Research Article

Tahani Jabbar Khraibet, Bayda Atiya Kalaf*, and Wafaa Mansoor

# A new metaheuristic algorithm for solving multi-objective single-machine scheduling problems

https://doi.org/10.1515/jisys-2024-0373
received August 07, 2024; accepted December 04, 2024

**Abstract:** Multi-objective scheduling problems are inherently complex due to the need to balance competing objectives, such as minimizing the total weighted completion time, reducing the number of delayed jobs, and minimizing the maximum weighted delay. To address these challenges, this article introduces the meerkat clan algorithm (MCA), inspired by the dynamic, cooperative, and adaptive behaviors of meerkats, which enhances the exploration and exploitation of solution spaces. The MCA is further integrated with the traditional branch-and-bound (BAB) method, utilizing it as an upper bound to significantly improve the accuracy and efficiency of the solutions. Comprehensive computational experiments were conducted to evaluate the MCA's performance against state-of-the-art algorithms, including the bald eagle search optimization algorithm (BESOA) and the standalone BAB method. The MCA demonstrated superior scalability and efficiency, effectively solving problems involving up to $n$ = 30,000 jobs, whereas the BESOA was limited to handling instances with $n$ = 1,000 jobs. Additionally, the integration of MCA with the BAB method achieved exceptional precision and efficiency for smaller problem instances, handling up to $n$ = 13 jobs effectively. The results underscore the MCA algorithm's potential as a robust solution for multi-objective scheduling problems, combining speed and accuracy to outperform traditional methods. Moreover, the hybrid approach of integrating MCA with BAB provides a flexible and versatile framework capable of addressing a wide range of scheduling scenarios, from small-scale to large-scale applications. These findings position the MCA as a transformative tool for solving complex scheduling problems in both theoretical and practical domains.

**Keywords:** single machine scheduling problem, multi-objective, Meerkat Clan Algorithm, Bald Eagle Search, BAB method

## 1 Introduction

Scheduling is a critical decision-making instrument in numerous applications such as industry, engineering, and commerce; their goal is to reduce costs or improve profitability, performance, and efficiency [1]. For the manufacturing and service sectors, it is used to reduce or enhance the costs and benefits of industrial production so that companies are competitive [2]. Furthermore, scheduling entails assigning machines to jobs to complete all tasks within the specified limits. Finding the most efficient sequence for executing these

*** Corresponding author: Bayda Atiya Kalaf**, Department of Mathematics, College of Education for Pure Science, Ibn Al – Haitham, University of Baghdad, Baghdad, 10071, Iraq, e-mail: baydaa.a.k@ihcoedu.uobaghdad.edu.iq
**Tahani Jabbar Khraibet:** Department of Mathematics, College of Education for Pure Science, Ibn Al – Haitham, University of Baghdad, Baghdad, 10071, Iraq; Department of Mathematics, Thi-Qar Directorates of Education, Ministry of Education, Thi-Qar, 64001, Iraq, e-mail: Tahany.Jabbar2103p@ihcoedu.uobaghdad.edu.iq
**Wafaa Mansoor:** Department of Mathematics and Statistics, Murdoch University, Perth, 6150, Australia, e-mail: Wafaa.mansoor2@murdoch.efu.au

jobs on each machine is necessary to minimize the objective function. The primary objective is to optimize allocating one or more resources to activities across a period. Single-machine scheduling is a well-researched scheduling variant because it can divide complex machine environments into smaller subproblems [3].

Multi-objective single-machine scheduling problems are crucial because they reflect real-world scenarios where decision-makers must balance various performance metrics to achieve optimal outcomes [4]. In addition, they extend the classic problems by incorporating multiple, often conflicting, objectives. This helps improve operational efficiency and meet diverse stakeholder requirements, making it a vital area of study [5,6]. Most studies on scheduling theory assume that order processing takes the same amount of time over the entire planning horizon. While traditional optimization methods can be powerful tools for solving well-defined scheduling problems, their application to multi-objective single-machine scheduling problems often faces significant challenges due to the complexity of handling multiple objectives, computational and scalability issues, and the need for robust and diverse solutions. These challenges usually necessitate specialized multi-objective optimization techniques, metaheuristic algorithms, or hybrid approaches to effectively address the complexities of multi-objective scheduling problems [7,8]. Although metaheuristic algorithms have successfully solved challenging real-world MSP problems, a universally applicable algorithm cannot solve every problem involving the no-free-lunch theorem [9].

The meerkat clan algorithm (MCA) employs several specific strategies to be effective. These strategies are designed to find near-optimal solutions by leveraging the social behavior of meerkats and balancing exploration and exploitation [10,11]. Hence, the MCA was used to solve the new multi-objective mathematical model for single-machine scheduling problems. Therefore, this article made several contributions that can be summarized as follows:

- Proposed multiple-objective model for minimizing the total weighted discounted completion time $\sum_{h=1}^{m} w_h(1 - e^{-\alpha C_h})$, the number of tardy jobs $\sum_{h=1}^{m} U_h$, and the maximum weighted tardiness $T_{\max}^{w}$. for a single machine scheduling problem.
- For the first time, the MCA was introduced to solve the multi-objective model for the single-machine scheduling problem. The results were compared with the bald eagle search because it is a practical method for solving non-deterministic polynomial (NP)-hard and complex problems. bald eagle search optimization algorithm (BESOA) is a practical method for solving NP-hard and complex problems [12,13].
- Enhanced the branch-and-bound (BAB) method by utilizing the MCA to improve the upper-bound (UP) estimation. This modification allows for more effective search space pruning, leading to a more efficient solution.

The remainder of this article is organized as follows: Section 2 provides the related works of the proposed study. Section 3 presents the methodology of the mathematical formulation of the problem. Sections 4 and 5 present metaheuristic algorithms (MCA and BESOA). Section 6 provides the exact method (branch and bound). Computational study and results are presented in Section 7. Finally, conclusions and suggestions for future work are given in Section 8.

