

Modified gorilla troops optimization for the quadratic assignment problem

Hussein Fouad Almazini¹, Salah Mortada², Hassan Al-Mazini³

¹Department of Computer Technology Engineering, Shatt Al-Arab University College, Basra, Iraq

²Department of Petroleum Project Management, Basra University for Oil and Gas, Basra, Iraq

³Department of Network Technologies and Computer Software, Southern Technical University, Basra, Iraq

Article Info

Article history:

Received Feb 20, 2025

Revised Feb 27, 2026

Accepted Apr 22, 2026

Keywords:

Combinatorial optimization

Gorilla troops optimizer

Metaheuristic

Optimization

Quadratic assignment problem

Swarm intelligence

ABSTRACT

Balancing exploration and exploitation remain a fundamental challenge in artificial intelligence-based optimization, particularly when addressing discrete combinatorial problems such as the quadratic assignment problem (QAP). The gorilla troops optimizer (GTO), inspired by the collective social behavior of gorillas, has shown promising results in continuous domains but faces limitations when directly applied to discrete optimization. To address this, the present study introduces a modified gorilla troops optimizer (MGTO), a novel discrete adaptation designed specifically for the QAP. The proposed MGTO strategically integrates a swapping-based diversification mechanism to enhance exploration within discrete solution spaces, while a modified uniform crossover operator promotes effective exploitation of high-quality solutions. Extensive experiments on benchmark instances from the quadratic assignment problem library (QAPLIB) show that MGTO achieves superior convergence behavior and solution quality compared with several state-of-the-art algorithms. These results demonstrate MGTO's capacity to maintain a balanced equilibrium between exploration and exploitation, effectively navigating complex discrete landscapes to yield high-quality solutions with strong computational efficiency.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Salah Mortada

Department of Petroleum Project Management, Basra University for Oil and Gas

Basra, Iraq

Email: salah.mortada@buog.edu.iq

1. INTRODUCTION

An example of a complex optimization problem (COP) that is categorized as nondeterministic polynomial-time hard (NP-hard) is the quadratic assignment problem (QAP). Commercial transaction modelling and quadratic allocation have been used in several applications, such as scheduling and site construction for hospitals and colleges [1]. The theoretical fields of computer science and operations research both face considerable challenges in solving this problem. Consequently, the growth of an effective algorithm to solve the QAP is crucial in terms of theory and practice [2]. Conventional approaches to solving the QAP can be separated into two groups: approximation algorithms and exact algorithms, often known as heuristics and metaheuristics, respectively [3]. Although exact algorithms can identify the global best solution, their time complexity drastically increases as the problem size grows, rendering them unsuitable for real-world applications. By contrast, approximate algorithms sacrifice accuracy for faster computation, attempting to discover a feasible solution close to the optimal one within an acceptable amount of calculation time [4]. The original method may no longer be optimal when the problem configuration changes because heuristics, which are typical approximation algorithms, suffer from poor adaptability [5]. In such cases, the model may require

revision. Conventional approaches can principal to a ‘combinatorial explosion’ when the problem size rises, causing temporal and spatial complexities to grow exponentially [6]. Consequently, the development of effective algorithms to solve COPs remains a challenging endeavor.

With the quick growth of computer science, artificial intelligence, and predator simulation methods, several new methods have emerged to address COPs by modelling predator behaviors such as hunting, attacking, quarrelling, and foraging [7]. These approaches offer an innovative approach to address the QAP, utilizing nature-inspired metaheuristics and predator behavior [8]. By emulating the adaptive and efficient strategies found in nature and predator-prey interactions, these heuristics and metaheuristics provide novel solutions for combinatorial optimization problems. They have demonstrated encouraging results in solving complex real-world problems across diverse fields, including transportation, logistics, network design, and scheduling.

Combining computer science with natural processes offers remarkable opportunities to improve optimization approaches and enable more complex, intelligent paradigms for the resolution of problems. Researchers are always working to improve these approaches that are inspired by nature, the QAP and other optimization problems stand to benefit. This study proposes a nature-inspired optimizer that replicates the collective behavior and foraging strategies of gorilla troops optimizer (GTO) [9]. This optimizer aims to replicate the interactions and decision-making processes amongst gorilla social systems to address challenging optimization issues [10]. In addressing continuous and discrete optimization problems, the GTO outperformed other metaheuristics [11]. This study introduces a modified gorilla troops optimizer (MGTO) as a discrete adaptation solution to solve the QAP by incorporating additional strategies to improve solutions through the exploration and exploitation of the search space, aiming to minimize the total cost of the task. The incentives and advantages of choosing the GTO are as follows.

The behavior of gorilla troops introduces exploration and exploitation, preventing the algorithm from falling into local optima, which is a crucial feature when addressing complicated problems such as the QAP. The GTO can be easily modified to handle discrete problems, such as the QAP, which require searching through permutations. The GTO mimics natural behaviors to equilibrium global exploration (to seek extensively) and local exploitation (to improve solutions). Additionally, the GTO has fewer parameters, reducing its storage requirements and overall complexity. The following are the study's main contributions.

This study addresses the fundamental challenge in artificial intelligence-driven optimization of achieving a robust balance among exploration and exploitation within discrete search spaces. To this end, a MGTO is proposed as a discrete adaptation of the standard GTO for solving the QAP. Unlike the original GTO, which was designed for continuous domains, the MGTO introduces a discrete swapping mechanism to enhance diversification and a modified uniform crossover operator to intensify exploitation around promising regions. Experimental results show the effectiveness of the suggested MGTO method is shown by testing on 36 examples from the quadratic assignment problem library (QAPLIB) test library.

