INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

MMCTS: AN ADAPTIVE HYBRID MMAS–MCTS FRAMEWORK FOR THE TRAVELING SALESMAN PROBLEM

Ahmad Hilal Al-Kurdi
Al-Shamal Private University,
Faculty of Informatics Engineering,
Sarmada, Syria.

Mariam Zmirly
Sham University, Faculty of Engineering,
Department of Informatics,
Aizaz, Syria.

Abstract:

The traveling salesman problem (TSP) is an NP-hard problem with significant theoretical and practical implications. This paper presents a hybrid MMAS–MCTS algorithm for the Traveling Salesman Problem that incorporates an adaptive last-mile intensification mechanism to improve solution quality near convergence. The proposed framework integrates a stagnation-aware MCTS destroy-and-repair strategy with regret-based reconstruction and progressive widening to enhance exploration while preserving promising solution structures. As the search converges, adaptive 2-opt and lightweight 3-opt refinements are employed to strengthen exploitation and further improve solution quality.

Experimental evaluations on TSPLIB benchmark instances demonstrate the effectiveness of MMAS–MCTS, which achieves optimal solutions for many instances and maintains very small optimality gaps on larger problems. Comparative analyses further show that the proposed approach consistently outperforms conventional MMAS and MMAS augmented with 2-opt/3-opt local search in terms of solution quality and robustness.

Keywords: Traveling Salesman Problem. Max–Min Ant System. Monte Carlo Tree Search. Hybrid Metaheuristics. Adaptive Last-Mile Intensification. Destroy–Repair Search. Local Search.

Introduction

The traveling salesman problem (TSP) [1- 4] is a classical combinatorial optimization problem. It describes a scenario which a salesman needs to visit N cities, each exactly once, and finally return to the city where they originated to find the optimal path.

The TSP is often used as a model for real-world problems such as logistics route planning [5-7], circuit board manufacturing [8-9], and genetic sequencing [10-12], among others,

Due to its wide range of practical applications and NP-hard computational complexity, the Traveling Salesman Problem (TSP) remains one of the most extensively studied combinatorial optimization problems [3–14]. As the number of cities increases, the search space grows exponentially, making the

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

problem increasingly difficult to solve optimally [3,13]. Consequently, a wide variety of solution approaches have been developed, including exact algorithms, approximation methods, heuristics, and metaheuristic techniques [3 –14].

Exact algorithms [3,2], such as dynamic programming and branch-and-bound methods, exhibit exponential growth in computational requirements as the problem size increases. Approximation algorithms [15,16] aim to generate near-optimal solutions with provable performance guarantees. Nevertheless, this computational efficiency is achieved at the expense of optimality, and the solutions obtained are not guaranteed to be optimal [15,16]. Comprehensive surveys on the approximability of TSP variants are provided in [17].

In contrast to exact and approximation algorithms, heuristic algorithms [18,19,20] and metaheuristic algorithms [19,21,22] emphasize computational efficiency and are capable of generating near-optimal solutions within acceptable computational times, making them particularly suitable for large-scale TSP instances.

Among the most widely used heuristic methods for the TSP are nearest-neighbor and insertion strategies [20], as well as the Lin–Kernighan local search heuristic [23,24]. In addition, several metaheuristic techniques have been widely investigated for solving the TSP, including Genetic Algorithm (GA) [24-25], Particle Swarm Optimization (PSO) [27-28], Simulated Annealing (SA) [29-30], Tabu Search (TS) [31-32] and Ant Colony Optimization (ACO) [33].

Ant Colony Optimization (ACO) originates from observations of the foraging behavior of real ants. Ants deposit pheromone trails while moving between the nest and food sources, and these trails influence the path-selection decisions of other ants. The collective use of pheromone information enables the colony to gradually identify efficient routes, which inspired the development of ant-based optimization algorithms [34].

The first formal implementation of this concept was introduced through the Ant System (AS), where artificial ants iteratively construct solutions using pheromone information and problem-specific heuristic knowledge. The quality of generated solutions influences subsequent pheromone updates, allowing the search process to progressively favor promising regions of the solution space [35].

Several extensions were later proposed to improve the performance of the original AS. The Ant Colony System (ACS) enhanced the search process by introducing a state-transition mechanism and local pheromone updating strategy, resulting in improved exploration–exploitation balance and superior performance on TSP benchmarks [33]. Subsequently,

The MAX–MIN Ant System (MMAS) further improved the original framework by introducing upper and lower bounds on pheromone values and restricting pheromone updates to the best-performing solutions. These modifications helped limit search

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

stagnation and improved the stability of the algorithm across different problem instances [36].

The success of ACS [33] and MMAS [36] motivated the development of numerous ACO extensions and hybrid approaches. Many of these methods have been applied to increasingly large and complex combinatorial optimization problems, where improvements in solution quality and computational efficiency remain active research objectives [42].

