre, A., & Ochoa, C. (2014). Simulation optimization for a flexible jobshop scheduling problem using an estimation of distribution algorithm. International Journal of Advanced Manufacturing Technology, 73, 3-21.

[27] Ceberio, J., Irurozki, E., Mendiburu, A., & Lozano, J. (2012). A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. Progress in Artificial Intelligence, 1, 103-117.

[28] Wang, Y., Chen, Y., & Wang, k. (2009). A case study of genetic algorithms for quay crane scheduling. In B. Chien, & T. Hong (Ed.), Opportunities and challenges for next-generation applied intelligence. 214. Berlin, Heidelberg: Studies in Computational Intelligence, Springer.

[29] Figliozzi, M. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transportation Research Part C: Emerging Technologies, 18(5), 668-679.

[30] Kamkar, I., Poostchi, M., and Totonchi, M.R.A. (2010). A cellular genetic algorithm for solving the vehicle routing problem with time windows. In: Gao, X-Z., Gaspar, A., Köppen, M., Schaefer, G., and Wang, J. (eds.) Soft Computing in Industrial Applications, Advances in Intelligent and Soft Computing 75, Springer-Verlag, Berlin Heidelberg, pp. 263-270.

[31] Kaveshgar, N., Huynh, N., & Rahimian, S.-K. (2012). An efficient genetic algorithm for solving the quay crane scheduling problem. Expert Systems with Applications, 39(18), 13108-13117.

[32] Chung, S.-H., & Choy, K.-L. (2012). A modified genetic algorithm for quay crane scheduling operations. Expert Systems with Applications, 39(4), 4213-4221.

[33] Tas, D., Dellaert, N., van Woensel, T., and Kok, T. (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. Computers & Operations Research, 40(1), 214-224.

[34] Chung, S.-H., & Chan, T.-S. (2013). A workload balancing genetic algorithm for the quay crane scheduling problem. International Journal of Production Research, 51(16), 4820-4834.

[35] Vidal, T., Crainic, T., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers & Operations Research, 40(1), 475-489.

[36] Garg, H. (2014). A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data. In P. Vasant, Handbook of Research on Artificial Intelligence Techniques and Algorithms. IGI Global.

[37] Phanden R.K., & Jain, A. (2015). Assessment of makespan performance for flexible process plans in job shop scheduling. IFAC-PapersOnLine. 48(3):1948-1953.

[38] Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. Applied Mathematics and Computation, 274, 292-305.

[39] Garg, H. (2019). A hybrid GSA-GA algorithm for constrained optimization problems. Information Sciences, 478, 499-523.

[40] Peña, J., Robles, V., Larrañaga, P., Herves, V., Rosales, F., & Pérez, M. (2004). GA-EDA: Hybrid Evolutionary Algorithm Using Genetic and Estimation of Distribution Algorithms. In R. Orchard, C. Yang, & M. Ali (Ed.), IEA/AIE 2004, LNAI 3029, (pp. 361-371). Ottawa: Springer-Verlag Berlin Heidelberg.

[41] Zhang, Q., Sun, J., Tsang, E., & Ford, J. (2006). Estimation of distribution algorithm with 2-opt local search for the quadratic assignment problem. Stud. Fuzziness Soft Comput, 192, 281-292.

[42] Liu, H., Gao, L., & Pan, Q. (2011). A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem. Expert Systems with Applications, 38, 4348-4360.

[43] Fang, C., Kolisch, R., Wang, L., & Mu, C. (2015). An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem. Flex Serv Manuf J 27, 585–605.

[44] Wang, K., Huang, Y., & Qin, H. (2016). A fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling problems under machine breakdown. Journal of the Operational Research Society, 67, 68-82.

[45] Ceberio, J., Irurozki, E., Mendiburu, A., & Lozano, J. (2014). A Distance-Based Ranking Model Estimation of Distribution Algorithm for the Flowshop Scheduling Problem. IEEE Transactions on Evolutionary Computation, 18(2), 286-300.

 10.12928/ijio.v3i1.5862

[46] Pérez-Rodríguez, R., Hernández-Aguirre, A., & Cruz, I. (2017). An estimation of distribution algorithm coupled with the generalized Mallows distribution for a school bus routing problem with bus stop selection. RIAI Revista Iberoamericana de Automática e Informática industrial, 14, 288-298.

[47] Pérez-Rodríguez, R., & Hernández-Aguirre, A. (2018). A hybrid estimation of distribution algorithm for flexible job-shop scheduling problems with process plan flexibility. Applied Intelligence. 48, 3707–3734.

[48] Pérez-Rodríguez, R., & Hernández-Aguirre, A. (2019). A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows. Computers & Industrial Engineering, 130, 75-96.

[49] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems, 89, 228-249.

[50] Mallows, C. (1957). Nonnull ranking models. Biometrika, 44(1-2), 114-130.

[51] Fligner, M., & Verducci, J. (1986). Distance based ranking models. Journal of the Royal Statistical Society, 48(3), 359–369.

[52] Fligner, M., & Verducci, J. (1988). Multistage ranking models. Journal of the American Statistical Association, 83(403), 892–901.

[53] Yan, Y., & Wang, G. (2007). A job shop scheduling approach based on simulation optimization. Proceedings of the 2007 I.E. IEEM

[54] Toth, P., & Vigo, D. (2001). The vehicle routing problem. Monographs on discrete mathematics, and applications. Philadelphia: SIAM.

[55] Sun, D., Tang, L., & Baldacci, R., (2019). A Benders decomposition based framework for solving quay crane scheduling problems. European Journal of Operational Research, 273(2), 504–515.

[56] Borda, J. (1784). Memoire sur les elections au scrutin. Histoire de l'Academie Royale des Science.

[57] Meilă, M., Phadnis, K., Patterson, A., & Bilmes, J. (2007). Consensus ranking under the exponential model. UAI 2007, pp. 285-294.

[58] Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. Management Science, 34(3), 391–401.

[59] Fisher, H., & Thompson, G. (1963). Probabilistic learning combinations of local job-shop scheduling rules. In J. Muth, & G. Thompson (Ed.), Industrial Scheduling (pp. 225–251). Prentice Hall, Englewood Cliffs.

[60] Lawrence, S. (1984). Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement). Graduate School of Industrial Administration. Carnegie-Mellon University, Pittsburgh.

[61] Applegate, D., & Cook, W. (1991). A computational study of the job-shop scheduling problem. ORSA Journal on Computing, 3(2), 149–156.

[62] Storer, R., Wu S, S., & Vaccari , R. (1992). New search spaces for sequencing problems with application to job shop scheduling. Manag Sci, 38(10), 1495–1509.

[63] Yamada, T., & Nakano, R. (1992). A genetic algorithm applicable to large-scale job-shop problems. PPSN, 2, pp. 281–290.

[64] Homberger, J., & Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. European Journal of Operational Research, 162(1), 220–238.

[65] Kim, K., & Park, Y. (2004). A crane scheduling method for port container terminals. European Journal of Operational Research, 156(3).

[66] Li, X., Xing, K., Wu, Y., Wang, X., Luo, J. (2017). Total energy consumption optimization via genetic algorithm in flexible manufacturing systems. Computers & Industrial Engineering, 104, 188-200.