The MGTO is also compared with four other optimization algorithms: ant colony (AC), bat algorithm (BA), genetic algorithm (GA), and tabu search (TS). The primary advantage of MGTO lies in its capability to rapidly identify ideal solutions. The proposed MGTO algorithm has strong potential for diverse real-world applications that can be represented as optimization problems in terms of assignment or layout. For instance, in hospital design, it can optimize the placement of departments to enhance workflow efficiency and patient movement; in very-large-scale integration (VLSI) chip design, it can minimize interconnect length and power consumption through refined module placement; and in manufacturing or facility layout, it can effectively allocate machines to reduce handling costs. Similarly, in campus or building planning, MGTO can manage multiple spatial objectives, ensuring accessibility and functional proximity. These applications highlight the algorithm's versatility and its ability to generalize across complex, discrete optimization domains.

The study is structured as follows: section 2 presents the QAP, while section 3 provides a literature review. Section 4 describes the suggested MGTO algorithm for solving the QAP. Section 5 covers the experimental findings, performance assessment, and QAPLIB benchmark instances utilized in this work. Finally, section 7 summarizes the article and makes recommendations for further work.

2. QUADRATIC ASSIGNMENT PROBLEM

The QAP is one of the most difficult NP-hard problems. This problem holds considerable theoretical value, especially in the investigation of high-performance algorithms. The QAP involves assigning economic facilities to locations whilst minimizing the associated costs, as illustrated in Figure 1. Allocating a set of facilities (n) to a set of locations (n), which has specified flows and distances between the facilities, is necessary. The flows and locations are represented by two $n \times n$ matrices, denoted by W and R , respectively. The flow between facilities p and q is denoted as W_{pq} , and the distance between locations i and j is denoted as R_{ij} .

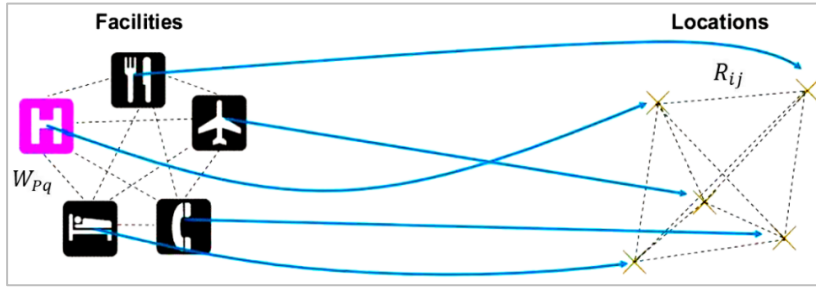


Figure 1. Description of the QAP at a high level

The goal is to position the facilities in such a manner that minimizes the total cost, which is the product of the flows and distances for all facility-location pairings. Mathematically, the problem is represented as (1).

$$F(\emptyset) = \sum_{p=1}^n \sum_{q=1}^{n-1} W_{pq} R_{\emptyset p \emptyset q} \tag{1}$$

As illustrated in Figure 2, the edges of the undirected graphs, formed by the flow and distance matrices, will be assigned values based on the assignment. These values represent the interaction strength or proximity between nodes, enabling clearer visualization and analysis of the problem structure. In addition to its theoretical importance, the QAP also has practical applications in various technical designs, such as the wiring of integrated circuits, work scheduling, the design of typewriter keyboards, and the layout of hospitals.

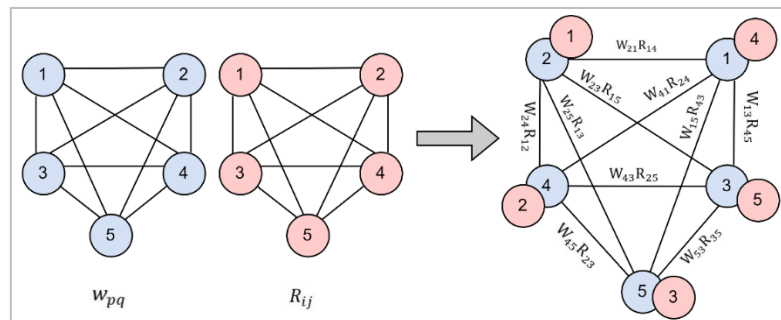


Figure 2. A graph model for QAP

3. LITERATURE REVIEW

Despite the availability of several algorithms, the development of highly efficient solutions to the QAP and the enhancement of existing algorithms continue to be active research areas. Additionally, no singular algorithm is capable of resolving all optimization problems, according to the No Free Lunch theorem. Therefore, solving the QAP using new or improved algorithms tailored to specific datasets is feasible. James *et al.* [12] introduced a cooperative parallel tabu search (CPTS) technique for solving the QAP. In contrast to other concurrent search methodologies that just compile data upon completion of execution, CPTS processors communicate information throughout the execution of the algorithm, making it a cooperative parallel algorithm. This collaboration is enabled by maintaining a global reference set, allowing for the exchange of information to promote strategic intensification and diversification in a synchronized context. The CPTS method demonstrates high solution quality for all problems whilst maintaining reasonable computing times. A limitation of the CPTS method is its high communication overhead between processors, which can reduce efficiency as the number of parallel processes increases. According to Benlic and Hao [13], breakout local search (BLS) was investigated as a method for addressing the QAP. BLS explores the research space by combining local research and adaptive disorder techniques. Experimental assessments on the QAPLIB benchmark sets, with an average compute time of less than 4.5 hours, show that the proposed method achieves the best-known results for all but two instances. A drawback of the BLS method is its sensitivity to parameter tuning, which can significantly affect performance across different problem instances.