Unlike previous hybrid ACO frameworks, the proposed MMCTS framework introduces a stagnation-aware MCTS destroy–repair mechanism integrated directly into the MMAS search process through pheromone feedback. Furthermore, an adaptive last-mile intensification strategy dynamically activates lightweight 3-opt refinements only when the search enters near-convergence regions, providing a balance between diversification and exploitation.

Literature Review

Recently, several researchers have presented hybrid algorithms to solve the traveling salesman problem

2.1 Recent Hybrid ACO Algorithms

- Hybridization has become one of the most successful strategies for improv ing the performance of ACO algorithms on large-scale TSP instances. Exis ting studies have combined ACO with evolutionary algorithms, local searc h procedures, simulated annealing, tabu search, and adaptive large neighbo rhood search techniques in order to improve the balance between explorati on and exploitation.
- Bai et al. (2023) [37] proposed a hybrid ACS-GA algorithm in which the Ant Colony System and Genetic Algorithm operate concurrently and exchange high-quality solutions during the search process. The method was designed to combine the exploitation strength of ACS with the diversification capability of GA, leading to improved solution quality and enhanced convergence behavior on TSP benchmark instances.
- Wu et al. (2025) [38] introduced an adaptive two-stage ACO-SA algorithm for the Traveling Salesman Problem. In the proposed framework, ACO is used to construct promising tours, while Simulated Annealing (SA) performs subsequent local refinement. The integration of the two methods enhanced search efficiency and improved the quality of the final solutions.
- Tian et al. (2024) [39] integrated Ant Colony Optimization (ACO) with the Sparrow Search Algorithm (SSA) to improve the exploration capability of the search process. By utilizing SSA to enhance population diversity and guide the search toward promising regions, the proposed framework reduced premature convergence and improved solution quality relative to conventional ACO.
- Wang and Han (2021) [40] investigated parameter optimization strategies for Ant Colony Optimization to reduce its

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

dependence on manually selected parameter values. Their approach employed auxiliary optimization mechanisms to improve pheromone initialization and automatically adjust key control parameters, resulting in more stable convergence and improved solution quality across TSP benchmark instances.

- Liu (2023) [41] proposed the Dynamic Adaptive Ant Colony Optimization (DAACO) algorithm. The method integrates convex hull analysis and K-means clustering to characterize the search space and dynamically regulate the ant population during the optimization process. This adaptive strategy improved search diversity and computational efficiency while maintaining competitive solution quality.

2.2 Research Gap

Despite the success of existing hybrid ACO approaches, most methods rely primarily on local search or auxiliary metaheuristics for intensification and diversification. Comparatively little attention has been devoted to integrating Monte Carlo Tree Search within MMAS as an adaptive stagnation-recovery mechanism. Furthermore, existing approaches rarely incorporate convergence-aware intensification strategies that dynamically adjust the search effort near high-quality regions. To address these limitations, this paper proposes the MMCTS framework, which combines MMAS, MCTS-based destroy–repair search, and adaptive last-mile intensification within a unified optimization process.

3. Background and Optimization Techniques

3.1 ACO-MMAS Framework

The proposed algorithm employs the Max–Min Ant System (MMAS) as the primary search engine. MMAS is an advanced variant of Ant Colony Optimization in which pheromone values are restricted within predefined upper and lower bounds [36,42], thereby preventing premature convergence and preserving search diversity.

At each iteration, a colony of ants constructs complete tours probabilistically according to the pheromone trails and heuristic visibility information [36,42]. The probability of moving from city i to city j is defined as

$$P^k_{ij} = \frac{\tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}}{\sum_{l \in N^k} \tau_{il}^{\alpha}\,\eta_{il}^{\beta}}$$

Where τij denotes the pheromone intensity, $\eta ij = 1/dij$ represents the heuristic information, α and β control the relative influence of pheromone and visibility, and Nk denotes the feasible neighborhood of ant k.

To reduce computational complexity, candidate lists composed of nearest-neighbor cities are employed during solution construction [36, 42]. After all ants generate tours, pheromone evaporation and reinforcement are performed according to the MMAS update mechanism [36,42].

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

Furthermore, multiple high-quality solutions participate in pheromone reinforcement through a rank-based weighting scheme inspired by Bullnheimer et al. [43]. The top-ranked ants deposit pheromone with contributions proportional to their ranking, thereby strengthening promising search directions while maintaining exploration capability and search diversity.

3.2 2-opt Local Search

To enhance the quality of newly generated solutions, a candidate-based 2-opt local search procedure is periodically applied to the iteration-best tour [44]. The objective of 2-opt is to remove edge crossings and reduce unnecessary detours by replacing two existing edges with two alternative edges that produce a shorter route.

Given two edges (a, b) and (c, d), the variation in tour length is computed as

$$\Delta = [d(a, c) + d(b, d)] - [d(a, b) + d(c, d)]$$

Where $d(\cdot)$ denotes the distance between two cities. Whenever $\Delta < 0$, the exchange is accepted and the intermediate segment is reversed. To reduce computational overhead, the search is restricted to candidate-neighbor cities rather than evaluating all possible edge combinations. Consequently, the procedure provides substantial solution improvements while preserving computational efficiency.

