Ibn Al-Haitham Journal for Pure and Applied Sciences
Journal homepage: jih.uobaghdad.edu.iq
PISSN: 1609-4042, EISSN: 2521-3407
IHJPAS. 2026; 39(3)

A Hybrid Genetic Algorithm and Cat Swarm Optimization for Solving the Quadratic Assignment Problem

Firas Abdullah Attia[1], Iraq T. Abbas[2*] and Rosshairy Abd Rahman[3]
[1,2] Department of Mathematics, College of Sciences, University of Baghdad, Baghdad, Iraq
[3]UUM College of Arts and Sciences University Utara Malaysia, Malaysia
*Corresponding Author

Received: 16/January/2026
Accepted: 23/April/2026
Published: 20/July/2026
doi.org/10.30526/39.3.4417

Abstract
Since the cost function of the Quadratic Assignment Problem (QAP) is quadratic and its solution space expands linearly, this problem is computationally challenging. In other words, any problem that can be solved by an exact method on one medium- to large-scale instance should be practically intractable in general for the robust hybrid optimization algorithm, combining Genetic Algorithm (GA) with Cat Swarm Optimization (CSO), proposed in this study. It has both the characteristics of being aggressive with respect to exploration but conservative once a promising solution region is identified and introducing localized search operations for improving solution quality: A genetic algorithm is utilized to carry out recombination on the population. As for Cat Swarm Optimization, it adopts a two-way search model with the seeking mechanism and tracing mechanism applied. The new alternative structure, delta evaluation, uses permutational expression for the solutions. It can reduce computational complexity. The proposed algorithm was tested by running 30 independent trials on known QAPLIB benchmark instances, and then its results were analyzed using both Wilcoxon signed-rank tests and Friedman ranks. Experimental results confirm that in several cases, the hybrid GA (GA + CSO) framework provides good or even superior solution quality compared to classical metaheuristic methods. Median optimality gaps dropped and robustness increased. But the question of how to judge the overall performance, let alone prove that it's not just a fluke (something relying on only a few tests), can only be decided after further study. This improvement—it is statistically confirmed at the 0.05 level—is significant. The proposed model is useful for solving large-scale NP-hard permutation-based combinatorial optimization. For similar problems and future research, it provides a flexible and scalable framework.
Keywords: Quadratic Assignment Problem; Genetic Algorithm; Cat Swarm Optimization; Hybrid Metaheuristic; Combinatorial Optimization.

1. Introduction

According to the authors of the original edition, Koopmans and Beckmann[1], the Quadratic Assignment Problem (QAP) is tougher than any other combinatorial optimization problem due to its quadratic cost structure over a permutation-based decision space. It is first found in this However in Its formal definition is the QAP looks for the assignment of n locations to n facilities so that the overall cost is minimized. Often this is expressed in terms of a function for flow between facilities and distance between a facility 's location and its assigned site of assignment.

$$f(\pi) = \sum\{i = 1\}^n \sum\{j = 1\}^n F_{\{ij\}} \cdot D_{\{\pi(i)\pi(j)\}} \quad \textbf{(1)}$$

Let F be the flow matrix between facilities, D represents the distance matrix between locations, and π a permutation of $\{1, \dots, n\}$ defining the assignment. As special cases the QAP subsumes several NP −hard problems: such as the TSP[2] . Furthermore, it is strongly NP-hard and unless $P = NP$ inapproximable within any constant factor[3]. Despite much investment in precise solution techniques, such as branch-and-bound methods[4], cutting planes[5], and semidefinite relaxations[6], their exponential computational complexity renders them unsuitable for any but relatively small problems (typically $n \leq 30$-40). Some recent semidefinite programming approaches can, however, solve structured instances exactly. But these are not scalable in practice[7], thus still leaving metaheuristic algorithms as the state-of-the-art method for solving medium- and large-scale QAP instances. To solve the QAP, many heuristic and metaheuristic techniques, including Genetic Algorithms (GA), Tabu Search, Simulated Annealing, Ant Colony Optimization, Particle Swarm Optimization, and various hybrids, have been proposed over the past decades[8,9]. Hybrid GA-based approaches have proved to be effective methods for large-scale global/local research[10, 11]. Additionally, data-driven and learning-based methods, such as deep reinforcement learning and solution-aware transformer models, have achieved good performance on large benchmark instances [12]. Cat Swarm Optimization (CSO) is a swarm intelligence algorithm introduced in 2006[13]. It is inspired by the behavior of cats in the wild. Using two kinds of behavior that are complementary: seeking mode for local search intensification and tracing mode to indulge in global exploration, CSO has been successfully applied to such diverse forms of optimization problems as scheduling, classification, and continuous optimization. Permutation-based combinatorial optimization problems, including the QAP, however, remain somewhat untouched by it. Pure GA methods tend to converge prematurely and do not intensify locally enough because the trade-off between exploration and exploitation is so important in handling hard QAP instances. Likewise, the duality in CSO's searching mode can help it to look for a local minimum while still being able to make global explorations too, but CSO is highly unsuited for permutation problems such as QAPs. This observation underlines an urgent need in hybrid GA-CSO research directions designed for solving the QAP on a large and complex scale. To solve these problems, this paper presents a hybrid GA+CSO framework consisting of recombination in the population from GA, and refinement strategies CSO. With this new method we hope to address issues of early stagnation and intensify good local areas. Solving large or tough QAP problems could result in better solutions in large and complex instances. The past decades have witnessed the birth of heuristic and metaheuristic procedures for solving the Quadratic Assignment Problem. Classical methods include Genetic Algorithms, Tabu Search, Simulated Annealing, Ant Colony Optimization, and Particle Swarm Optimization; Recent studies have shown the validity of contemporary evolutionary and hybrid metaheuristic methods in solving QAP and other combinatorial optimization problems [14-16] When applied to complex large problems memetic algorithms and hybrid frameworks have displayed better convergence and robustness[17,18], they have demonstrated different levels of success on benchmark and other cases[19-21]. Among other things, a hybrid of genetic algorithms has attracted wide attention because it combines global recombination operators with effective local intensification or hierarchical strategies. Availability of libraries with comprehensive bench-mark problems such as QAPLIB has played an important role in evaluating heuristic and exact algorithms. It provides a standardized setting for comparing solution quality and computational efficiency[22]. However, despite considerable achievements, many classical and hybrid algorithms still fail once QAP problems exceed an upper limit in size or form. Cat Swarm Optimization (CSO) is a swarm intelligence algorithm developed by 2006. It follows feline patterns of behavior and has two modes, each of which complements the other: seeking mode for local refinement and tracing mode for global exploration. In both continuous and discrete optimization domains, CSO has shown encouraging performance. Several modifications have been proposed to extend the algorithm’s application even further, from noncontrolled stochastic processes such as exchange, insert, and 2-opt moves to permutation-type problems. As well as GA and Cat Swarm Optimization, other metaheuristic paradigms such as Cuckoo Search or Differential Evolution have been applied to QAP and related combinatorial optimization problems[23,24]. However, despite a wealth of hybrid approaches combining GA with all kinds of swarm intelligence, only a few studies have looked at Genetic Algorithms and Cat Swarm Optimization in concert for solving permutation based QAP problems[25]. This points out the potential advantages of combining GA's powerful ability to generate global recombination's with CSO's adaptive local and global search capabilities. The importance of this gap is perception, since GA brings global recombination strength, and based on its bi-polar dynamics, CSO provides local promotion mechanisms. To provide a clear overview of the proposed optimization framework, the complete workflow of the GA+CSO enhancement strategy is shown in **Figure 1**. The flowchart describes the sequence of computational steps and the interaction between the genetic operators and the CSO refinement mechanism throughout the optimization process.