Bouzidi and Riffi [14] examined the enhancement of the discrete cat swarm optimization (EDCSO) method, stimulated by the natural behavior of felines, to address the travelling salesman problem and the QAP. Simulated investigations were shown on numerous benchmark cases obtained from the operations research (OR)-library. The findings demonstrate the efficiency of the suggested modification in addressing real-world applications related to the two examined problems. A weakness of the enhanced discrete cat swarm optimization method is its tendency to converge prematurely on complex instances, leading to suboptimal solutions. A hybrid approach was introduced [15] to address the shortcomings of the standard biogeography-based optimization (BBO) method in solving the QAP. In this method, a tabu search technique is used instead of the mutation operator. Through tests on the QAPLIB benchmark instances, the proposed hybrid approach can deliver satisfactory results within a reasonable amount of time. In a test involving 61 benchmark examples, the proposed strategy successfully obtained the most well-known solutions for 57% of them. A limitation of the hybrid biogeography-based optimization with tabu search (BBO-TS) approach is its relatively low success rate in reaching the global optimum for more complex or large-scale QAP instances. Another related study, Riffi and Sayoti [16] investigated the performance of the golden ball algorithm combined with simulated annealing (GBSA) and focused on improving its performance to solve the QAP. The GBSA is based on several principles of football. The simulated annealing search may get trapped in a local optimum due to undesirable transitions. To effectively explore the search space and avoid local optimum solutions, the proposed method guides the search process using a simulated annealing algorithm. Several simulations were performed on 64 instances from the QAPLIB, comparing the GBSA with established techniques in the QAP literature, aiming to verify the GBSA methodology. A limitation of the GBSA method is its relatively high computational cost due to the combined use of golden ball optimization and simulated annealing procedures.

Additionally, a similar study conducted in [17] investigated the use of discrete rat swarm optimizer (DRSO) for solving the QAP. A mapping approach was applied to transform real numbers into discrete ones, and mathematical operators were reformulated to suit combinatorial problems, aiming to tackle the combinatorial aspect of the QAP. Furthermore, local search strategies like 2-opt and 3-opt were used to improve solution quality. The efficiency of the DRSO method was confirmed through simulations using test cases from the QAPLIB test library. A drawback of the DRSO method is its increased computational time caused by the integration of multiple local search heuristics, such as 3-opt and 2-opt.

The literature shows that most QAP metaheuristics suffer from early convergence and fail to reach optimal or near-optimal performance for larger instances within an acceptable amount of time. This discovery often results from an imbalance between local and global research, along with local stagnation and other influential variables. Consequently, the primary issue in developing a novel method lies in achieving an appropriate equilibrium between exploration and exploitation. Therefore, the search capability of the MGTO is enhanced for the QAP by implementing a swapping mechanism that leverages the exploration capacity of the GTO, whilst exploitation is facilitated through a modified uniform crossover, preserving equilibrium within the search space. The authors believe that no such algorithm has been developed to solve the QAP that fully leverages new methods in exploration and exploitation within the GTO framework. Table 1 (see Appendix) presents a comparative summary of key algorithms, highlighting their main strengths and weaknesses, along with the distinctive features of the proposed MGTO.

4. GORILLA TROOPS OPTIMIZATION

The GTO algorithm is stimulated by the behavior of gorilla packs, mimicking their social behavior while foraging for food such as leaves, stems, and fruits. As social animals, gorillas live in groups called packs. Each pack has an adult male, several adult females, and their young [9]. The GTO algorithm uses five distinct operators to manage exploration and exploitation during optimization tasks. During the exploration phase, three of these operators are used: migration to unfamiliar areas, movement towards other gorillas, and migration to familiar locations. Two factors are used during the exploitation stage to improve search efficiency. The GTO algorithm adheres to several guidelines, such as using the best solution (the silverback gorilla), gorilla position vectors (X), and candidate gorilla position vectors (GX). The algorithm aims to simulate the social dynamics of gorillas by moving from low-quality solutions to the optimal solution, thereby improving the positions of all gorillas. A detailed description of each phase of the algorithm is provided below to facilitate understanding.

4.1. Exploration phase

The processes that occur during the exploration phase are analyzed in the GTO algorithm. In the wild, gorillas live in groups led by a dominant adult male. Upon completion of the exploration stage, the costs of all options are assessed, and the ideal solution is identified as the dominant male. Similarly, in the

GTO algorithm, each gorilla represents a possible solution, and the optimal solution is chosen to become the dominant male. When embarking on an adventure, gorillas use three different mechanisms: migrating to a new area, returning to a familiar area, and interacting with another gorilla. The choice of these mechanisms is governed by a parameter known as p . These processes are modeled using (2). The exploration phase enables the algorithm to explore the entire range of the problem, improving the efficiency of exploration and avoiding local optimum solutions. The costs of all solutions are assessed at the conclusion of the exploration stage, and the most beneficial solution is chosen to become the dominant male.

$$X(t+1) \begin{cases} (UB - LB) \times r_1 + LB & rand > p \\ (r_2 - C) \times X_r(t) + L \times H & rand \geq 0.5 \\ X(t) - L \times (L \times (X(t) - GX_r(t)) + r_3 \times (X(t) - GX_r(t))) & rand < 0.5 \end{cases} \quad (2)$$

In (2) defines the position vector $GX(t+1)$ of the competing gorilla in the next iteration t , where $X(t)$ represents the gorilla's current position vector. The values of r_1 , r_2 , r_3 , and $rand$ are randomly chosen between 0 and 1 and are updated at each iteration of the algorithm. Before optimization begins, a value for the coefficient p must be set between 0 and 1. This number shows the chance of choosing a movement method to a place that hasn't been named. The factors' upper and lower limits are shown by UB and LB , correspondingly. In addition to GX_r , one of the candidate gorilla position vectors that is randomly chosen and updated at each stage, X_r refers to a randomly selected gorilla from the population. The values of C , F , L , H , and Z are finally calculated using (3) to (7), respectively.

$$C = F \times \left(1 - \frac{Iteration}{MaxIteration}\right) \quad (3)$$

$$F = \cos(2 \times r_4) + 1 \quad (4)$$

$$L = C \times l \quad (5)$$

$$H = Z \times X(t) \quad (6)$$

$$Z = [-C, C] \quad (7)$$

The current iteration value is represented by the variable in (3), and $MaxIteration$ represents the total number of iterations required to perform the optimization. The value of F is determined using (4), where \cos represents the cosine function, and r_4 represents random values between 0 and 1, adjusted at each iteration. In (3) produces numbers that exhibit abrupt fluctuations across a wide range during the early stages of optimization. However, this shift gradually diminishes as optimization progresses toward its final stages. The variable L is determined using (5), where l represents a random value within the range of -1 to 1. In (5) is used to model the driving behavior of a silverback gorilla. A silverback gorilla might initially not have sufficient experience to make good choices in terms of foraging or running the group. However, over time, this gorilla gains adequate knowledge, thus demonstrating stable leadership. Furthermore, in (2) necessitates the computation of H using (6), where Z is defined by (7). In this equation, Z denotes a random value inside the problem's parameters, which is constrained to the interval of $-C$ to C .