3.3 3-opt-lite Intensification

Although 2-opt efficiently removes most edge crossings, it often becomes trapped in local minima. Therefore, an additional 3-opt-lite intensification mechanism is activated when the search approaches high-quality regions.

The method removes three edges from the current tour and reconnects the resulting segments using a limited set of promising reconnection patterns [44,45,24]. Only the three most promising reconnection schemes commonly associated with edge-exchange improvements are evaluated, while the remaining configurations are discarded to reduce computational overhead.

In contrast to the classical 3-opt procedure, which exhaustively examines all feasible reconnections, the proposed variant restricts the search to candidate-neighbor structures and bounded segment lengths, thereby significantly reducing computational complexity while preserving solution quality.

The gain associated with a 3-opt move can be expressed as

$$Gain = L\text{current} - L\text{new}$$

Where Lcurrent and Lnew denote the tour lengths before and after the reconnection operation. The move is accepted whenever $Gain > 0$.

To detect near-convergence conditions, an adaptive intensification criterion is employed. The lightweight 3-opt procedure is activated when no improvement in the global

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

best solution is observed for $Tint$ consecutive iterations:

$$NoImprovement \geq T_{int}$$

Where $Tint$ represents the intensification threshold. This mechanism allows the algorithm to concentrate computational effort near convergence while avoiding unnecessary local search during early exploration stages.

By activating this procedure only when the solution quality approaches a predefined threshold, additional intensification is achieved without significantly increasing runtime.

3.4 MCTS-Based Destroy–Repair Search

To overcome stagnation, the proposed algorithm integrates a Monte Carlo Tree Search (MCTS) module [46-47]. Unlike traditional applications of MCTS in game-playing environments, the present implementation uses MCTS as a destroy–repair optimization operator that is activated only after a prolonged period without improvement.

The search tree consists of candidate tours generated through a set of neighborhood transformation operators. Three actions are considered:

1. Destroy–Repair.
2. Segment Shuffle.
3. Segment Swap.

Node selection follows the Upper Confidence Bound for Trees (UCT) policy [52].

$$UCT_i = Q_i + C\sqrt{\frac{\ln N}{n_i}}$$

Where Qi denotes the average reward of node i, N is the visit count of the parent node, ni is the visit count of the current node, and C controls the exploration–exploitation balance.

To limit excessive tree growth, Progressive Widening is employed:

$$m(N) = kN^{\alpha}$$

Where $m(N)$ is the maximum number of children that may be expanded [46,47], while k and α regulate the growth rate.

The destroy–repair operator removes a subset of cities associated with costly edges and subsequently reinserts them using a regret-based insertion heuristic inspired by the regret insertion strategies of Solomon [49] and later adaptive large neighborhood search frameworks [50].

The reward is computed as:

$$R = \frac{L_{\text{root}} - L_{\text{rollout}}}{L_{\text{root}}}$$

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

Where *L*root is the objective value of the root solution and *L*rollout is the objective value obtained after the rollout process.

During rollout, a partial candidate solution is further refined using a stochastic destroy–repair procedure followed by a candidate-based 2-opt improvement step.

The collected information is propagated backward through the tree according to

$$N_i \leftarrow N_i + 1$$

$$W_i \leftarrow W_i + R$$

$$Q_i = \frac{W_i}{N_i}$$

Thereby guiding future exploration toward promising search regions [46].

3.5 Proposed MMCTS Framework

The proposed framework combines the complementary strengths of population-based exploration, local improvement, and adaptive intensification within a unified optimization process.

Initially, MMAS performs global exploration and constructs candidate tours using pheromone-guided probabilistic decisions. The resulting iteration-best solution is subsequently refined through candidate-based 2-opt local search. When the search approaches high-quality regions, a lightweight 3-opt intensification stage is activated to further improve solution quality.

If no improvement is observed for a predefined number of iterations, the MCTS destroy–repair module is triggered. This component performs a focused exploration around the current best solution using stochastic tree search and adaptive neighborhood transformations. Any improved solution discovered by MCTS is immediately incorporated into the pheromone model through

$$\tau ij \leftarrow (1 - \rho)\tau ij + \Delta\tau ij$$

Following the pheromone update, the standard MMAS pheromone limits are enforced to prevent excessive trail accumulation and search stagnation. The updated pheromone values are bounded according to allowing the information obtained by MCTS to influence future solution construction by the ant colony [36,42].

$$\tau ij = \min(\tau\max, \max(\tau\min, (1 - \rho)\tau ij + \Delta\tau ij))$$

Where τmin and τmax denote the lower and upper pheromone bounds defined by t he MMAS framework.

Consequently, MMAS provides global exploration, 2-opt and 3-opt-lite perform local intensification, and MCTS serves as an adaptive diversification mechanism for escaping stagnation. The interaction among these components creates a balanced search process capable of achieving both solution quality and robustness across different TSPLIB instances.