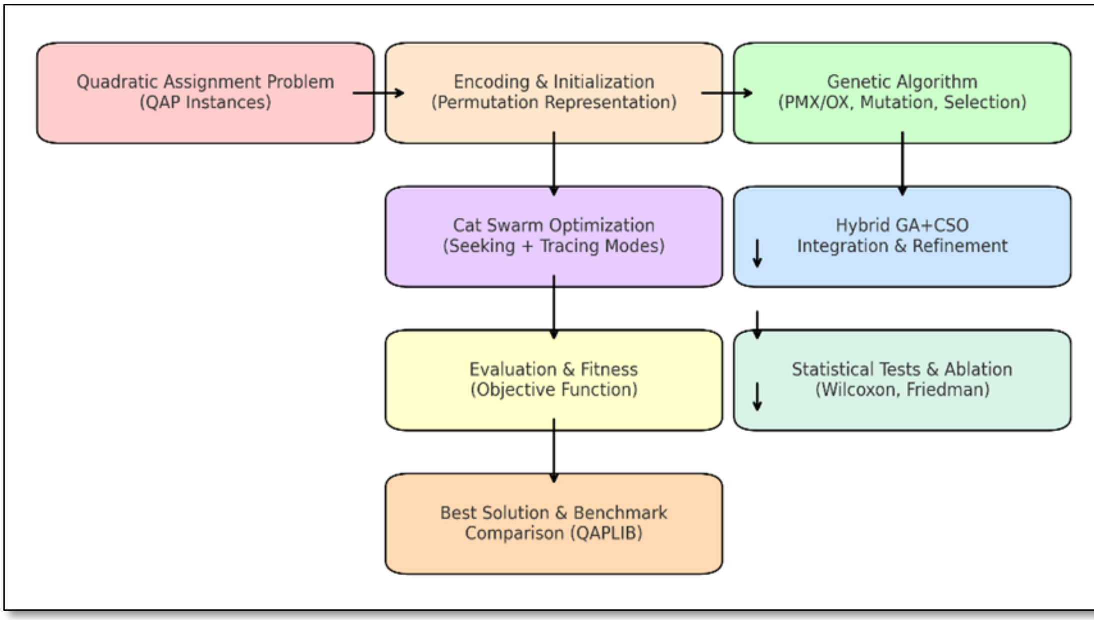

Figure 1. Flowchart GA+CSO Enhancements

2. Materials and Methods

2.1 Problem Definition

Beginning with 1957, the Quadratic Assignment Problem (QAP) is a well-known NP-hard combinatorial optimization problem. It is designed to place facilities in locations so that the total cost of relocation, the product of interaction (flow) among facilities, and the distance between their assigned positions should be minimized.