## 2 Literature review

Most previous studies on scheduling have focused on a single method of measuring performance [14]. Several researchers have extensively studied and documented as follows.

Al-Zuwaini and Janam studied the problem of $1/r_j/\sum_{j=1}^{n} F_j + E_{\max} + \sum_{j=1}^{n} U_j$ using the BAB method and some dominance rules to solve the problem [15]. Al-Zuwanini and Khraibet considered the BAB method to solve the problem of $1//\sum_{j=1}^{n} w_j(1 - e^{-rc_j}) + L_{\max}^{h}$ [16]. Abbas used the BAB method to solve the problem $1//\sum_{j=1}^{n}(U_j + C_j + T_j + T_{\max})$ for $n \le 20$ [17]. Ibrahim et al. suggested $1//\sum_{j=1}^{n}(C_j + T_j + E_j + V_j + U_j)$ problem and then used the BAB method to acquire the most efficient solution for this problem for $n \le 17$ [18]. Ahmed and Ali used the BAB method to solve the problem $(1//\sum C_j + R_L + T_{\max})$ [19]. Neamah et al. studied two problems $1//\sum_{j=1}^{n}(C_j, E_j, T_{\max})$ and $1//\sum_{j=1}^{n}(C_j + E_j) + T_{\max})$ and then solved these problems by the BAB method up to $n$ = 19 to $n$ = 50 jobs, respectively, in a standard time frame [20]. Several researchers have extensively studied and documented [21–26].

Single machine scheduling problems have become increasingly sophisticated and NP-hard during the last few decades. Therefore, metaheuristic algorithms are proposed to obtain optimal or near-optimal solutions for the problem under consideration. Zlobinsky and Cheng used simulated annealing to solve the problem of minimizing weighted earliness and tardiness [27]. Jiaji et al. considered particle swarm optimization and Tabu search to minimize the weighted number of tardiest [28]. Ali and Ahmed used the Bee algorithm and particle swarm optimization to solve the problem $\left(1//\sum C_j + R_L + T_{\max}\right)$ [29]. Moharam et al. introduced a chimp optimization algorithm to minimize the tardy/lost (TL) penalties [30]. Antonio and Fernandez studied a single-machine scheduling problem, where the objective is minimum total tardiness using the Harmony search [31]. Wu and Zheng used the Tabu search to find a near-optimum solution for minimizing the makespan [32]. Moreover, we can see that some researchers are interested in metaheuristic algorithms [6,33–35].

# 3 Methodology

## 3.1 The mathematical model

In this section, the multi-objective model of MSP, with $m$ jobs on a single machine, was examined that is consistently accessible and capable of performing them.

### 3.1.1 Notations

Some notations utilized in the formulation of the multi-objective model for MSP were presented as follows:

- $N = \{1, 2, 3, \ldots, m\}$;
- $\sigma$ = set of all schedules;
- $p_h$ = processing time for job $h$, where $h$ = 2, 3, …, $n$;
- $w_h$ = weight of job $h$;
- $d_h$ = due date of job $h$;
- $C_h$ = completion time of job $h$;
- $\alpha$ = discount rate;
- $T_{\max}$ = maximum tardiness;
- $T^w_{\max}$ = maximum weighted tardine;
- MOF = multiple objective's function;
- BAB = branch and bounded;
- MCA = Meerkat clan algorithm;
- MSP = machine scheduling problem;
- UB = upper bound;
- LB = lower bound;
- WDSPT = weighted discounted shortest processing time;
- MA = Moor algorithm;
- LA = Lawler algorithm;
- EDD = earliest due date;
- BESOA = bald eagle search optimization algorithm.

### 3.1.2 Objective function and constraint

In the present study, three objective functions (minimize the total weighted discounted completion time $\sum_{h=1}^{m} w_h(1 - e^{-\alpha C_h})$, the number of tardy jobs $\sum_{h=1}^{m} U_h$, and the maximum weighted tardiness $T_{\max}^{w}$ $(1//\sum_{h=1}^{m}(w_h(1 - e^{-\alpha C_h}) + U_h) + T_{\max}^{w}))$ were solved at the same time. The main problem is shown by TA, and it can be in the following way:

$$\mathrm{TA} = \min_{\sigma \in S}\{M_{(\sigma)}\} = \min_{\sigma \in S}\left\{\sum_{h=1}^{m}(w_h(1 - e^{-\alpha c_h}) + U_h) + T_{\max}^{w}\right\}, \tag{1}$$

s.t.

$$\left.\begin{array}{c} C_{(h)} \geq p_{\sigma(h)} \\ C_{(h)} = C_{h-1} + p_{\sigma(h)} \\ O < \alpha < 1 \\ U_{(h)} = \begin{cases} 1 \text{ if } C_{(h)} > d_{\sigma(h)} \\ 0 \text{ otherwise} \end{cases} \\ L_{(h)} = C_{(h)} - d_{(h)} \\ T_{(h)} \geq \max\{C_{(h)} - d_{(h)}, 0\} \\ w_{\sigma(h)} > 0,\ d_{\sigma(h)} > 0, \quad E_{\sigma(h)} \geq 0, \quad T_{(h)} \geq 0 \quad p_{\sigma(h)} > 0 \end{array}\right\} h \text{ from } 1 \text{ to } m\Bigg\}. \tag{2}$$

## 4 Meerkat clan algorithm

The MCA is a metaheuristic algorithm that draws inspiration from the food-finding behavior of meerkats in the desert that was proposed by Sadiq et al. [36]. This algorithm uses efficient methods to address the optimization problem and obtain the optimal solution. Meerkats are social animals that live in groups of several; each group has an area that inhabits states with 5–30 individuals. As amicable creatures, they collaborate on lavatory duties and parental supervision. Every group has a dominant alpha female and a commanding alpha male who exerts a strong influence. Every crowd has a territory; they occasionally relocate when food is scarce or when a more dominant group displaces them. It is based on three fundamental components of social behavior. The initial social behavior of the meerkat clan involves selecting one or more individuals to act as guards or observers while others are hunting or playing to alert them in the event of any dangerous situations [37]. In addition, the meerkat follows a different route every day and leaves the area they visit for at least a week to give the area a chance to replenish its food supply. The meerkat must balance care and hunting better [38,39].