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

3.6 MMCTS Algorithm

Algorithm 1 summarizes the overall workflow of the proposed MMCTS framework.

```
Initialize pheromone trails
Generate candidate lists
BestGlobal ← ∞
while stopping criterion not met do
    Construct solutions using MMAS
    Apply candidate-based 2-opt
    Update iteration-best solution
    if IterBest < BestGlobal then
        BestGlobal ← IterBest
        Reset stagnation counter
    else
        Increase stagnation counter
    end if
    Update pheromone trails
    Apply MMAS pheromone bounds
     if NoImprovement ≥ T_int then
        Apply 3-opt-lite
     end if
     if NoImprovement ≥ T_stag then
        Execute MCTS destroy-repair
        Update pheromone trails
     end if
end while return
BestGlobal
```

4. Methodology

4.1 Overall Methodology

The proposed MMCTS algorithm follows a multi-stage optimization methodology that integrates global exploration, local exploitation, adaptive diversification, and convergence-oriented intensification within a unified optimization process. The framework combines the exploration capability of the Max–Min Ant System (MMAS) with the adaptive search ability of Monte Carlo Tree Search (MCTS), while lightweight local search operators are selectively employed to improve solution quality. The overall methodology of the proposed framework is illustrated in Figure 1.

The optimization process starts by reading the TSPLIB instance and constructing the corresponding distance matrix. Candidate lists are then generated to restrict neighborhood exploration and reduce computational complexity during solution construction. Subsequently, the pheromone matrix is initialized according to the MMAS strategy using an initial solution generated by the nearest-neighbor heuristic. These initialization steps provide the information required for the ant colony to begin the optimization process efficiently.

During each iteration, all ants construct complete tours according to the MMAS probabilistic transition rule based on pheromone intensity and heuristic visibility. The best solution obtained in the current iteration is refined using a candidate-based 2-opt local search before updating the global best solution whenever an improvement is

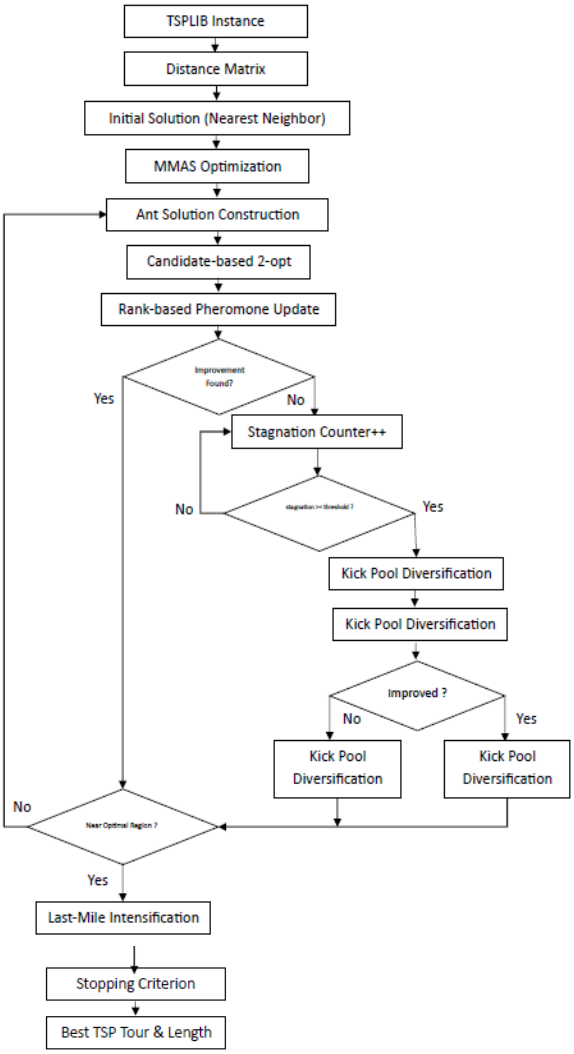
INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

achieved. Afterwards, rank-based pheromone reinforcement is performed, followed by the application of the standard MMAS pheromone bounds in order to preserve the balance between exploration and exploitation.

Whenever the search stagnates for a predefined number of consecutive iterations, the optimization process activates the MCTS-based destroy–repair module. The MCTS procedure performs stochastic tree search using destroy–repair, segment shuffle, and segment swap operators to explore alternative solution structures around the incumbent solution. Any improved solution produced by MCTS is immediately incorporated into the pheromone model, allowing subsequent MMAS iterations to exploit the newly discovered search information.

As the optimization approaches near-convergence regions, an Adaptive Last-Mile Intensification strategy is activated. Instead of applying computationally expensive refinement throughout the optimization process, additional search effort is concentrated only during the final optimization stage. This phase enlarges the candidate neighborhood, increases the MCTS computational budget through additional simulations and deeper search, and performs lightweight 3-opt refinement followed by candidate-based 2-opt cleanup. Consequently, the proposed framework achieves improved solution quality while maintaining computational efficiency throughout the search process.

The complete optimization procedure is summarized in Algorithm 1, whereas the interaction among the major optimization components is illustrated in Figure 1.