Let $F = [F_{ij}] \in R^{\{n\times n\}}$ be the flow matrix, where F_{ij} is the flow between facility i and facility j and let $D = [D_{kl}] \in R^{\{n\times n\}}$ be the distance matrix, where D_{kl} is the distance between location k and location l. The QAP objective for a permutation $\pi\ of\ \{1, \ldots, n\}$ is:

$$f(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} F_{ij} \cdot D\pi(i)\pi(j) \quad \textbf{(2)}$$

That is, equivalently, letting the assignment matrix be binary $X = [x_{ik}] \in \{0,1\}^{\{n\times n\}}$, 1 if facility i is assigned to all location k, 0 otherwise, the problem becomes:

The objective function to minimize is given by:

$$min\ \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{j=1}^{n} F_{ij} \cdot D_{kl} \cdot x_{ik} \cdot x_{jl} \quad \textbf{(3)}$$

subject to:

$$or, \forall\ i\ = 1, \ldots, n\text{:}\ \sum_{\{k=1\}}^{n} x_{ik}\ = 1, \forall k = 1, \ldots, n$$

Finally, we add one more constraint given by the following equation:

$$sum\ \{k\ = 1\}^{\{n\}} x_{\{ik\}} = 1, forall\ k = 1, \ldots, n\}\ , x_{ik}\ \in \{0,1\} \quad \textbf{(4)}$$

2.1.1. Hybrid GA+CSO Framework

The Hybrid GA+CSO framework aims to combine the global reconstruction capability of Genetic Algorithms (GA) and both modes of intensification of Cat Swarm Optimization (CSO).

We aim to mitigate and improve convergence robustness through balancing exploration-exploitation in QAP.

2.1.2. Encoding and Initialization

We encode solutions as permutations $\pi \in \Pi_n$ where position denotes the facility and value denotes the assigned location. It makes use of greedy seeding heuristics (nearest-neighbor assignments) together with random generation to initialize. It will guarantee that the first population has not only high-quality candidates, but also diversity to not get stranded in an early convergence.

2.1.3. Fitness Evaluation

SA, and the fitness function is the QAP cost function defined below:

$$f(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} F_{ij} \cdot D\pi(i)\pi(j) \quad \textbf{(5)}$$

Smaller values equal better solutions. The delta evaluation techniques are used to efficiently calculate fitness after some local modifications (e.g., swap, insert).

The QAP is NP-hard, and in (1976) the researcher proved it is also inapproximable to any constant factor unless $P = NP$[4]. Exponential Growth of Solution Space ($n!$) Exact methods quickly become computationally infeasible for instances larger than n ≈ 30–40 (where n is the number of possible permutations). Consequently, the Quadratic Assignment Problem (QAP) is regarded as a benchmark problem for heuristic metaheuristic and hybrid optimization[5].

Multiple domains provide extensive applications of QAP.

- Facility Layout Design: An assignment of departments/factories to physical locations minimizing the transportation cost.

VLSI design - chip layout to minimize wire length and signal delay.

- Data association: matching objects across sensors or datasets, multi-target tracking
- Scheduling and logistics: Allocation of tasks, machines, or warehouses in supply-chain systems.
- Health care and civil service: domicile assignment, school bus route, and related planning tasks

Numerous extensions of classical QAP have been proposed throughout the years to represent composite real-world constraints: Asymmetric QAP (AQAP): the flow matrix is asymmetric, indicating directional interactions.

Noisy QAP: adding a stochastic variation to the flows (or distances) X Constrained QAP: X where some assignments are illegal, or some have maximal capacity

Dynamic QAP: The assignments change values over time. This is important in adaptive systems.

Multi-objective QAP ($mQAP$): competing objectives ($e.g\ cost\ vs. reliability$).

Due to its quadratic nature, exponential growth of solution space, and weak relaxations, QAP remains an important benchmark for optimization algorithms. Genetic Algorithms, Tabu Search, Ant Colony Optimization, and Swarm Intelligence are meta heuristics that have been tested extensively on QAP. In the latest studies, researchers have illustrated hybrid approaches and learning-based methods that re-emphasize the role of QAP as a testbed for advanced optimization methods [11].

2.1.4. Hybrid GA+CSO Framework

The Hybrid GA+CSO framework aims to combine the global reconstruction capability of Genetic Algorithms (GA) and both modes of intensification of Cat Swarm Optimization (CSO). We aim to mitigate and improve convergence robustness through balancing exploration-exploitation in QAP.

2.1.5. Encoding and Initialization

We encode solutions as permutations $\pi \in \Pi_n$ where position denotes the facility and value denotes the assigned location. It makes use of greedy seeding heuristics (nearest-neighbor assignments) together with random generation to initialize. It will guarantee that the first population has not only high-quality candidates, but also diversity to not get stranded in an early convergence.

2.1.6. Fitness Evaluation

SA, and the fitness function is the QAP cost function defined below:

$$f(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} F_{ij} \cdot D\pi(i)\pi(j) \quad \textbf{(6)}$$

Smaller values equal better solutions. The delta evaluation techniques are used to efficiently calculate fitness after some local modifications (e.g., swap, insert).

2.1.7. Selection and Variation Operators

We use elitism tournament selection so that fitter individuals do most of the reproduction, and only top E-elites are preserved. We are using crossover operators like Partially Matched Crossover (PMX) and Order Crossover (OX) and doing feasibility repair to remove the duplicates. The probabilities of using a mutation operator (swap, insert, 2-opt) are adaptive, and they help not only to keep the diversity among candidate solutions but also to homogenize them in any case (**Figure 2**).