4.2. Exploitation phase

The GTO algorithm utilizes two behaviors during the exploitation phase: 'follow the silverback' and 'compete for adult females.' The silverback gorilla directs decisions and food sources, serving as the group's leader. Should the silverback weaken or die, another gorilla may assume leadership of the group. The choice between the two behaviors is determined by the C value in (3). If $C > W$, the 'follow the silverback' method is used; conversely, if $C < W$, competition among adult females is preferred. The youthful and vigorous silverback directs the group. In contrast, during the adult female behavioral competition, juvenile gorillas vie to enlarge their group. These behaviors may be replicated using (8) to (11). Subsequent to the exploitation phase, a group formation operation is executed, identifying the new silverback as optimal solution discovered.

$$GX(t+1) = L \times M \times (X(t) - X_{silverback}) + X(t) \quad (8)$$

$$M = \left(\frac{1}{N} \sum_{i=1}^N GX_i(t)\right) g \frac{1}{g} \quad (9)$$

$$g = 2^l \quad (10)$$

The gorilla location vector is represented as $X(t)$, the silverback gorilla location represented the best solution found, is referred to as Silverback. Furthermore, (5) is used to compute L , and (9) is used to calculate M . In (9) includes $GX_i(t)$, which shows the position of each possible gorilla in space throughout repetition t . The total number of gorillas is represented by the variable N . The assessment of L is similarly strong-minded via (5), and (7) is used to estimate g .

A position vector $X_{silverback}$ denotes the ideal solution for the silverback, as shown in (11), whilst $X(t)$ denotes the present position of the gorilla vector. The influence force is prejudiced by the adjustable Q , which is determined by (12), where the variable r_5 assumes random values within the range of 0 to 1. In (13) is used to compute the coefficient vector, which plays a role in determining the level of aggressiveness in disputes. To begin the optimization process, the parameter β must be allocated a value in (13), and the variable E is computed by (14) to mimic the effect of aggressiveness on the solution dimensions.

$$GX(i) = X_{silverback} - (X_{silverback} \times Q - X(t) \times Q) \times A \quad (11)$$

$$Q = 2 \times r_5 - 1 \quad (12)$$

$$A = \beta \times E \quad (13)$$

$$E = \begin{cases} N_1, & rand \geq 0.5, \\ N_2, & rand < 0.5, \end{cases} \quad (14)$$

5. MODIFIED GORILLA TROOPS OPTIMIZER FOR QAP

This section introduces an MGTO algorithm, emphasizing alterations to its diversification and intensification mechanisms. The GTO was designed to solve problems related to continuous improvement, but QAP is separate, so the GTO structure needs to be changed. In continuous space, the values inside the vector are adjusted to revise the positions. In discrete space, permutation transformations may be executed by operations like the swapping mechanism and modified uniform crossover, which are elaborated upon in subsection 5.1. While the fundamental concepts of GTO are preserved, this discrete variation functions inside the finite realm of permutations instead of continuous space.

5.1. Methods used

The fundamental concept behind neighborhood structure is that the solution is gradually refined through operations that specifically alter the solution within the neighborhood. Solutions that can be obtained through such movements are known as neighborhood solutions. A broad range of operators has been extensively used to address various COP problems, considerably improving the capability of algorithms to find optimum solutions throughout the optimization process. Consequently, to promote population variety and diminish the likelihood of premature convergence to local optima, this study incorporates two methods, namely the swapping mechanism and the modified uniform crossover, into a MGTO algorithm. The swapping mechanism and modified uniform crossover were selected because to its preservation of permutation validity, facilitation of nuanced solution enhancement, seamless integration with GTO, computing efficiency, and shown efficacy in prior QAP research. More intricate operators, such as 2-opt, partially mapped crossover (PMX), or cycle crossover, are either less compatible with the goal structure of QAP or entail more computational expense with less added advantage [18]. The following subsections describe these operators in detail.

5.1.1. Swapping mechanism

Figure 3 illustrates the concept of this technique. This figure involves the random swapping of two selected portions. The random swapping of these portions represents the main idea of the technique.



Figure 3. Swapping mechanism (before and after)

5.1.2. Modified uniform crossover

Genetic algorithms were the first to use the crossover operation for generating a new population. The goal of crossover when creating a new generation is to combine the best traits from both sets of parents. Several crossover operators have been presented as QAP solutions in the literature: the cohesive crossover [19], cycle crossover [20], swap path crossover [21], ordered crossover [22], multiple parent crossover [23], and uniform crossover [24].

This study proposes a modification to the uniform crossover to enhance population diversity and prevent early convergence into undesired areas of the search space. The MGTO algorithm incorporates this modified uniform crossover to improve its search strategy. The modified uniform crossover follows a similar procedure to the conventional crossover by scanning both parents from left to right and copying any items that occupy the same location in both parents to the offspring. The unallocated elements are organized into segments of size frequency (f). The first segment is inherited from one parent to the kid, whereas the subsequent segments are taken from the other parent, as long as they are not already present in the offspring. Otherwise, the missing elements are inherited from either parent in a specific manner to ensure that an offspring does not inherit unassigned locations from both parents. Figure 4 illustrates the modified uniform crossover procedure. Pseudocode 1 and Figure 5 present a pseudocode and flowchart to aid in understanding the MGTO algorithm.