5. Experimental setup and result analysis

5.1 Running Environment and Parameters

This section presents the comparative experimental results and analysis of the proposed MMAS-MCTS(Max–Min Ant System with Monte Carlo Tree Search), and

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

some other algorithms on the basis of the TSP instance set.

Experimental Environment: The algorithms MMAS-opt2-opt3 and MMAS-MCTS are all written in MATLAB R2019a. The operating environment is Windows11X64 with Intel(R) Core (TM) i5-12500H processor and 16 GB RAM.

The main parameter settings in these algorithms are shown in Table 1. Each instance runs 30 iterations, and each iteration is run 1000 times.

Table 1 : Main parameter settings of the proposed algorithm

Parameter	Value
α	1.2
β	5
ρ	0.2
Candidate list size	35
pbest	0.05
MCTS simulations	900
MCTS depth	4
UCT constant	1.3
Destroy fraction	0.18
Repair multistart	10
Stagnation limit	60

Parameter	Value
α	1.2
β	5.0
ρ	0.20
Q	1.0
K_{cand}	35
p_{best}	0.05
Maximum iterations	1000
Number of ants	Adaptive
Stagnation limit	60
MCTS simulations	900
MCTS depth	4
UCT constant (C_p)	1.30
Destroy fraction	0.18
Repair multistart	10
Progressive widening k	1.8
Progressive widening α	0.55
2-opt	Enabled
3-opt-lite	Enabled
Deep restart threshold	4

Table 2: Set the parameters of each algorithm

Component	Parameter	Value
MMAS	Pheromone importance (α)	1.2
MMAS	Heuristic importance (β)	5.0
MMAS	Evaporation rate (ρ)	0.20
MMAS	Pheromone de po	1.0

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

	sit fac tor (Q)	
MMAS	Candidate list size (K_{cand})	35
MMAS	p_{best}	0.05
MMAS	Maximum iterations	1000
MMAS	Number of ants	Adaptive according to problem size
Local Search	2-opt	Enabled
Local Search	2-opt frequency	Every iteration (small/medium) or every 2–4 iterations (large instances)
Local Search	2-opt budget	Adaptive ($\leq 8n$)
Local Search	3-opt-lite	Enabled
Local Search	3-opt activation threshold	$1.08 \times OPT$
Stagnation Control	Stagnation limit	60 iterations
Stagnation Control	Smoothing factor (λ)	0.18
MCTS	Number of simulations	900
MCTS	Maximum depth	4
MCTS	UCT exploration constant (Cp)	1.30
MCTS	Rollout steps	1
MCTS	Destroy fraction	0.18
MCTS	Repair multistart	10
MCTS	Progressive widening coefficient (k)	1.8
MCTS	Progressive widening exponent (αpw)	0.55
MCTS	Maximum children	30
MCTS	Reward mode	Relative
Destroy–Repair	Segment shuffle fraction	0.06
Destroy–Repair	Segment swap fraction	0.04
Deep Restart	Restart threshold	4 consecutive MCTS failures
Deep Restart	Restart factor	0.85

To test the effectiveness of the experiments will be conducted using TSP instance of different sizes from the TSPLIB database, which are arranged as follows:

In order to illustrate the performance of MMAS-MCTS for different scales of TSP, TSP instances of scales from 16 to 1304 were selected from TSPLIB for resolving.

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

The results were provided in Table 2. From the table, the best objective value from 30 replications for each instance is listed in column four, and its optimality gap (in column five) was computed (in percentage) from the known optimal value (from column three).

Table 3: Results of the objective values and average time

Instance	Number of Cities	Optimal	Best Value Gap (%)		Standard Deviation	Average Time(sec)
a280	280	2579	2579	0	0	96.426
att48	48	10628	10628	0	0	3.3462
berlin52	52	7542	7542	0	0	0.098
burma14	14	3323	3323	0	0	0.011
ch130	130	6110	6110	0	0	14.1095
d198	198	15780	15781	0.006337	0.36158	201.8262
eil51	51	426	426	0	0	2.80236
fl417	417	11861	11863.13	0	0.031711	33.62686
gil262	262	2378	2378	0	0.144952	39.4699
gr202	202	40160	40160	0	0.091059	97.83114
lin318	318	42029	42029	0	0.260625	39.4699
p654	654	34643	34683.9	0.063505	0.036619	907.5
rl1304	1304	252948	254435.5	0.154577	0.292434	2836.62
ulysses16	16	6859	6859	0	0	0.012
ulysses22	22	7013	7013	0	0	0.018
st70	70	675	675	0	0	1.2
pr76	76	108159	108159	0	0	1.4
kroA100	100	21282	21282	0	0	0.5
lin105	105	14379	14379	0	0	8.2
rat195	195	2323	2323	0	0.22568	22.1
pr226	226	80369	80369	0	0	30.4

From the results, the hybrid MMAS algorithm was able to solve medium sizes instances to reach the optimal solution.