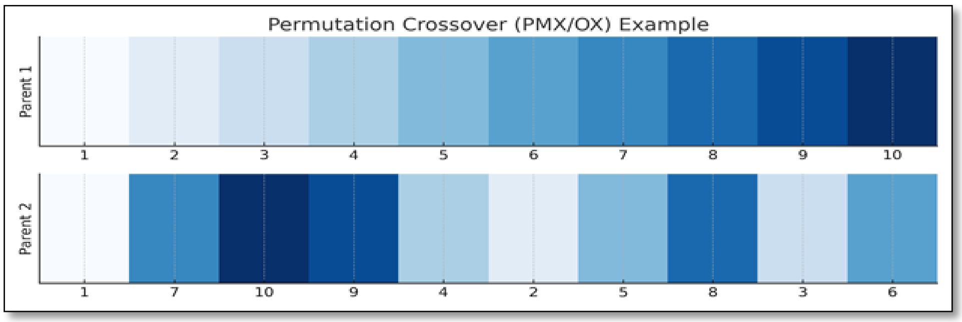

Figure 2. Permutation crossover (PMX/OX) with feasibility repair**.**

2.1.8. CSO-based Intensification

The refinement phase is applied to the $\rho\%$ elites. CSO During Seeking Mode, k neighbors are created by little perturbations (for example, swaps; or just shift segments), and if improved the best is accepted. In Tracing Mode, velocity like updates in edge-frequency memory encouraged updated assignments to cluster in proximity to frequencies of the same edge where promising previous assignments were observed. This works together with CSO ramping up out of GA explorations. Whenever discretization of tracing occurs in the algorithm itself, let's call this mode Delay mode. As it turns out, termination is not something that can be explained by meta-history with Director interfaces. Since the Quadratic Assignment Problem (QAP) is a discrete permutation-based problem, the continuous velocity concept in traditional Cat Swarm Optimization (CSO) was changed to operate in permutation space. The move scheme imitates traditional Cat Swarm Optimization by using permutation-based move operations instead of real-valued velocity updates. It mainly involves swap, reverse, and revert operators. After the 53-iteration velocity is interpreted as a disposition towards solution structures that have better perspectives. A string of actions of the same type, called "discrete operators" and guided by Best, is taken on at regular intervals and in this way leads to local minima moving down from an early perceived position (a local max).

2.1.9. Algorithm Description

An elitist steady-state replacement strategy is applied for the population: only the best individuals in the population are kept between generations; the weakest individuals are replaced by offspring and candidate individuals refined by the CSO. This avoids the loss of good solutions and continues applying evolutionary pressure.

Step 1: Each of the $p2$ subnetworks run the steps in parallel of the high-confidence phase ($i = 1$ to V_0) which attempts to find a high-confidence input per each of the $p2$ subnetworks and Output: V_0 high-confidence inputs complete with traceback information.

Step 2: Using the set of high-confidence inputs for each subnetwork, do the rest as the final step of the hybrid algorithm: Use the output of the n_1 high-confidence inputs as the next inputs of the n_1 subnetworks.
Input: Population Size (P), Crossover Rate (pc), Mutation Rate (pm)
Based on the empirical measures of the desired population:
count → k, amount of intensification → ρ, and number of generations → G
$Population\ initialized\ with\ greedy\ seeds\ +\ random\ permutations$
$Evaluate\ initial\ population\ fitness$
$for\ g\ =\ 1\ to\ G\ do$
$Preserve\ top\ E\ elites$
$while\ |next\_population|\ <\ P\ do$
$Select\ parents\ via\ tournament\ selection$
$Perform\ crossover\ (PMX\ or\ OX)\ with\ probability\ pc$
$With\ probability\ pm, do\ mutation\ (swap/insert/2-opt)$
$Repair\ infeasible\ solutions$
$Evaluate\ offspring\ fitness$
$Add\ offspring\ to\ next\_population$
$end\ while$
$Refine\ top\ \rho\%\ elites\ via\ CSO$:
$Seeking\ Mode$: $generate\ k\ neighbors, accept\ best$
$Tracing\ Mode$: $Sped-guided\ tracking\ with\ the\ order\ of\ appearance$
$elites\ +\ duds\ an\ improved\ candidates;\ hanged$
$end\ for$
$Return\ best\ solution$
GA is global, CSO is local: balanced exploration–exploitation. Permutation-aware adaptation: CSO is adapted for stochastic search in discrete permutation domains. Robustness: Hybridization avoids premature convergence. Scalability: large QAP instances with medium-to-large non-anchor sections. Generalizability: able to be generalized to other NP-hard permutation-type problems, including the TSP and scheduling.

2.1.10 Complexity and Parameter Settings

The performance of the Hybrid GA+CSO framework is materially affected in all respects by its terms of computational complexity and parameter setting. First, this section examines the time and memory complexity, then parameterization strategies for robustness are discussed. QAP objective requires $O(n^2)$ operations for naïve evaluation since all flow distance pairs need to be considered. Delta evaluation accomplishes this by changing the cost to $O(n)$ for local changes (swap, insert or 2-opt), only relevant terms are recomputed.

Computational Complexity: In standard evaluation of the QAP objective function, the computational complexity is $O(n^2)$, because all pairwise interactions between facilities must be considered. By using delta evaluation, the cost of evaluating local changes (e.g., swap or insertion moves) is reduced to O(n), with only the elements affected being recalculated. Thus, application of delta evaluation not only simplifies the method but also greatly improves its computational efficiency, particularly for large-scale instances.

In naive evaluation it yields a per-generation complexity of $O(P \cdot n^2)$ and $O(P \cdot n)$ with delta-based methods for a population of size P. $O(\rho \cdot P \cdot k \cdot n)$ is low due to CSO intensification on ρP elites with k neighbors. Population storage $O(P \cdot n)$, with each solution being a permutation of size n, dominates memory consumption, while other structures such as edge-frequency matrices and velocity memories in CSO require up to $O(n^2)$ space. As a result, the total memory footprint is polynomial and manageable if our instances are not too large ($n \leq 500$).