In multi-objective single-machine scheduling problems, the objectives of minimizing total weighted discounted completion time, minimizing the number of tardy jobs, and minimizing maximum weighted tardiness often interact in complex ways. The MCA effectively balances these objectives. MCA operates within a multi-objective optimization framework aiming to approximate the Pareto front. The Pareto front represents a set of non-dominated solutions where no objective can be improved without worsening another. MCA searches for diverse solutions along this front, allowing decision-makers to choose the best trade-offs based on their preferences. In addition, MCA divides the solution space into different clans, each focusing on various regions and aspects of the problem. This diversity allows the algorithm to explore different trade-offs through the objective functions. After that, each clan is guided by a leader who helps refine the solutions within that clan. This process helps balance the trade-offs between objectives by focusing on different aspects of the solution space. MCA evaluates solutions based on a combination of objectives. The fitness of a solution is determined by how well it balances the trade-offs between the objective functions. Then, MCA employs mechanisms to maintain diversity among the solutions in the population. This prevents the algorithm from converging too quickly on a suboptimal region of the solution space and ensures that various trade-offs are explored. This helps in finding solutions that offer a good balance between the different objectives.

The following steps of this study show how the MCA solves the single-machine scheduling problems:

**Step 1**: Problem Definition: The jobs with processing times, deadlines, etc.

**Step 2**: Generate an initial population of meerkats (solutions).

**Step 3**: Calculate each solution's objective functions (fitness evaluation) (e.g., completion time, lateness).

**Step 4**: Meerkats update their positions (schedules) based on the best solutions they find by the Exploration and Exploitation

**Step 5**: Stopping Criterion: The algorithm stops after a predefined number of iterations or if the solutions stabilize.

## 5 Bald eagle search optimization algorithm

In 2020, the BESOA was introduced as a novel metaheuristic optimization method by Alsattar et al. [40]. It draws inspiration from the hunting behavior exhibited by the bald eagle. Many researchers studied and written about the BESOA algorithm including [41–46].

The bald eagle inhabits expansive, open spaces with abundant prey and ancient trees for nesting. Their keen vision allows them to locate prey from considerable distances, and they are capable of seeing in both directions at once. This condor's prey–prey behavior is simulated by the BESOA algorithm, which divides the process into three stages: selection, search, and soaring. During the selecting phase, the eagle selects the area containing the most significant number of preys. During the searching phase, it searches for prey within the designated area.

The following bald eagle search algorithm to solve a mathematical model for a single machine scheduling problem is as follows:

**Step 1:** Randomly generate the first population.

**Step 2:** Determine every node's fitness function ($\min_{\sigma \in S}\{\sum_{h=1}^{m}(w_h(1 - e^{-\alpha c_h}) + U_h) + T_{\max}^{w}\}$) for all the initialized populations and select the best cluster head.

**Step 3:** Set the iteration number $t = 0$.

**Step 4:** Set $t_{\max}$ as the maximum iteration

**Step 5:** Check the termination criteria $< t_{\max}$; if it is greater, then replace the initial position with the updated position else generate the random numbers for determining the best values.

**Step 6:** Gather every node from the initial population and follow three steps to find out the best cluster head node.

**Step 7**: At first, as per equation ($Ȩ_{\text{new},i} = Ȩ_{\text{best}} + \alpha \times r(Ȩ_{\text{mean}} - Ȩ_i)$), in which $\alpha$ is a position control parameter with a value between 5 and 10, and $r$ is a random number between 0.5 and 2, the bald eagle selects the region to hunt for the best solution.

Further, evaluate the fitness function for both the new position and the best position for acquiring the best search space.

**Step 8**: The bald eagle can choose the hunting area depending on the search space using an equation:

$$Ȩ_{i,\text{new}} = Ȩ_i + y(i) \times (Ȩ_i - Ȩ_{i+1}) + x(i) \times (Ȩ_i - Ȩ_{\text{mean}})$$

where $x(i) = \frac{xr(i)}{\max|xr|}$ and $y(i) = \frac{yr(i)}{\max|yr|}$

$xr(i) = r(i) \times \sin(\theta(i))$, $yr(i) = r(i) \times \cos(\theta(i))$

$\theta(i) = \alpha \times \pi \times \text{rand}$ and $r(i) = \theta(i) + R \times \text{rand}$, where $C_1, C_2 \in \{0.1\}$.

Like search space, it also finds the fitness and acquires the best position for select space.

**Step 9**: In this swooping stage, bald eagle deploying search space's new position for swooping the prey towards.

**Step 10:** Repeat the above step until it attains the termination criteria.

# 6 BAB method

The BAB method is developed with the forward sequence branching rule. If the jobs are strung together at the first $k$ places in the search tree, the nodes at level $k$ are representative of the initial partial order. The derived lower bound (LB) determines the cost of the unscheduled orders, and the objective function determines the cost of scheduling the orders at a particular node. The BAB technique is dominant if the node has LB ≥ UB at each level. The backtracking method is then used to repeat the process until all nodes have been considered. Backtracking is the step in the BAB method that leads from the lowest to the highest level. Some researchers have worked out a method BAB [47–49].

The BAB method's efficiency largely hinges on the effectiveness of the bounding strategies. Calculating UB and LB accurately prunes solution space, leading to faster and more efficient optimal solutions in various combinatorial and optimization problems.