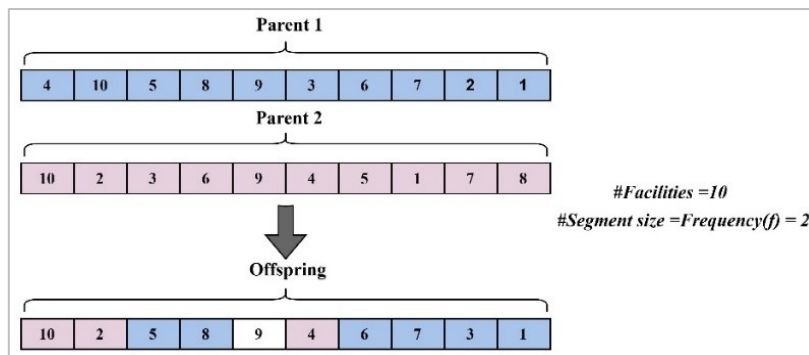


Figure 4. Example of modified crossover process

Pseudocode 1. MGTO algorithm for QAP

```
// MGTO Algorithm
Input: Parameters, maximum iterations T, population size N
Output: Best solution X_silverback
1: Start
2: Set parameters
3: Initialize population of gorillas
4: Evaluate fitness of initial solutions
5: for t = 1 to T do // Main loop
6:   Update parameters C and L using (3) and (5)
7:   // Exploration Phase
8:   for i = 1 to N do
9:     if random_condition_met then
10:      Update i-th gorilla using (2)
11:      Calculate i-th gorilla fitness using (1)
12:     else
13:      Update best candidate gorilla using swapping mechanism
14:      Calculate best candidate gorilla fitness using (1)
15:     end if
16:     if best candidate gorilla fitness > X then
17:       Replace X with best candidate
18:       Save as X_silverback
19:     end if
20:   end for
21:   // Exploitation Phase
22:   for i = 1 to N do
23:     if |C| >= 1 then
24:       Update i-th gorilla using (8)
25:       Calculate i-th gorilla fitness using (1)
26:     else
27:       Update i-th gorilla using (11)
28:       Calculate i-th gorilla fitness using (1)
```

```

29:     end if
30:   end for
31:   if GX (best current gorilla) fitness > X then
32:     Replace X with GX
33:   end if
34:   Update best candidate gorilla using modified uniform crossover
35:   Calculate best candidate fitness using (1)
36:   if best candidate fitness > X then
37:     Replace X with best candidate
38:     Save as X_silverback
39:   end if
40: end for
41: Return X_silverback
42: Stop
    
```

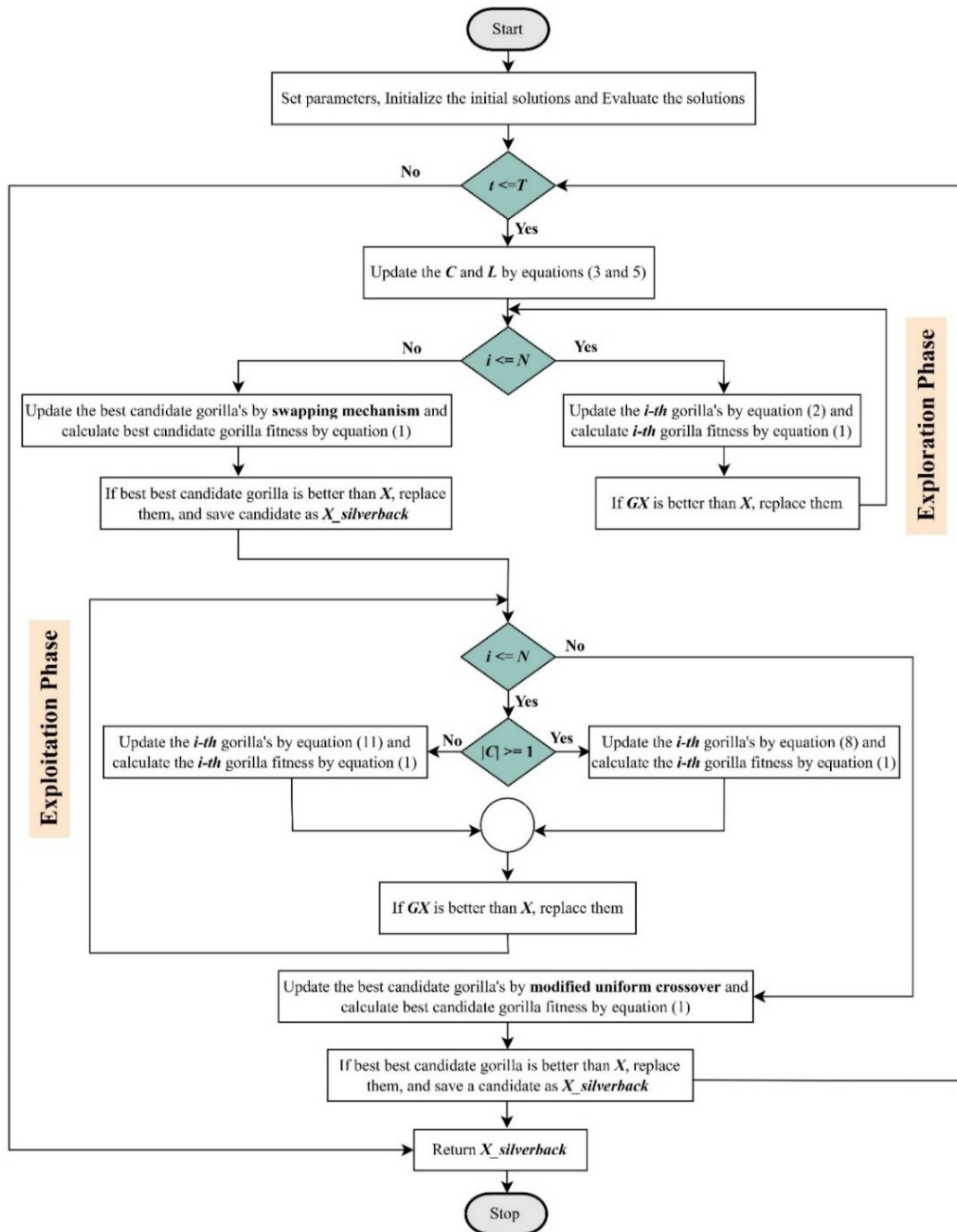


Figure 5. Flowchart of the MGTO algorithm for QAP

6. EXPERIMENTAL SETUP AND RESULTS

Computational tests are conducted to assess the performance of the suggested MGTO algorithm using benchmark data from QAPLIB. The benchmark QAP instances are categorized four groups: group 1 contains real-world examples derived from real-world QAP applications; group 2 comprises unstructured, randomly generated examples where the flow and distance matrices are generated using a uniform distribution; group 3 comprises randomly generated examples with a structure resembling real-world examples; and group 4 includes examples where the distances are calculated using the Manhattan distance on a grid. Comprehensive details regarding the problem instances and best-known solutions can be found at <https://coral.ise.lehigh.edu/data-sets/qaplib/qaplib-problem-instances-and-solutions/>.