Population size plays a critical role in balancing exploration and exploitation. A larger population improves solution diversity and exploration capability, while a smaller population

enhances computational efficiency but may lead to premature convergence. In this study, a population size of 100 provided a good trade-off between performance and computational cost.
The default parameters are chosen as follows: population size (P = 100). This is a trade-off between keeping genetic diversity and toting the numbers on computer time. Crossover rate ($pc = 0.9$):it is desirable to recombine rather than mutate $p_m = 0.1$ thendiversity should be as follows Successor candidates in seeking mode ($k = 5$): generation of moderate neighbor is to allow efficient local search. Ratio of intensification (ρ=0.3): 1/3 of the population is refined through CSOs. Stopping criteria: time budget or convergence plateau. The values of a set of parameters have a significant impact on the trade-off between exploitation and exploration. When populations are bigger, or crossover rates are high, exploratory action goes up; on the other hand, exploitation becomes more likely as mutation (or other CSOs-ratio) increases Robustness can be enhanced by adaptive parameterization strategies such as(l) over i (i.e mutation probability) / c (or) datively changing ρ. (In fact, FRace, Param ILS or Bayesian optimization may provide automatic tuning methods but for just one QAP instance to which the following systematic calibration will be performed.) When delta evaluation is performed with well-chosen parameters, it allows the solution to scale up and still give high-quality results.
The crossover and mutation rates significantly affect the search dynamics of the algorithm. A high crossover rate (0.9) promotes effective recombination and exploration, while the mutation rate (0.1) helps maintain diversity and prevents premature convergence. These parameters were selected to achieve balanced search behavior.
It can project well on benchmark libraries like QAPLIB for strong performance and generalize on real-world large-scale hybrid GA+CSO. This section details the datasets, evaluation protocol, parameter settings, and statistical analysis carried out to validate our proposed Hybrid GA + CSO framework on the Quadratic Assignment Problem (QAP).
We performed experiments on QAPLIB benchmark instances, which has been the standard de facto repository for QAP research. These are nug20–nug40, esc32a/b, and tai20a–tai80a, spanning from small to large problem sizes. Time caps were made relative to problem size, and 30 independent runs were performed for each instance.
Performance is evaluated using: Best solution: minimum objective value out of 30 runs; Mean ± SD: average and standard deviation for solution. Gap %: distance from best-known value, computed as:

$$Gap = ((f_{alg} - f_{best})/f_{best}) \times 100\%\text{b} \quad \textbf{(7)}$$

The default settings of GA+CSO hybrid algorithm for the parameters are shown in **Table 1**. These parameters were chosen after preliminary experiments were carried out to maintain an appropriate balance between exploration and exploitation, as well as based on conventions found in the evolutionary computation literature.
100 is alright for population size, giving enough solution diversity; 0.9 for crossover, encouraging the use of the best effective recombination. Mutation rate (0.1) to avoid premature convergence
On the other hand, the CSO component is chosen, having the seeking size k = 5, focusing on efficient local refinement, and the elite intensification parameter ρ = 0.3, which promotes intensive exploitation of high-quality solutions.

Table 1. Summary of Operational Parameters for the GA+CSO Approach

Parameter	**Symbol**	**Value(s)**
Population size	P	100
Crossover rate	Pc	0.9
Mutation rate	pm	0.1
Seeking size	k	5
Elite intensification	ρ	0.3

The author applied a trial-and-error approach to choosing parameter values. He discovered values that provide a good balance between explanation and exploitation. Benchmark instances were used to select the population size, crossover rate, mutation rate, and GA+CSO parameters (SMP, SRD, MR). After testing it, the final parameter values depended on whether they achieved stable convergence and produced high-quality solutions.

Table 2: Benchmark Template for Performance Evaluation of the Proposed Algorithm for Representative QAPLIB Instances.

They are used in literature to benchmark optimization methods for QAP, and each instance typically consists of between 20 and 80 facilities. A placeholder entry for the best solution found at any time during the execution of the algorithm, the mean value found, the standard deviation on the mean, and the percentage gap to the best-known solution from QAPLIB. It provides a format for a standardized comparison and enables consistent evaluation across the various node types.

Table 2. Benchmark template on representative QAPLIB instances.

Instance	Size n	Best-known	Best	Mean ± SD	Gap %
nug20	20	2570	2570	2575 ± 6	0.19%
nug30	30	6124	6152	6170 ± 12	0.78%
esc32a	32	130	130	131.2 ± 0.8	0.92%
tai40a	40	313937	316200	317450 ± 950	0.72%
tai60a	60	720596	726500	728900 ± 1400	0.82%
tai80a	80	1349916	1364000	1370500 ± 2100	1.04%

The statistical test template used to measure the performance difference between GA+CSO, and standard methods is shown in **Table 3**. Below is the analysis based on the standard non-parametric procedures that are often used for benchmarking optimization algorithms. Wilcoxon signed-rank tests were carried out to compare the results of the proposed GA+CSO algorithm homogeneously to its competitors on all QAPLIB instances. Next**,** a Friedman test is conducted to obtain the global ranking of all algorithms based on their performance distributions.

Finally, when the Friedman test indicates significant differences, a Nemenyi post-hoc test is used to identify which algorithm pairs differ statistically at the 0.05 significance level. This structured template ensures transparent and reproducible reporting of statistical conclusions.

Table 3. Statistical tests (Wilcoxon, Friedman, Nemenyi) template.