## 6.1 Upper bound

This subsection introduces the three UBs, we will be chosen the best one as follows:

$$\mathrm{UB} = \min\{\mathrm{UB}_1, \mathrm{UB}_2, \mathrm{UB}_3\}$$

(1) $\mathrm{UB}_1$: Where the $n$ jobs are ordered in the (WDSPT) rule, that is sequencing the jobs in non-decreasing order of

$$\frac{w_1 e^{-ap_1}}{1 - e^{-ap_1}} \le \frac{w_2 e^{-ap_2}}{1 - e^{-ap_2}} \le \ldots \le \frac{w_m e^{-ap_m}}{1 - e^{-ap_m}}$$

(2) $\mathrm{UB}_2$: Where the $n$ jobs are ordered in the (EDD) rule, that is sequencing the jobs in increasing order of due dates $d_1 \le d_2 \le \ldots \le d_m$ and then the cost is calculated.

(3) $\mathrm{UB}_3$: This UB was determined by employing the MCA. MCA provides an estimate of the best possible solution that can be achieved, which can be used to guide and improve the BAB process. In addition, MCA's adaptive search mechanisms can generate high-quality UBs by exploring different regions of the solution space and using cooperative strategies. Also, MCA can dynamically update the UB based on its ongoing search, providing more accurate and relevant UBs during the BAB process. Hence, the BAB method can more effectively prune branches of the search tree. MCA's UBs help the BAB method to focus on more promising regions of the solution space. By providing a good estimate of the best possible solution, MCA guides the search more efficiently, leading to faster convergence to the optimal or near-optimal solution.

## 6.2 Lower bound

The LB is one of the most important constraints in determining a satisfactory solution to a problem. Obtaining LB for an NP-hard multi-objective problem is clearly difficult. A decomposition of the problem (TA) will be utilized here. In order to determine a LB for problem TA, we divided the problem into three subproblems (3), (4), and (5), as illustrated in:

$$\left.\begin{array}{c} N_1 = \min_{\sigma(h)} = \left\{\sum_{h=1}^{m} w_h (1 - e^{-\alpha c_{\sigma(h)}})\right\} \\ s.\ t \\ C_{\sigma(h)} \ge p_{\sigma(h)} \\ C_{\sigma(h)} = C_{\sigma(h-1)} + p_{\sigma(h)} \\ 0 < \alpha < 1 \\ w_{\sigma(h)} \ge 1,\ p_{\sigma(h)} > 0,\ d_{\sigma(h)} > 0 \end{array}\right\} h \text{ from } 1 \text{ to } m\Bigg\}, \tag{3}$$

$$\left.\begin{array}{c} N_2 = \min_{\sigma(h)} = \left\{\sum_{h=1}^{m} U_{\sigma(h)}\right\} \\ s.\ t \\ C_{\sigma(h)} \geq p_{\sigma(h)} \\ C_{\sigma(h)} = C_{\sigma(h-1)} + p_{\sigma(h)} \\ U_{(h)} = \begin{cases} 1 & if C_{(h)} > d_{\sigma(h)} \\ 0 & \text{otherwise} \end{cases} \\ p_{\sigma(h)} > 0, d_{\sigma(h)} > 0 \end{array}\right\} h \text{ from 1 to } m \Bigg\}, \quad (4)$$

$$\left.\begin{array}{c} N_3 = \min_{\sigma(h)} = T^{w}_{\max} \\ s.\ t \\ C_{\sigma(h)} \geq p_{\sigma(h)} \\ C_{\sigma(h)} = C_{\sigma(h-1)} + p_{\sigma(h)} \\ L_{(h)} = C_{(h)} - d_{(h)} \\ T_{(h)} \geq \max\{C_{(h)} - d_{(h)}, 0\} \\ w_{\sigma(h)} \geq 1, p_{\sigma(h)} > 0, d_{\sigma(h)} > 0, T_{\sigma(h)} \geq 0 \end{array}\right\} h \text{ from 1 to } m \Bigg\}. \quad (5)$$

For subproblem (3), we used the WDSPT rule $\frac{w_h e^{-rp_h}}{1 - e^{-rp_h}}$ [3] to solve $1//\sum_{h=1}^{m} w_h(1 - e^{-\alpha c_h})$ problem, for subproblem (4), the Moor algorithm (MA) [50] was used to solve $1//\sum_{h=1}^{m} U_h$ problem, and for the subproblem (5) the Lawler algorithm (LA) [51] was used to solve $1//T^{w}_{\max}$ problem.

**The following steps of algorithm**

**Step 1:** Enter: $n$, $p_h$, $d_h$, and $w_h$ where $h$ from 1 to $m$.

**Step 2:** Order the jobs by using the (WDSPT) rule.

**Step 3:** Calculate the value for each job $h$ that schedules the jobs in non-decreasing order of ratio: $\frac{w_h e^{-ap_h}}{1 - e^{-ap_h}}$ and compute $\sum_{h=1}^{m} w_h(1 - e^{-\alpha c_h})(\text{WDSPT}) = \sum_{h=1}^{m} U_h(\text{WDSPT}) = T^{w}_{\max} = (\text{WDSPT})$ the WDSPT gives optimal solution.

**Step 4:** We obtain that $\text{LB}_1 = \text{WDSPT}$.

**Step 5:** Order the jobs using the (EDD)-rule.

**Step 6:** Calculate the value for each job $h$ that schedules the jobs in non-decreasing order of of due dates $d_h$ and compute $(\sum_{h=1}^{m} w_h(1 - e^{-\alpha c_h})(\text{MA})) = \sum_{h=1}^{m} U_h(\text{MA}) = T^{w}_{\max} = (\text{MA})$ the MA gives optimal solution.

**Step 7:** We obtain that $\text{LB}_2 = \text{MA}$.

**Step 8:** Let $h^*$ such that $f_h{}^*(\sum_{h \in M} p_h) = \min_{h \in f} f_h(\sum_{h \in M} p_h)$.

**Step 9:** Set $M = M - \{h^*\}$ and sequence job $h^*$ in $\sigma$, i.e., $\sigma = (h^*, \sigma)$. Modify $F$ to represent the new set of schedulable jobs.