The proposed MGTO was implemented in MATLAB R2021a and run on a machine equipped with an Intel Core i7-8665U CPU at 4.80 GHz. The parameters for both methods were set consistently across all test cases to allow for a reasonable comparison of their performances. The starting population size was set to 50 for all algorithms. The deployed parameter values for the MGTO algorithm, based on [9], are exposed in Table 2.

Table 2. Parameter settings for the MGTO algorithm

Parameter	Value
Population size	50
β	3
W	0.8
p	0.03

The computational findings are presented in two steps. Firstly, the results of the conventional GTO and the proposed MGTO are compared by applying them to 36 benchmark instances from QAPLIB, as shown in Table 3. Next, further testing is conducted with the MGTO algorithm against various optimization algorithms, as shown in Table 4. The algorithms performance is assessed using the criteria presented in Table 2. The proposed MGTO algorithm is run 10 times independently, and the average deviation is calculated using (15).

Table 3. Comparison of MGTO against GTO

Instance	Best known solution	GTO		MGTO	
		Best	$A - best\ GTO$	Best	$A - best\ MGTO$
esc16a	68	68	0.000	68	0.000
esc16b	292	292	0.000	292	0.000
esc16c	160	160	0.000	160	0.000
esc16d	16	16	0.000	16	0.000
had12	1652	1660	1.800	1652	0.000
had14	2724	2784	2.401	2724	0.000
had16	3720	3900	3.468	3720	0.000
had18	5358	5524	3.682	5358	0.000
had20	6922	7156	4.143	6922	0.000
chr12a	9552	9703	22.721	9552	0.000
chr12b	9742	9851	15.246	9742	0.000
chr12c	11156	11344	10.623	11156	0.000
chr15a	9896	12387	82.112	9896	0.000
chr15b	7990	10321	90.542	7990	0.000
chr15c	9504	12577	110.191	9504	0.000
chr18a	11098	23661	126.697	11098	0.000
chr18b	1534	2075	33.012	1534	0.000
nug12	578	578	0.000	578	0.000
nug14	1014	1079	6.101	1014	0.000
nug15	1150	1298	9.198	1150	0.000
nug16a	1610	1722	9.402	1610	0.000
nug16b	1240	1374	7.982	1240	0.000
nug17	1732	1811	7.120	1732	0.000
nug18	1930	2208	14.201	1930	0.000
nug20	2570	2746	10.287	2570	0.000
tai10a	135028	138457	4.545	135028	0.000
tai12a	224416	241620	9.132	224416	0.000
tai15a	388214	402982	6.991	388214	0.000
tai17a	491812	551128	12.772	491812	0.000
tai120a	703482	768832	9.878	703482	0.000
scr12	31410	37121	9.127	31410	0.000
scr15	51140	58663	19.115	51140	0.000
scr20	110030	139882	37.181	110030	0.000
rou12	235528	248760	8.126	235528	0.000
rou15	354210	397703	13.342	354210	0.000
rou20	725522	798871	14.523	725522	0.000

Table 4. Overall findings regarding the average performance

Algorithm	GTO	MGTO
Best average distance	1.930556	1.069444

$$(AD) = \frac{A - \text{best MGTO} - bks}{bks} \times 100 \tag{15}$$

Where $A - \text{best MGTO}$ is the average best MGTO value, and bks is the best-known solution value, as shown in Table 3.

Notably, the compared algorithms (best known solution) do not have any extra sets of parameters. The optimal comparison results from the best-known solutions, MGTO and conventional GTO are presented in Table 3. A wide variety of benchmark examples were used in the evaluation. The MGTO algorithm evidently performs far better than the conventional GTO method across all tested examples. This can find the ideal or best-known solutions for all 36 occurrences, whilst the conventional GTO algorithm can only identify five instances (esc16a, esc16b, esc16c, esc16d, and nug12). The average deviations from the most well-known solutions are also substantially lower for the MGTO across all benchmark instances tested.

Table 4 shows the statistical analysis of MGTO vs GTO performance using Holm's post hoc test and the nonparametric Friedman test. The goal of this test is to identify the most prevalent MGTO among 36 datasets. It includes the average distance. The lowest rank always denotes excellent algorithm performance, as seen in Table 4. The MGTO therefore attains the highest average rank in distance. As shown in Figures 6 and 7, MGTO successfully achieves the ideal solution for these instances (chr, tail, and scr), whereas GTO fails to reach the ideal solution for the majority of these instances. Table 5 presents a comparison of optimization algorithms, highlighting the optimal results of MGTO in this experiment. The table compares various algorithms from [25], including AC, BA, GA, and TS. The results in Table 5 demonstrate that, compared to AC, MGTO delivers the best results in three out of the cases and performs equally well in four cases. MGTO also outperforms BA, GA, and TS, achieving superior results in five cases and matching the results in two instances.