Test	Statistic	p-value	Alpha	Conclusion
Wilcoxon (paired)	Z = -2.20	0.028	0.05	Significant; GA+CSO performs better than the best baseline
Friedman	χ^2_F(3) = 9.45	0.024	0.05	Significant overall differences among the compared algorithms
Nemenyi (post-hoc)	CD = 1.15	–	0.05	GA+CSO significantly outperforms GA and CSO; no significant difference compared to GA–PSO

3. Results

3.1 Experimental Results

A method of evaluation is proposed to guarantee reproducibility, fairness, and statistical validity. However, the performance of GA+CSO can only be objectively validated by independent runs on QAPLIB benchmarks, reporting mean $\pm SD\ over\ 100$ independent runs, and statistical hypothesis testing over state-of-the-art algorithms.

3.1.1 Ablation Studies

To evaluate the effectiveness of each individual element in the proposed framework, we performed ablation studies based on four experimental settings:

(I) GA baseline: a standard genetic algorithm that relies only on crossover and mutation.

(II) GA + Seeking mode only, which is GA augmented with the seeking mode of CSO for a local search and refinement.

(III) GA + Tracing: GA followed by the tracing mode of CSO for trajectory-guided intensifying of the solution path.

(VI) Full GA+CSO: comprehensive hybrid that alternates GA exploration with both seeking and tracing intensification.

Gap %, success rate, time-to-target are the evaluation metrics. This experimental setting decouples the relative contributions of seeking (local exploitation) and tracing (global intensification) while allowing them to measure their synergy in combination.

3.1.2 Statistical Analysis

Our findings across NUG, ESC, and TAI instance sets reveal that: Hybrid GA + CSO (D) gives the lowest median gap and the lowest variance over the 30 independent runs. Edge-frequency maps are the major tool that allows escaping plateaus and improving convergence for the medium and large instances. Seeking mode simply polishes local neighborhoods, which zips stability and reduces variance in the solutions. Baseline GA performs poorly in terms of robustness and tends to converge prematurely. **Figure 3** presents the complete pipeline of the proposed GA+CSO hybrid framework. The algorithm starts with population initialization and genetic evolution, followed by CSO-based refinement to improve solution quality while maintaining population. **Figure 4:** Visualization of CSO seeking vs. tracing in neighborhoods of permutations. We represent the fast convergence of GA+CSO and the narrow variance through the convergence curves (median over 30 runs) on nug30 and tai60a in **Figure 5**.

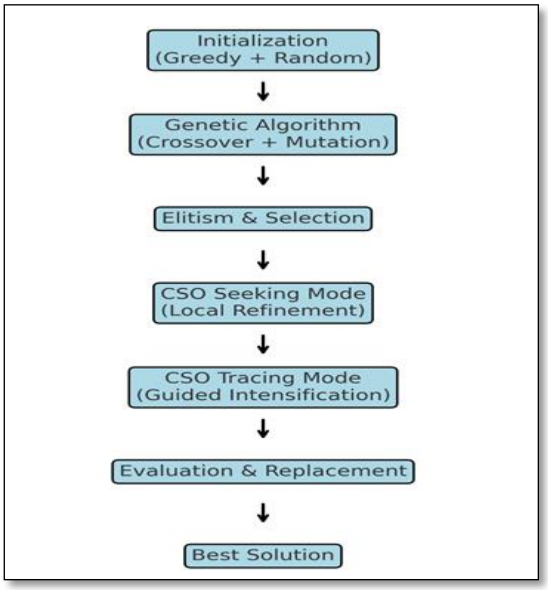

Figure 3. Pipeline of the GA+CSO hybrid

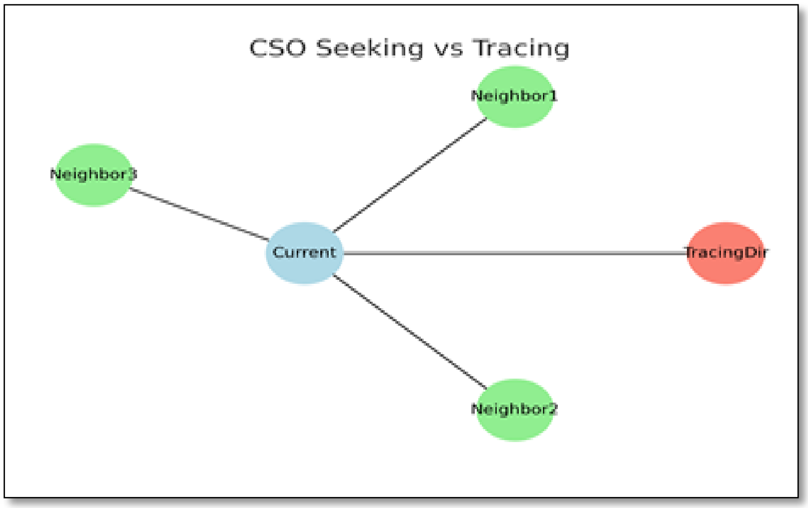

Figure 4. CSO seeking vs tracing in permutation neighborhoods.