**Step 10:** If $M = \varphi$ STOP; otherwise, go to step (9).

**Step 11:** Compute $(\sum_{h=1}^{m} w_h(1 - e^{-\alpha c_h})(\text{LA})) = \sum_{h=1}^{m} U_h(\text{LA}) = T^{w}_{\max} = (\text{LA})$ the LA gives optimal solution.

**Step 12:** We obtain that $\text{LB}_3 = \text{LA}$.

**Step 13:** $\text{LB} = \text{LB}_1 + \text{LB}_2 + \text{LB}_3$.

# 7 Computational results

To verify and evaluate the performance of the MCA for solving the multi-objective model based on a single-machine scheduling problem, we used a variety of problems with medium and big size limits and significantly equal constraint sizes from 3 to 30,000 jobs. The results also compare the performance of MCA with two other methods: BESOA and BAB method. The processing time is uniformly distributed across in $U$[1,10], and weights were generated from the set {1,2,…,10}, which is now a standard method for creating single-machine scheduling problems with due dates. The due dates are uniformly distributed within the range [$P$(1 − TF-RDD/2), $P$(1

− TF + RDD/2)], where $P = \sum_{j=1}^{n} p_j$, which is influenced by the relative range of due date (RDD) and the average tardiness factor (TF). The TF value is extracted from the set of values 0.1, 0.2, 0.3, 0.4, and 0.5, while the RDD value is obtained from the set of values 0.8, 1.0, 1.2, 1.4, 1.6, and 1.8. According to an analysis of the algorithms' performances, the MATLAB programming language encoded and resolved these comparable examples. As the stopping criterion, each algorithm, including MCA, BESOA, and BAB, was executed for 1,000 iterations. Table 1 shows all parameters set for the MCA and BESOA.

**Table 1:** Parameters (MCA and BESOA)

| MCA | BESOA |
|---|---|
| $m = n = 6$ | $\alpha \in [5,10]$ |
| $c = 2$ | $r \in [0.5,2]$. |
| $C = n - m - 1$ | |
| $F_r = 0.3$ | |
| $C_r = 0.25$ | |
| $k = 2$ | |

For the performance on small jobs (3–18 jobs), the algorithms show nearly identical results when dealing with a few jobs. Table 2 shows the results of the BAB method for the problem with different values of $n$ ($n$ = 3–13), the optimal value, the UB, the initial LB, the computing time in seconds (time) that the BAB is stopped after a fixed period of time, here after 1,800 s (i.e., after 30 min). In Table 3, for instance, the completion times for MCA, BESOA, and BAB are virtually identical for three orders and are each reporting approximately 21.60 units. As the job size increases to 12 jobs, MCA significantly outperforms BAB, MCA's average completion time is 92.41 units, BAB average completion time: 123.56 units. BESOA's performance remains very close to MCA's at 92.54 units, indicating that BESOA and MCA are better suited for even small job sizes than BAB. For job sizes between 500 and 1,000, MCA shows an advantage in both performance and scalability compared to BAB. For 500 jobs, MCA records an average completion time of 4433.84 units, while BESOA slightly edges it out with 4402.76 units. BAB does not have available results at this scale, which could indicate its limitations with larger datasets. For 750 jobs, MCA yields 6662.68 units, while BESOA performs slightly better at 6626.59 units. Again, BAB's results are missing in this range. At 1,000 jobs, MCA clocks 9121.45 units, closely followed by BESOA at 9059.63 units, showing that both algorithms can effectively handle medium-scale job sizes. In addition, for the Jobs (20,000–30,000 jobs). When tackling 20,000 jobs, MCA achieved 183,071.57 units in completion time. This demonstrates that MCA can handle large-scale scheduling problems without significant performance degradation. Unfortunately, BESOA and BAB do not produce results in this range, implying potential limitations in

**Table 2:** Comparison results BAB with ($n = 3 - 13$)

| $n$ | Av. of UB | Av. of LB | Av. of BAB | Av. of time |
|---|---|---|---|---|
| 3 | 21.60018091 | 14.69335723 | 21.60018073 | Ver |
| 4 | 21.23403563 | 21.28426712 | 21.23403563 | Ver |
| 5 | 44.46860132 | 29.53891478 | 43.13134148 | Ver |
| 6 | 52.39646015 | 30.91867809 | 50.77212583 | Ver |
| 7 | 62.91662674 | 34.16032028 | 58.15483137 | Ver |
| 8 | 64.60277214 | 40.26112022 | 59.33084972 | Ver |
| 9 | 86.30803146 | 51.64899101 | 78.73443725 | Ver |
| 10 | 106.6516991 | 57.84292984 | 98.73360194 | 3.289942 |
| 11 | 109.1562874 | 56.81628265 | 96.84323284 | 13.35572437 |
| 12 | 104.9581734 | 55.37373085 | 92.414069 | 123.5631877 |
| 13 | 117.8423424 | 68.08335323 | 103.1497112 | 790.4244712 |