The optimality gaps for most instances were within 1%. Column six showed the standard deviation of the objective values from 30 replications in percentage. The deviations were less than 1% in all tested instances, which implied consistency of the algorithm. The last column in the table was the average solution time (in seconds) for each instance.

Table 4: Results of the average gaps

	Average Gap (%)		
Instance	Hybrid	2-opt,3-opt	MMAS
a280	0	1.339667	2.873207
att48	0	0.382991	0.550903
berlin52	0	1.205914	1.115752
burma14	0	0.019561	0.019561
ch130	0	1.654664	2.067921
d198	0.097275	0.665399	3.324461
eil51	0	0.997653	0.446009
fl417	0.020234	0.849844	5.411854
gil262	0.136669	1.858705	2.546257

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

gr202	0.061006	2.292829	5.183516
lin318	0.25756	2.339694	3.338766
p654	0.118061	1.466963	10.43068
rl1304	0.587789	2.573691	10.78981
ulysses16	0	0	0
ulysses22	0	0	0
st70	0	1.496296	1.288889
pr76	0	0.996866	2.334711
kroA100	0	0.792219	1.158961
lin105	0	0.836984	1.10265
rat195	0.28842	1.515282	1.183814
pr226	0	0.835023	2.704899

To compare the performance of the hybrid MMAS algorithm with the MMAS- 2-opt,3-opt heuristic and the traditional MMAS algorithm, additional experiments were performed for those methods by using the same settings.

The average optimality gaps from each method were reported in Table III. From the results, the hybrid MMAS algorithm outperformed the MMAS- 2-opt,3-opt heuristic and the traditional MMAS algorithm in almost all cases.

The proposed MMAS-MCTS algorithm was compared with [37] and [51], where MMAS-MCTS had better results.

Figure 4-5 show the routes of 70,226 cities generated by MMAS, MMAS, opt-2, opt-3, and hybrid algorithms, respectively.