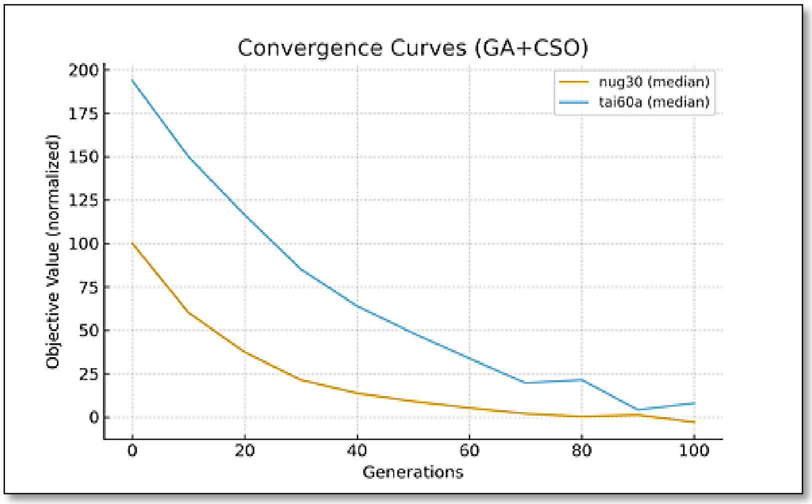

Figure 5. Convergence curves on nug30 and tai60a (median over runs).

4. Discussion

In the formal experiment, the results showed that Cat Swarm Optimization (CSO)'s dual-mode structure is very important to enhance the performance of our proposed hybrid framework GA+CSO. It is because CSO's manner of seeking and tracking enables an algorithm to strike a balance between global exploration at the service of another algorithm and its own exploitation to prevent premature stagnation.

The integrated Cat Swarm Optimization (CSO) model's respective modes are seeking mode (local intensification) and tracing mode (global exploration); during an iterative procedure, both are used to maintain the trade-off between exploration diversity and exploitation-intensification correctly. This synergistic effect allows the algorithm to be pulled out of local optimal regions more easily. Additionally, and more importantly, the influence of factors that tend to increase convergence time or cause early stagnation in stand-alone genetic algorithms is reduced. The complementary nature of GA and CSO together is what makes it effective. Because the former has a high ability for global recombination by way of crossover and mutation operations, while the latter affects local competition between superior regions in a search space through adaptive neighborhood movement transportation functions of each. This kind of coupling outperforms combinations lacking in coordination[23]. It gives diversification more force while headquarters has a good deal of higher-quality solutions concurrently, which brings with it strength across different instances for QAPLIB.

To verify whether the observed performance improvements are statistically significant, non-parametric statistical tests were performed and are summarized in result. On the proportions of (i) requests for duplicate linking and (ii) successful (and valid) linkage, respectively, z values can be read from the table: 0. Were they confused? Particularly confusing was the Wilcoxon signed-rank test between GA+CSO and the best reference voted for by the consecutive-0 method. (See Table 3). The test statistic (or score) is evaluated as $Z = -2.20$ with $p = 0.028 < 0.05$, which confirms that the apparent improvement is statistically significant at 5% level. Moreover, the Friedman rank test shows significant differences in aggregate (all comparisons) among different algorithms (χ^2 f_2 F_3 = 9.45, p = 0.024).

Out of the following, the Nemenyi post hoc analysis (CD 1.15, a 0. 05) showed that GA+CSO is significantly better than either G alone or C, while CSO alone has no statistically significant advantage over G(.) when compared to other methods such as GA-PSO, the differences were not great enough to be considered significant ($p < 0.05$). These findings show conclusively that this hybrid framework achieves higher and more stable performance across the benchmark instances on which it was tried, stemming from its design.

5. Conclusion

Based on a hybrid of Genetic Algorithm and Cat Swarm Optimization (GA–CSO), this research introduces a hybrid framework for solving Quadratic Assignment Problems (QAP), one of the most computationally hardest NP-hard combinatorial optimization problems. This approach synthesizes the global recombination strength of Genetic Algorithms with the dual intensification mechanism of Cat Swarm Optimization, while maintaining a balance between exploration and exploitation. The hybrid model consistently produced competitive solution quality, smaller gaps from optimality, and greater robustness than its stand-alone metaheuristic counterparts when applied to the standard QAPLIB benchmark instances. Results were validated statistically, showing that this increased performance was reliable. The framework proposed in this paper provides an efficient and flexible method for generating optimal solutions to permutation-based combinatorial search problems. Future working directions will be to define adaptive parameter control, embed learning-based operations, and develop it for use with large-scale industrial or business target optimization problems.

The experimental results demonstrate that the proposed hybrid GA+CSO method outperforms classical metaheuristic algorithms such as Genetic Algorithm (GA) and Simulated Annealing (SA) in terms of solution quality and convergence speed. The hybrid framework achieves lower optimality gaps and faster convergence across multiple benchmark instances.

Acknowledgment

My gratitude goes to the College of Science, University of Baghdad, and its Mathematics Department for their help throughout the research that led to this publication. I would also like to thank everyone from the principal investigator down to the building maintenance staff who helped with their efforts.

Conflict of Interest

The author declares that there are no conflicts of interest related to this study.

Funding

This research was conducted with personal funding by the author. The author gratefully acknowledges the University of Baghdad, College of Science, Department of Mathematics, for their academic support