**Table 3:** Comparison results between MCA and BESOA with ($n$ = 3 – 30, 000)

| *n* | MCA | | BESOA | | Best value |
|---|---|---|---|---|---|
| | Av. of MCA | Av. of time | Av. of BES | Av. of time | |
| 3 | 21.60018091 | Ver | 21.60018091 | Ver | 21.60018091 |
| 4 | 21.23403575 | Ver | 21.23403575 | Ver | 21.23403575 |
| 5 | 43.13134127 | Ver | 43.13134127 | Ver | 43.13134127 |
| 6 | 50.77212601 | Ver | 50.77212601 | Ver | 50.77212601 |
| 7 | 58.15483208 | Ver | 58.15483208 | Ver | 58.15483208 |
| 8 | 59.33085098 | Ver | 59.33085098 | Ver | 59.33085098 |
| 9 | 78.7344368 | Ver | 78.82521896 | Ver | 78.7344368 |
| 10 | 98.73360138 | Ver | 98.73546371 | Ver | 98.73360138 |
| 11 | 96.8432312 | Ver | 97.24269104 | Ver | 96.8432312 |
| 12 | 92.41406898 | Ver | 92.53869667 | Ver | 92.41406898 |
| 13 | 103.2228207 | Ver | 103.3695221 | Ver | 103.0840004 |
| 14 | 113.1100166 | Ver | 113.3693558 | Ver | 113.1100166 |
| 15 | 105.9779259 | Ver | 106.1474007 | Ver | 105.9779259 |
| 16 | 127.433773 | Ver | 127.433773 | Ver | 127.433773 |
| 17 | 138.5919113 | Ver | 140.1413879 | Ver | 138.5886749 |
| 18 | 142.9502785 | Ver | 144.1621925 | Ver | 142.9502785 |
| 19 | 145.7566879 | Ver | 147.6488274 | Ver | 145.7566879 |
| 20 | 177.9658623 | Ver | 178.7056702 | Ver | 177.9626106 |
| 25 | 216.0834427 | Ver | 218.0747879 | Ver | 215.7490799 |
| 30 | 241.8208191 | Ver | 242.9052719 | Ver | 241.4472733 |
| 35 | 286.6710388 | Ver | 290.0697525 | Ver | 286.16922 |
| 40 | 310.0009354 | Ver | 311.6062653 | Ver | 309.4566055 |
| 45 | 371.494136 | Ver | 373.6911591 | Ver | 370.5327942 |
| 50 | 368.2818085 | Ver | 365.8695709 | Ver | 363.8038086 |
| 100 | 886.5877319 | Ver | 887.0128784 | Ver | 883.5224304 |
| 150 | 1269.941943 | Ver | 1258.826587 | Ver | 1258.070142 |
| 200 | 1800.366394 | Ver | 1791.468628 | 1.8059691 | 1790.628101 |
| 250 | 2186.928687 | Ver | 2174.076563 | 2.5731 | 2173.921033 |
| 500 | 4433.843042 | 1.3063921 | 4402.761548 | 5.1519146 | 4400.566048 |
| 750 | 6662.680957 | 1.985482 | 6626.586523 | 7.6567608 | 6618.163184 |
| 1,000 | 9121.448633 | 2.613544 | 9059.625684 | 10.000 | 9059.625684 |
| 1,250 | 11465.10869 | 3.2811735 | — | — | — |
| 1,500 | 13631.59043 | 4.0678417 | — | — | — |
| 1,750 | 15755.91357 | 4.7154018 | — | — | — |
| 2,000 | 18242.31357 | 5.4730787 | — | — | — |
| 2,500 | 22725.31875 | 7.2085663 | — | — | — |
| 5,000 | 45462.90313 | 16.1576449 | — | — | — |
| 10,000 | 91449.23672 | 33.515658 | — | — | — |
| 15,000 | 137490.7961 | 49.1566127 | — | — | — |
| 20,000 | 183071.5672 | 54.2131458 | — | — | — |
| 30,000 | 273323.2672 | 95.0794322 | — | — | — |

these methods for extremely large datasets. At the upper end, with 30,000 jobs, MCA reports a completion time of 273,323.27 units, further confirming its scalability. We can conclude that MCA shows robust performance across all job sizes. Its ability to handle job counts up to 30,000 without any significant performance loss is a testament to its efficiency and scalability. Tuning parameters such as population size, mutation rate, and crossover rate ensure an optimal balance between exploration and exploitation. Although it is competitive with BESOA, MCA's performance is marginally lower than BESOA for medium job sizes (500–1,000 jobs).

# 8 Conclusion

In this study, a new multi-objective model for a single-machine scheduling problem was used MCA as a novel metaheuristic approach for solving the model. Additionally, we enhanced the BAB method by incorporating MCA to improve its UB calculations. Our results demonstrate that this integration significantly improves the performance of the BAB method, achieving outcomes in up to 13 test cases. Based on the results, The MCA offers advantages over BESOA and BAB regarding accuracy and computational efficiency for multi-objective single-machine scheduling problems. MCA excels with its diverse solution representation, adaptive search strategies, and ability to handle multi-objective optimization effectively and solve up to 30,000 jobs. While, the second rank was the BESOA delivered and solved up to 1,000. However, MCA does have limitations related to parameter tuning, scalability, convergence speed, and resource consumption. Future research can address these limitations by developing automated tuning methods, scalable algorithms, dynamic adaptations, and resource-efficient implementations, enhancing MCA's effectiveness and applicability in real-world scenarios. In addition, some suggestions for further studies based on multi-objective model for a single machine scheduling problem.

**Acknowledgements:** The authors would like to express their gratitude and appreciation to the referees and editors for their constructive suggestions and valuable remarks, which have helped improve this article.

**Funding information:** The authors received no specific funding for this study.

**Author contributions:** Tahani Jabbar Khraibet: conceptualization, methodology, formal analysis, investigation, writing – original draft, visualization, writing – review & editing. Bayda Atiya Kalaf, Wafaa Mansoor: conceptualization, methodology, writing – review & editing, project administration, supervision.

**Conflict of interest:** The authors declare no conflict of interest.

**Data availability statement:** The data used in this study were specifically created for simulation purposes and not derived from real-world sources.