Figure 6. Comparison between MGTO and GTO using the Chr datasets

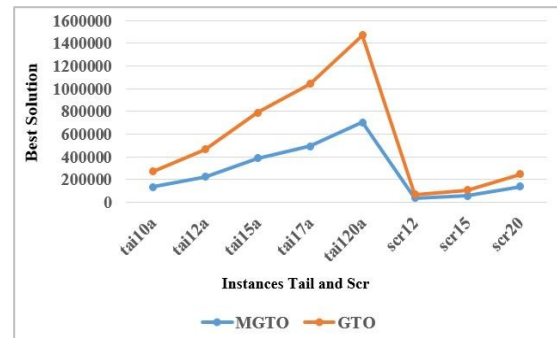


Figure 7. Comparison between MGTO and GTO using tail and the Scr datasets

Table 5. Results obtained by the MGTO via other algorithms

Problem instance	AC	BA	GA	TS	MGTO
	Best	Best	Best	Best	Best
nug12	578	594	586	594	578
nug28	5,178	5,924	5,986	5,292	5,166
esc16a	68	68	68	68	68
esc32e	2	2	2	2	2
tai12a	224,416	237,560	237,292	224,416	224,416
tai20a	705,622	771,094	781,742	724,558	703,482
tai30a	185,383,6	202,4320	203,9546	188,3148	181,8146

7. CONCLUSION

Over several decades, the development of intelligent behavior in swarm-based algorithms has become one of the most notable areas of study in solving COPs. This study proposes a MGTO, specifically designed to resolve COPs, focusing on the QAP. The algorithm is tested on 36 benchmark datasets and compared to four commonly used algorithms in the literature to evaluate the effectiveness of the suggested

MGTO method in resolving the QAP. The investigational results show that MGTO outperforms AC, BA, GA, and TS in terms of solution quality. In contrast to the common practice of developing an algorithm tailored to a particular problem, one of the key advantages of the proposed algorithm lies in its versatility, aiming to address a wide range of problems. Overall, this work contributes to advancing discrete optimization research by providing an effective framework that systematically balances exploration and exploitation in combinatorial problems. Future research could explore extending MGTO to address other COPs, such as postal delivery, school bus routing and the facility layout problem.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hussein Fouad Almazini	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓			✓
Salah Mortada	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	
Hassan Al-Mazini	✓	✓	✓	✓		✓	✓			✓	✓			✓

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

- [1] S. Samanta, D. Philip, and S. Chakraborty, "A Pareto communicating artificial bee colony algorithm for solving bi-objective quadratic assignment problems," *Opsearch*, vol. 62, no. 4, pp. 2239–2271, 2025, doi: 10.1007/s12597-024-00899-2.
- [2] A. Silva, L. C. Coelho, and M. Darvish, "Quadratic assignment problem variants: A survey and an effective parallel memetic iterated tabu search," *European Journal of Operational Research*, vol. 292, no. 3, pp. 1066–1084, 2021, doi: 10.1016/j.ejor.2020.11.035.
- [3] Z. Bai, V. Snášel, S. Mirjalili, B. Vo, L. Kong, and X. Wang, "A novel membrane-inspired evolutionary algorithm framework for VRPTW," *Applied Intelligence*, vol. 56, no. 2, 2026, doi: 10.1007/s10489-025-07068-y.
- [4] A. M. Jabbar and K. R. Ku-Mahamud, "Adaptive grey wolf optimization algorithm with neighborhood search operations: an application for traveling salesman problem," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 6, pp. 539–553, 2021, doi: 10.22266/ijies2021.1231.48.
- [5] V. Santucci and J. Ceberio, "On the use of the doubly stochastic matrix models for the quadratic assignment problem," *Evolutionary computation*, vol. 33, no. 3, pp. 425–457, 2025, doi: 10.1162/evco_a_00369.
- [6] S. Bouzidi, M. Bouzidi, and M. E. Riffi, "Solving the quadratic assignment problem using the swallow swarm optimization problem," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 3116–3120, 2019, doi: 10.35940/ijeat.F9132.088619.
- [7] D. P. F. Cruz, R. D. Maia, and L. N. De Castro, "A critical discussion into the core of swarm intelligence algorithms," *Evolutionary Intelligence*, vol. 12, no. 2, pp. 189–200, 2019, doi: 10.1007/s12065-019-00209-6.
- [8] M. N. M. Salleh *et al.*, "Exploration and exploitation measurement in swarm-based metaheuristic algorithms: an empirical analysis," in *Advances in Intelligent Systems and Computing*, 2018, vol. 700, pp. 24–32, doi: 10.1007/978-3-319-72550-5_3.
- [9] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems," *International Journal of Intelligent Systems*, vol. 36, no. 10, pp. 5887–5958, 2021, doi: 10.1002/int.22535.
- [10] J. Christiansen and K. S.-Miles, "Instance space analysis for the quadratic assignment problem," Jun. 2025, *arXiv:2506.20172*.
- [11] A. G. Hussien, A. Bouaouda, A. Alzaqebah, S. Kumar, G. Hu, and H. Jia, "An in-depth survey of the artificial gorilla troops optimizer: outcomes, variations, and applications," *Artificial Intelligence Review*, vol. 57, 2024, doi: 10.1007/s10462-024-10838-8.
- [12] T. James, C. Rego, and F. Glover, "A cooperative parallel tabu search algorithm for the quadratic assignment problem," *European Journal of Operational Research*, vol. 195, no. 3, pp. 810–826, 2009, doi: 10.1016/j.ejor.2007.06.061.
- [13] U. Benlic and J. K. Hao, "Breakout local search for the quadratic assignment problem," *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4800–4815, 2013, doi: 10.1016/j.amc.2012.10.106.