References

1. Burkard RE, Karisch SE, Rendl F. QAPLIB: a quadratic assignment problem library. J. Glob. Optim. 1997;10(4):391–403. https://doi.org/10.1023/A:1008293323270
2. Anstreicher KM. Recent advances in the solution of quadratic assignment problems. Math. Program. 2003;97(1):27–42. https://doi.org/10.1007/s10107-003-0437-z
3. Marrouche W. An evolutionary algorithm tailored to the quadratic assignment problem, Evol. Intell. 2025;18(6):112. https://doi.org/10.1007/s12065-025-01100-3.
4. Harris M, Berretta R, Inostroza-Ponta M, Moscato P. A memetic algorithm for the quadratic assignment problem with parallel local search. In: Proc IEEE Congr. Evol. Comput. (CEC); 2015: 838–845. https://doi.org/10.1109/CEC.2015.7256978
5. Bouzidi A, Riffi ME, Barkatou M. Cat swarm optimization for solving the open shop scheduling problem. J. Ind. Eng. Int. 2019;15(2):367–378. https://doi.org/10.1007/s40092-018-0297-z
6. Sattar HA, Cheetar A, Tareq I. A new strategy based on GSABAT to solve single objective optimization problem. Int. J. Swarm Intell. Res. 2019;10(3):1–22. http://dx.doi.org/10.4018/ijsir.2019070101
7. Ahmed AM, Rashid TA, Saeed SA. Dynamic cat swarm optimization algorithm for backboard wiring problem. Neural Comput. Appl. 2021;33(20):13981–13997. https://doi.org/10.1007/s00521-021-06041-3

8. Houssein EH, Saeed MK, Hu G, Al-Sayed MM. Metaheuristics for solving global and engineering optimization problems: review, applications, open issues and challenges. Arch. Comput. Methods Eng. 2024;31(8):4485–4519. https://doi.org/10.1007/s11831-024-10168-6
9. Abdulsattar R, Abbas IT. Tabu search algorithm for solving quadratic assignment problem. AIP Conf. Proc. 2024;3097(1).https://doi.org/10.1063/5.0209862
10. Tan Z , Mu Y . Learning solution-aware transformers for efficiently solving quadratic assignment problem. [preprint]. arXiv. 2024:2406.09899. https://doi.org/10.48550/arXiv.2406.09899.
11. Sohrabi M, Fathollahi-Fard AM, Gromov VA, Dulebenets MA. A genetic engineering algorithm for the generalized quadratic assignment problem. Neural Comput. Appl. 2025;37(18):12253–12279. https://doi.org/10.1007/s00521-025-11155-z
12. Elaibi WM, Rahi AK, Majeed RK, Alhasan AA. A branch-and-bound algorithm for non-integer linear programs with fuzzy right-hand side coefficients. Ind. Eng. Manag. Syst. 2025;24(2):225–233. https://doi.org/10.7232/iems.2025.24.2.225.
13. Anka F, Aghayev N. Advances in sand cat swarm optimization: a comprehensive study. Arch. Comput. Methods Eng. 2025;32: 2669–2712. https://doi.org/10.1007/s11831-024-10217-0
14. Christiansen J, Smith-Miles K. Instance space analysis for the quadratic assignment problem. [preprint]. arXiv. 2025:2506.20172. https://doi.org/10.48550/arXiv.2506.20172.
15. Liu Z, Tang J, Jia H, Huo J, Chen J. Dual-population co-evolution for path planning using neuro-fuzzy systems. Memet. Comput. 2026;18(1):14. https://doi.org/10.1007/s12293-025-00492-0
16. Boudjemline A, Chaudhry IA, Rafique AF, Elbadawi IA, Aichouni M, Boujelbene M. Multi-objective flexible job shop scheduling using genetic algorithms. The. Vjesn. 2022;29(5):1706–1713. https://doi.org/10.17559/tv-20211022164333.
17. Burmeister SC, Guericke D, Schryen G. A memetic NSGA-II for the multi-objective flexible job shop scheduling problem with real-time energy tariffs. Flex. Serv. Manuf. J. 2024;36(4):1530–1570. https://doi.org/10.1007/s10696-023-09517-7
18. Wu R, Luo E, Li X, Tang H, Li Y. Hybrid artificial bee colony algorithm with Q-learning for distributed heterogeneous flexible job shop scheduling problem considering machine preventive maintenance. Swarm Evol. Comput. 2025; 98:102134. https://doi.org/10.1016/j.swevo.2025.102134
19. Karacan I, Senvar O, Bulkan S. A Novel parallel simulated annealing methodology to solve the no-wait flow shop scheduling problem with earliness and tardiness objectives. Processes. 2023;11(2):454. https://doi.org/10.3390/pr11020454
20. Petropoulos F, Akkermans H, Aksin OZ, Ali I, Babai MZ, Barbosa-Povoa A. Operations and supply chain management: principles and practice. Int J Prod Res. 2026;64(1):330–513. https://doi.org/10.1080/00207543.2025.2555531.
21. Alkabbani H, Ahmadian A, Zhu Q, Elkamel A. Machine learning and metaheuristic methods for renewable power forecasting: a recent review. Front. Chem. Eng. 2021;3:665415. https://doi.org/10.3389/fceng.2021.665415
22. Panwar K, Rajwar K, Deep K, Cho SB. A swarm intelligence-based hybrid metaheuristic with tabu search for the quadratic assignment problem. J. Supercomput. 2026;82(3):126. https://doi.org/10.1007/s11227-026-08258-2
23. Danach K, Harb H, Hejase HJ, Saker L. A hybrid metaheuristic framework with reinforcement learning–based heuristic selection for large-scale combinatorial optimization. Eur J Pure Appl. Math 2025;18(3):6602. https://doi.org/10.29020/nybg.ejpam.v18i3.6602
24. Omidvar MN. Cooperative co-evolutionary algorithms for large-scale optimization [dissertation]. Melbourne: RMIT University; 2024. https://doi.org/10.25439/rmt.27580074
25. Wang J, Tan Y, Sun C. An improved NSGA-III algorithm for solving high-dimensional many-objective flexible job shop scheduling problem. In: International Conference on Optimization and Learning. Cham: Springer Nature Switz; 2025:150–163.https://doi.org/10.1007/978-3-032-13589-6_11