# References

[1] Baker KR, Trietsch D. Principles of sequencing and scheduling, 2nd edn. Set in 10/12pt Warnock by SPi Global, Pondicherry, India Printed: © 2019. John Wiley & Sons, Inc. Edition, 2009. doi: 10.1002/9781119262602.
[2] Stevenson WJ. Operations management. Fourteenth. New York-United States of America: McGraw-Hill Education; 2021.
[3] Pinedo ML. Scheduling: Theory, algorithms, and systems. Nova Iorque: Springer-Verla; 2016. doi: 10.1007/978-3-319-26580-3.
[4] Janis SN, Sven S, Udo B. A systematic review of multi-objective hybrid flow shop scheduling. Eur J Oper Res. 2023;309:1–23. doi: 10.1016/j.ejor.2022.08.009.
[5] Chachan HA, Hameed AS. Exact methods for solving multi-objective problem on single machine scheduling. Iraqi J Sci. 2019;60:1802–13. doi: 10.24996/ijs.2019.60.8.17.
[6] Ezugwu AS. Metaheuristic optimization for sustainable unrelated parallel machine scheduling: A concise overview with a proof-of-concept study. IEEE Access. 2024;12:3386–416. doi: 10.1109/ACCESS.20233347047.
[7] Noor AR, Yossra HA, Tarik AR. Unraveling the veling the versatility and impact of multi-objectiv ersatility and impact of multi-objective optimization: Algorithms, applications, and trends for solving complex real-world problems. J Soft Comput Computer Appl. 2024;1:1–18. doi: 10.70403/3008-1084.1008.
[8] Yin J, Huang R, Sun H, Ca S. Multi-objective optimization for coordinated production and transportation in prefabricated construction with on-site lifting requirements. 2024;189:110017. doi: 10.1016/j.cie.2024.110017.
[9] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput. 1997;1:67–82. doi: 10.1109/4235.585893.
[10] Srinivasan V, Boopathi CS, Sridhar R. A new meerkat optimization algorithm based maximum power point tracking for partially shaded photovoltaic system. Ain Shams Eng J. 2021;12:3791–802. doi: 10.1016/j.asej.2021.03.017.

[11] Navrinder K, Sarabjeet K. Optimized data hiding method based on Meerkat Clan Algorithm for image steganography. Int Res J Eng Appl Sci. 2023;11:12–8. doi: 10.55083/irjeas.2022.v11i01002.
[12] El-Shorbagy M, Bouaoudab A, Nabwey HA, Abualigahd L, Fatma A, Hashim FA. Bald eagle Search Algorithm: A comprehensive review with its variants and applications. Syst Sci Control Eng. 2024;12:1–59. doi: 10.1080/21642583.2024.2385310.
[13] Wenyan GW, Hou1 Z, Dai1 F, Wang X, Qiang Q. Bald eagle Search Optimization Algorithm combined with spherical random shrinkage mechanism and its application. J Bionic Eng. 2024;21:572–605. doi: 10.1007/s42235-023-00446-7.
[14] Kindt VT, Croce FD. A Single machine adversarial bilevel scheduling problems. Eur J Oper Res. 2024;315(1):63–72. doi: 10.1016/j.ejor.2023.11.018.
[15] Al-Zuwanini MK, Janam KA. Branch and bound method to solve multi objectives function. J Adv Math. 2014;9:2485–95. doi: 10.24297/jam.v12i3.494.
[16] Al-Zuwanini MK, Khraibet TJ. Branch and bound method to solve multi objectives function. J Adv Math. 2016;12:5964–74. doi: 10.24297/jam.v12i3.494.
[17] Abbass DA. Using branch and bound and local search methods to solve multi-objective machine scheduling problem. 2019 First International Conference of Computer and Applied Sciences (CAS). doi: 10.1109/CAS47993.2019.9075460.
[18] Ibrahim MH, Ali FH, Chachan HA. Solving multi-objective's function problem using branch and bound and local search methods. Int J Nonlinear Anal Appl (IJNAA). 2022;13:1649–58. doi: 10.22075/ijnaa.2022.5780.
[19] Ahmed MG, Ali FH. Exact method with dominance rules for solving scheduling on a single machine problem with multiobjective function. Al-Mustansiriyah J Sci. 2022;33:56–63. doi: 10.23851/mjs.v33i2.1091.
[20] Neamah NM, Atiya Bk, Mansoor W. Solving tri-criteria: Total completion time, total earliness, and maximum tardiness using exact and heuristic methods on single-machine scheduling problems. Math Model Eng Probl. 2024;11:987–95. doi: 10.18280/mmep.110415.
[21] Atiya Bk, Bakheet AJ, Abbas IT, Bakar MR, Soon LL, Monsi MB. Application of simulated annealing to solve multi-objectives for aggregate production planning. AIP Conf Proc. 2016;1739:020086-1–020086-6. doi: 10.1063/1.4952566.
[22] Bakar MR, Abbas IT, Kalal MA, AlSattar HA, Bakhayt AG, Atiya Bk. Solution for multi-objective optimisation master production scheduling problems based on swarm intelligence algorithms. J Comput Theor Nanosci. 2017;14:5184–94. doi: 10.1166/jctn.2017.6729.
[23] Khraibet TJ, Ghafil WK. Using hybrid GA-PSO algorithm to solve problem in machine scheduling. J Discrete Math Sci Cryptogr. 2021;24(7):2027–35. doi: 10.1080/09720529.2021.1958998.
[24] Yousif SF, Ali FH. Solving maximum early jobs time and range of lateness jobs times problem using exact and Heuristic methods. Iraqi J Sci. 2024;65:923–37. doi: 10.24996/ijs.2024.65.2.28.
[25] Abbas IT, Ghayyib MN. Using sensitivity analysis in linear programming with practical physical applications. Iraqi J Sci. 2024;65:907–22. doi: 10.24996/ijs.2024.65.2.27.
[26] Neamah NM, Atiya BK. Solving tri-criteria: Total completion time, total late work, and maximum earliness by using exact, and heuristic methods on single machine scheduling problem. Iraqi J Computer Sci Math. 2024;5:14–25. doi: 10.52866/ijcsm.2024.05.03.002.
[27] Zlobinsky N, Cheng L. A metaheuristic algorithm for single machine. South Afr Inst Electr Eng. 2018;109:58–68. doi: 10.23919/SAIEE.2018.8531800.
[28] Jiaji L, Yuvraj G, Appadoo SS. Algorithms for a two-agent single machine scheduling problem to minimize weighted number of tardy jobs. J Inf Optim Sci. 2021;42(4):785–811. doi: 10.1080/02522667.2020.1816637.
[29] Ali FH, Ahmed MG. Local search methods for solving total completion times, range of lateness and maximum tardiness problem. 6th International Engineering Conference" Sustainable Technology and Development IEC; 2020. doi: 10.1109/IEC49899.2020.9122821.
[30] Moharam R, Ali A, Morsy E, Ahmed MA, Mostafa MS. A discrete chimp optimization algorithm for minimizing tardy/lost penalties on a single machine scheduling problem. IEEE Access. 2022;10:52126–38. doi: 10.1109/ACCESS.2022.3174484.
[31] Antonio CA, Fernandez V. A modified harmony search for the T-single machine scheduling problem with variable and flexible maintenance. Expert Syst Appl. 2022;198:116897. doi: 10.1016/j.eswa.2022.116897.
[32] Wu H, Zheng H. Single-machine scheduling with periodic maintenance and learning effect. Sci Rep. 2023;13:1–16. doi: 10.1038/s41598-023-36056-w.
[33] Stanković A, Petrović G, Ćojbašić Z, Marković Z. An application of metaheuristic optimization algorithms for solving the flexible job-shop scheduling problem. Oper Res Eng Sci: Theory Appl. 2020;3:13–28. doi: 10.31181/oresta20303013s.
[34] Mohammadi G, Moaddabi E. Using two metaheuristic algorithms for scheduling parallel machines with sequence dependent set-up times in job shop industries. Int J Syst Sci. 2021;52:2904–17. doi: 10.1080/00207721.2021.1913534.
[35] Xinyu Li LG, Zhang O, Gao L. Domain knowledge used in meta-heuristic algorithms for the job-shop scheduling problem: Review and analysis. Tsinghua Sci Technol. 2024;29:1368–89. doi: 10.26599/TST.2023.9010140.
[36] Sadiq AT, Abdullah HS, Ahmed ZO. Solving flexible job shop scheduling problem using Meerkat Clan Algorithm. Iraqi J Sci. 2018;59:753–61. doi: 10.24996/ijs.2018.59.2A.13.
[37] Mohamad AS, Ahmed TS. Modified Meerkat Clan Algorithm for association rules mining. Baghdad Sci J. 2024;21(5):1743–53. doi: 10.21123/bsj.2023.8018.
[38] Hussein SA, Yousif AY. An improved Meerkat Clan Algorithm for solving 0-1 knapsack problem. Iraqi J Sci. 2022;63:773–84. doi: 10.24996/ijs.2022.63.2.32.