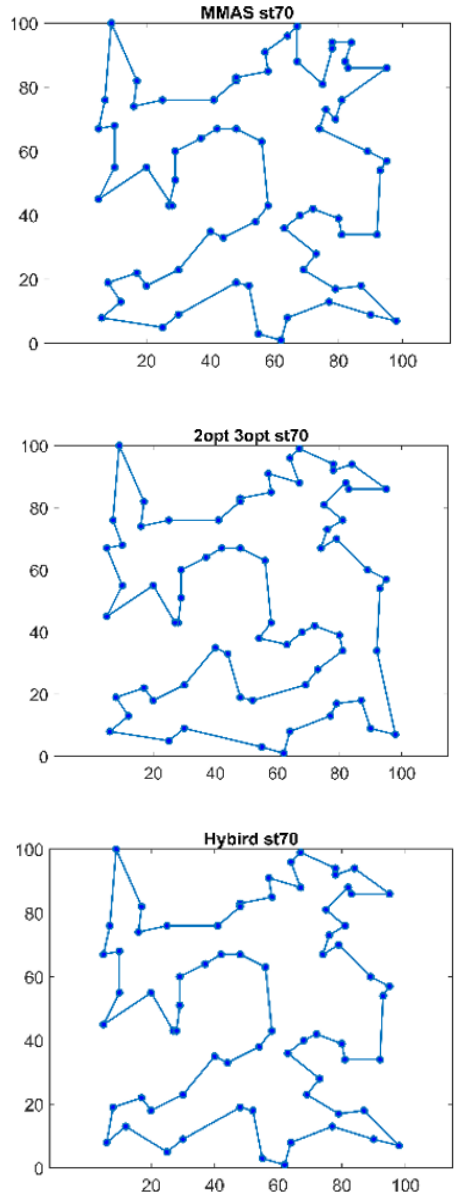


Fig 4: Algorithm MMAS, MMAS, opt-2, opt-3, and hybrid tours for 70 cities

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

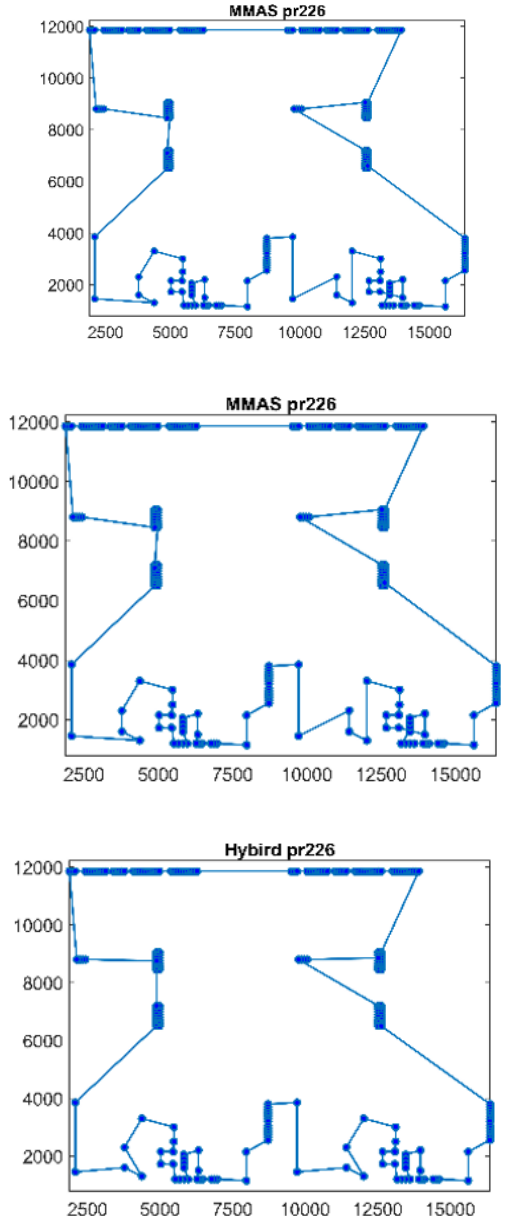


Fig 5: Algorithm MMAS, MMAS, opt-2, opt-3, and hybrid tours for 226 cities

In this paper, we developed a hybrid MCACS algorithm, conducted several numerical experiments, and compared the results with the [37] and [51] algorithms.

Conclusion

This paper presented MMCTS, a hybrid MMAS–MCTS framework for the Traveling Salesman Problem that combines pheromone-guided exploration, MCTS-based stagnation recovery, and adaptive last-mile intensification. Experimental results on TSPLIB benchmark instances demonstrated that the proposed approach consistently achieved optimal or near-optimal solutions with very small optimality gaps and high robustness across different problem sizes. The integration of MCTS destroy–repair search and adaptive 2-opt/3-opt-lite refinement effectively improved solution quality and outperformed conventional MMAS and MMAS enhanced with local search techniques.

References

1. Little, J.D., Murty, K.G., Sweeney, D.W., Karel, C.: An algorithm for the traveling salesman problem. Oper. Res. 11(6), 972–989 (1963)
2. Gutin, G., & Punnen, A. P. (Eds.). (2006). The traveling salesman problem and its variations (Vol. 12). Springer Science & Business Media.
3. Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton (2011)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

4. Y. Hu, Z. Zhang, Y. Yao, X. Huyan, X. Zhou and W.S. Lee, A bidirectional graph neural network for traveling salesman problems on arbitrary symmetric graphs. Eng. App. Artif. Intell. 97 (2021) 104061.
5. Crişan, G. C., Pintea, C. M., & Palade, V. (2017). Emergency management using geographic information systems: application to the first romanian traveling salesman problem instance. Knowledge and Information Systems, 50(1), 265-285.
6. J. Li, M. Liu and P. Liu, Route optimization of multi-vehicle cold chain logistics for fresh agricultural products. J. China Agr. Univ. 26 (2021) 115.
7. [7] Lu, Z., Wu, K., Bai, E., & Li, Z. (2025). Optimization of multi-vehicle cold chain logistics distribution paths considering traffic congestion. Symmetry, 17(1), 89.
8. H. Katagiri, G. Qingqiang, W. Bin, T. Muranaka, H. Hamori and K. Kato, Path optimization for electrical pcb inspections with alignment operations using multiple cameras. Proc. Comput. Sci. 60 (2015) 1051–1060.
9. Al-Janan, D. H., & Liu, T. K. (2016). Path optimization of CNC PCB drilling using hybrid Taguchi genetic algorithm. Kybernetes, 45(1), 107-125.
10. Monroe, J. G., Allen, Z. A., Tanger, P., Mullen, J. L., Lovell, J. T., Moyers, B. T., ... & McKay, J. K. (2017). TSPmap, a tool making use of traveling salesperson problem solvers in the efficient and accurate construction of high-density genetic linkage maps. BioData mining, 10(1), 38.
11. Z. Liu, H. Lou, K. Xie, H. Wang, N. Chen, O.M. Aparicio, M.Q. Zhang, R. Jiang and T. Chen, Reconstructing cell cycle pseudo time-series via single-cell transcriptome data. Nat. Commun. 8 (2017) 22.
12. Nałęcz-Charkiewicz, K., & Nowak, R. M. (2022). Algorithm for DNA sequence assembly by quantum annealing. BMC bioinformatics, 23(1), 122.
13. Lawler, E. L. (1985). The traveling salesman problem: a guided tour of combinatorial optimization. Wiley-Interscience Series in Discrete Mathematics.
14. Alkhalifa, R., Alkhomayes, F., Almazroua, B., Alhaidan, D., Alothman, M., & Almuhaidib, J. (2025). A comparative review of parallel exact, heuristic, metaheuristic, and hybrid optimization techniques for the traveling salesman problem. arXiv preprint arXiv:2505.18278.
15. Williamson, D. P., & Shmoys, D. B. (2011). The design of approximation algorithms. Cambridge university press.
16. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., & Protasi, M. (1999). Complexity and Approximation: Combinatorial

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

Optimization Problems and Their Approximability Properties. Springer.