Modified gorilla troops optimization for the quadratic assignment problem (Hussein Fouad Almazini)




- [14] A. Bouzidi and M. E. Riffi, "Discrete cat swarm optimization algorithm applied to combinatorial optimization problems," in *WCCCS 2014 - Proceedings; 2014 5th Workshop on Codes, Cryptography and Communication Systems*, 2014, pp. 30–34, doi: 10.1109/WCCCS.2014.7107914.
- [15] W. L. Lim, A. Wibowo, M. I. Desa, and H. Haron, "A biogeography-based optimization algorithm hybridized with tabu search for the quadratic assignment problem," *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–12, 2016, doi: 10.1155/2016/5803893.
- [16] M. E. Riffi and F. Sayoti, "Hybrid algorithm for solving the quadratic assignment problem," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 4, pp. 68–74, 2019, doi: 10.9781/ijimai.2017.10.003.
- [17] T. Mzili, I. Mzili, M. E. Riffi, D. Pamucar, V. Simic, and M. Kurdi, "A novel discrete rat swarm optimization algorithm for the quadratic assignment problem," *Facta Universitatis, Series: Mechanical Engineering*, vol. 21, no. 3, pp. 529–552, 2023, doi: 10.22190/FUME230602024M.
- [18] A. Misevičius, A. Andrejevas, A. Ostreika, D. Verenė, and G. Žekienė, "An improved hybrid genetic-hierarchical algorithm for the quadratic assignment problem," *Mathematics*, vol. 12, no. 23, pp. 1–25, 2024, doi: 10.3390/math12233726.
- [19] A. Misevičius and D. Verenė, "A hybrid genetic-hierarchical algorithm for the quadratic assignment problem," *Entropy*, vol. 23, no. 1, pp. 1–31, Jan. 2021, doi: 10.3390/e23010108.
- [20] W. Marrouche, "An evolutionary algorithm tailored to the quadratic assignment problem," *Evolutionary Intelligence*, vol. 18, no. 6, pp. 1–21, 2025, doi: 10.1007/s12065-025-01100-3.
- [21] M. A. M. T. Baldé, S. Gueye, and B. M. Ndiaye, "A greedy evolutionary hybridization algorithm for the optimal network and quadratic assignment problem," *Operational Research*, vol. 21, no. 3, pp. 1663–1690, 2021, doi: 10.1007/s12351-020-00549-7.
- [22] H. Du, J. Li, and L. Li, "Genetic crossover operator with staged crossover strategy for the capacitated vehicle routing problem," in *2025 IEEE 17th International Conference on Computer Research and Development, ICCRD 2025*, Jan. 2025, pp. 161–169, doi: 10.1109/ICCRD64588.2025.10963039.
- [23] F. E. Sapnken, "A new hybrid multivariate grey model based on genetic algorithms optimization and its application in forecasting oil products demand," *Grey Systems*, vol. 13, no. 2, pp. 406–420, 2023, doi: 10.1108/GS-09-2022-0100.
- [24] M. Sohrabi, A. M. F.-Fard, V. A. Gromov, and M. A. Dulebenets, "A genetic engineering algorithm for the generalized quadratic assignment problem," *Neural Computing and Applications*, vol. 37, no. 18, pp. 12253–12279, 2025, doi: 10.1007/s00521-025-11155-z.
- [25] T. Achary, S. Pillay, S. M. Pillai, M. Mqadi, E. Genders, and A. E. Ezugwu, "A performance study of meta-heuristic approaches for quadratic assignment problem," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 17, pp. 1–29, 2021, doi: 10.1002/cpe.6321.

APPENDIX




Table 1. Comparative summary of metaheuristic approaches for the QAP

Reference	Algorithm	Main concept/strategy	Strengths	Weaknesses	How MGTO differs
[12]	CPTS	Parallel cooperative search with a global reference set for shared information exchange	High-quality solutions with reasonable computing times; effective cooperation between processors	High communication overhead reduces efficiency with more processors	MGTO avoids heavy inter-processor communication by maintaining a balanced internal exploration–exploitation process
[13]	BLS	A combination of local search and adaptive perturbation	Achieves near best-known results for most QAPLIB benchmarks; effective diversification	Sensitive to parameter tuning; performance varies across instances	MGTO requires fewer sensitive parameters, relying on adaptive exploration and controlled crossover instead
[14]	EDCSO	Swarm-based method inspired by feline behaviour	Effective for benchmark and real-world applications	Premature convergence in complex instances	MGTO reduces premature convergence via swapping-based exploration and modified crossover exploitation
[15]	BBO-TS	Integrates tabu search for improved exploitation	Delivers satisfactory results; performs well on mid-sized instances	Low success rate for large or complex QAP instances	MGTO enhances scalability by balancing global and local search adaptively
[16]	GBSA	Football-inspired optimization hybridized with simulated annealing	Escapes local optima; explores efficiently	High computational cost due to dual-method integration	MGTO maintains efficiency with fewer hybrid layers while improving the balance between exploration and exploitation
[17]	DRSO	Discrete adaptation with local search (2-opt, 3-opt)	Improved solution quality; strong local refinement	Increased computational time from multiple local searches	MGTO achieves similar refinement effects with lower computational cost through integrated swapping and crossover strategies




BIOGRAPHIES OF AUTHORS

Hussein Fouad Almazini    has a Ph.D. degree from the School of Computing, Sintok, Universiti Utara Malaysia, Kedah, Malaysia. He holds a master's degree in Information Technology in the area of artificial intelligence. He works in artificial intelligence, evolutionary computation, meta-heuristic algorithms, and swarm intelligence. He is also interested in mining scientific datasets in domains such as IoT, healthcare, cyber security, social networks, business, demographics, sustainability, and intelligence analysis. He can be contacted at email: h.almazni22@gmail.com.



Salah Mortada    is a Doctor of Philosophy at Basra University for Oil and Gas (BUOG). He has a Ph.D. in Information Technology specialization in artificial intelligence from Universiti Utara Malaysia. His research interests include artificial intelligence, evolutionary computation, meta-heuristic algorithms, swarm intelligence, and their applications in the design and optimization of intelligent transportation management, such as VRPs. He can be contacted at email: m.mortada@buog.edu.iq.



Hassan Al-Mazini    holds a bachelor's degree in Computer Science from Shatt Al-Arab University College, Basrah, Iraq (2013). He obtained his master's degree (2018) and Ph.D. (2022) in Information Technology with a specialization in artificial intelligence from Universiti Utara Malaysia. With over eight years of research and development experience in both academia and industry, his expertise spans artificial intelligence, evolutionary computation, meta-heuristic algorithms, and swarm intelligence. He is currently a lecturer at Southern Technical University, Basrah, Iraq. He can be contacted at email: hassan.f.abbas@stu.edu.iq.