[39] Saleh MA, Sadiq AT. Modified Meerkat Clan Algorithm for association rules mining. Baghdad Sci J. 2023;21:1743–53. doi: 10.21123/bsj.2023.8018.
[40] Alsattar HA, Zaidan AA, Zaidan BB. Novel meta-heuristic Bald Eagle Search Optimization Algorithm. Artif Intell Rev. 2020;53:2237–64. doi: 10.1007/s10462-019-09732-5.
[41] Huang Y, Jiang H, Wang W, Sun D. Prediction model of soil electrical conductivity based on ELM optimized by Bald Eagle Search Algorithm. Electron J. 2021;25:50–6. doi: 10.53314/ELS2125050H.
[42] Alsaidan I, Shaheen AM, Hasanien HM, Alnafisah AS. A PEMFC model optimization using the enhanced Bald Eagle Algorithm. Ain Shams Eng J. 2022;13:1–11. doi: 10.1016/j.asej.2022.101749.
[43] Wang Y, Zhang W, Sun J, Wang L, Song X, Zhao X. Survival risk prediction of esophageal squamous cell carcinoma based on BES-LSSVM. Comput Intell Neurosci. 2022;1–12. doi: 10.1155/2022/3895590.
[44] Kankılıç S, Karpat E. Optimization of multilayer absorbers using the Bald Eagle Optimization Algorithm. Appl Sci. 2023;13:1–17. doi: 10.3390/app131810301.
[45] Zaky AA, Ghoniem RM, Selim F. Precise modeling of proton exchange membrane fuel cell using the modified Bald Eagle Optimization Algorithm. Sustainability. 2023;15:1–16. doi: 10.3390/su151310590.
[46] Youssef H, Kamel S, Hassan MH, Nasrat L, Jurado F. An improved Bald Eagle Search Optimization Algorithm for optimal home energy management systems. Soft Comput. 2024;8:1367–90. doi: 10.1007/s00500-023-08328-0.
[47] Abdul-Razaq TS. Machine scheduling problems: A branch and bound approach. (Ph.D. Thesis). Keele University; 1987.
[48] Forget N, Gadegaard SL, Nielsen LR. Warm-starting lower bound set computations for branch-and-bound algorithms for multi objective integer linear programs. Eur J Oper Res. 2022;302(3):909–24. doi: 10.1016/j.ejor.2022.01.047.
[49] Bożejko W, Pempera J, Uchroński M, Wodecki M. Quantum annealing-driven branch and bound for the single machine total weighted number of tardy jobs scheduling problem. Future Gener Computer Syst. 2024;155:245–55. doi: 10.1016/j.future.2024.02.016.
[50] Moore JM. An n job one machine algorithm for minimizing the number of late jobs. Manag Sci. 1968;15:102–9. https://www.jstor.org/stable/2628449.
[51] Lawler EL. Optimal sequencing of a single machine subject to precedence constraints. Manag Sci. 1973;19:544–6. doi: 10.1287/mnsc.19.5.544.