17. Saller, S., Koehler, J., & Karrenbauer, A. (2025). A survey on approximability of traveling salesman problems using the TSP-T3CO definition scheme. Annals of Operations Research, 351(3), 2129-2190.
18. Rego, C., Gamboa, D., Glover, F., & Osterman, C. (2011). Traveling salesman problem heuristics: Leading methods, implementations and latest advances. European journal of operational research, 211(3), 427-441.
19. Marinakis, Y. (2024). Heuristic and metaheuristic algorithms for the traveling salesman problem. In Encyclopedia of optimization (pp. 1-12). Cham: Springer International Publishing.
20. Rosenkrantz, D. J., Stearns, R. E., & Lewis, II, P. M. (1977). An analysis of several heuristics for the traveling salesman problem. SIAM journal on computing, 6(3), 563-581.
21. Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. Information sciences, 237, 82-117.
22. Talbi, E. G. (2009). Metaheuristics: from design to implementation. John Wiley & Sons.
23. Helsgaun, K. (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. European journal of operational research, 126(1), 106-130.
24. Coulom, R. (2006, May). Efficient selectivity and backup operators in Monte-Carlo tree search. In International conference on computers and games (pp. 72-83). Berlin, Heidelberg: Springer Berlin Heidelberg.
25. Potvin, J. Y. (1996). Genetic algorithms for the traveling salesman problem. Annals of Operations Research, 63, 337–370.
26. Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review, 13(2), 129–170.
27. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 5, p. 4104–4108 (1997)
28. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization:An overview. Swarm Intell. 1, 33–57 (2007)
29. Bertsimas, D., Tsitsiklis, J.: Simulated annealing. Statist. Sci. 8(1),10–15 (1993)
30. Corana, A., Marchesi, M., Martini, C., Ridella, S.: Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm-corrigenda for this article is available here. ACM Trans. Math. Softw. (TOMS) 13(3), 262–280 (1987)

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

31. Glover, F., Laguna, M.: Tabu search. In: Du, D.Z., Pardalos, P.M.(eds.) Handbook of Combinatorial Optimization. Springer, Boston (1998)
32. Glover, F.: Tabu search–part ii. ORSA J. Comput. 2(1), 4–32 (1990)
33. Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. BioSystems, 43(2), 73–81.
34. A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," Proceedings of the First European Conference on Artificial Life,pp. 134–142, 1991.
35. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE transactions on systems, man, and cybernetics, part b (cybernetics), 26(1), 29-41.
36. Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. Future generation computer systems, 16(8), 889-914.
37. Bai, Z., Snášel, V., Vo, B., Kong, L., & Wang, X. (2023, October). A hybrid algorithm acs-ga for solving the traveling salesman problem. In International Conference on Genetic and Evolutionary Computing (pp. 71-82). Singapore: Springer Nature Singapore.
38. Wu, Y., Wang, H., Li, M., Tan, H., Wang, D., & Sheng, M. (2025). The Adaptive two-stage ant colony simulated annealing algorithm for solving the traveling salesman problem. RAIRO-Operations Research, 59(2), 1199-1213.
39. Tian, Y., Zhang, J., Wang, Q., Liu, S., Guo, Z., & Zhang, H. (2024). Application of hybrid algorithm based on ant colony optimization and sparrow search in UAV path planning. International Journal of Computational Intelligence Systems, 17(1), 286.
40. Wang, Y., & Han, Z. (2021). Ant colony optimization for traveling salesman problem based on parameters optimization. Applied Soft Computing, 107, 107439.
41. Liu, H. DAACO: adaptive dynamic quantity of ant ACO algorithm to solve the traveling salesman problem. Complex & Intelligent Systems. 9, 4317–4330 (2023).
42. Dorigo, M., & Stützle, T. (2004). Ant Colony Optimization. MIT Press.
43. Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A New Rank Based Version of the Ant System: A Computational Study. Central European Journal for Operations Research and Economics, 7(1), 25–38.
44. Lin, S. (1965). Computer solutions of the traveling salesman problem. Bell System Technical Journal, 44(10), 2245-2269.
45. Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. Operations research, 21(2), 498-516.

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 2, No 05, 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-748 PAGE NO: 016-033

46. Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., ... & Colton, S. (2012). A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games, 4(1), 1-43.

47. Chaslot, G. M. J., Winands, M. H., Herik, H. J. V. D., Uiterwijk, J. W., & Bouzy, B. (2008). Progressive strategies for Monte-Carlo tree search. New Mathematics and Natural Computation, 4(03), 343-357.

48. Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research, 35(2), 254-265.

49. Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science, 40(4), 455-472.

50. Kocsis, L., & Szepesvári, C. (2006, September). Bandit based monte-carlo planning. In European conference on machine learning (pp. 282-293). Berlin, Heidelberg: Springer Berlin Heidelberg.

51. Janjarassuk, U. (2024, May). Hybrid Ant Colony Optimization Method for the Traveling Salesman Problem. In 2024 9th International Conference on Business and Industrial Research (ICBIR) (pp. 292-295). IEEE.