State-of-the-Art QAP Solvers and Extensions for Agent-Task Matching

Table of Contents

1. Introduction

The Quadratic Assignment Problem (QAP) is widely recognized as one of the most challenging combinatorial optimization problems in operations research. Originally formulated to address the optimal assignment of facilities to locations while minimizing the total cost based on inter-facility flows and inter-location distances, QAP has significant theoretical and practical implications [3] . Its applications span from facility layout design and electronic module placement to hospital room assignment and beyond [1] . As modern computing and artificial intelligence methods evolve, state-of-the-art solvers now integrate exact methods, heuristic and metaheuristic approaches, as well as machine learning strategies to tackle the enormous complexity inherent in large-scale QAP instances.

In this article, we provide a comprehensive analysis of state-of-the-art QAP solvers, with an emphasis on recent breakthroughs that integrate classical optimization techniques with advanced machine learning methods. We benchmark the effectiveness of exact solvers such as Gurobi, innovative metaheuristics exemplified by Hexaly's novel formulation, and hybrid methods that combine parallel and localized search strategies. Furthermore, emerging learning-based approaches—for example, the Solution Aware Transformer (SAWT)—are examined in depth, illustrating how reinforcement learning can be leveraged to improve initial solutions iteratively. Through detailed analysis of experimental performance data from the QAPLIB benchmark, we not only compare solver performance but also discuss extensions to related problems such as agent–task matching in AI systems.

The remainder of this article is organized as follows. Section 2 lays the theoretical foundation of QAP. Section 3 reviews state-of-the-art algorithms that solve QAP using exact, metaheuristic, and hybrid strategies. Section 4 discusses machine learning approaches with detailed case studies of Hexaly's model versus Gurobi and the SAWT framework. Section 5 describes benchmark datasets and provides a comparative performance analysis. Section 6 extends the discussion to modern applications such as online QAP variants and agent–task matching, while Section 7 highlights important algorithmic strategies and scalability considerations. Finally, Section 8 summarizes key findings and suggests future research directions.

2. Fundamentals of the Quadratic Assignment Problem

The Quadratic Assignment Problem (QAP) involves the optimal assignment of a set of facilities to a set of locations. For each pair of facilities, a flow is specified that often represents the amount of goods or communications between them. Additionally, for each pair of locations, a distance value is given. The primary objective is to minimize the total cost defined as the weighted sum of distances between pairs of assigned facilities, where the weights are the corresponding flows [3]. Mathematically, the problem can be represented by either a Mixed-Integer Quadratic Programming (MIQP) formulation or a trace formulation, incorporating binary decision variables that indicate whether a facility is assigned to a particular location [1] [3]. Despite its seemingly straightforward definition, the quadratic nature of the objective function and combinatorial explosion of assignment possibilities make QAP a strongly NP-hard problem [1].

Given the complexity of QAP, a wide range of algorithmic strategies has been explored. Early studies focused on exact methods aimed at obtaining optimal solutions for small-size instances. However, as the instances scale up (for example, instances with up to 256 facilities in the QAPLIB benchmark [3]), exact methods quickly become impractical due to exponential growth in computational requirements. Consequently, researchers have developed several heuristic and metaheuristic methods, including Tabu Search, Simulated Annealing, and Genetic Algorithms, to derive high-quality approximate solutions in acceptable time frames [1] [3] . More recently, hybrid solvers and machine learning-based methods have further pushed the boundaries of performance, especially under constrained computing time.

Overall, the QAP remains a benchmark problem for evaluating the performance of new optimization algorithms, and its study has led to significant advances in both theory and practice.

3. State-of-the-Art QAP Algorithms

Solving the QAP has attracted research from numerous perspectives with a wide array of algorithmic techniques. In this section, we survey the prominent methods by classifying them into exact methods, metaheuristics, and hybrid algorithms.

3.1 Exact Methods

Exact methods for QAP are designed to guarantee the discovery of an optimal solution. Modern commercial solvers, such as Gurobi, frequently use MIQP models where a quadratic number of binary variables represent the assignment matrix, and the objective function is a polynomial with millions (or even billions) of terms for large instances [3] . Despite the robust branch-and-bound and cutting plane strategies incorporated by solvers like Gurobi 11.0, the performance deteriorates significantly with increased problem size. For instance, in comparative benchmarks, Gurobi has shown an average gap of 18.5% to the best-known solutions within a 1-minute runtime, and in the largest instance (involving 256 facilities), it was unable to find any feasible solution even after 1 hour of computation [3] .

Exact methods remain valuable for smaller problem instances and serve as a baseline for validating the performance of heuristic and learning-based approaches.

3.2 Metaheuristics and Heuristic Solvers

Metaheuristic approaches are a popular alternative to exact methods, especially for large and complex QAP instances. Techniques such as Tabu Search, Simulated Annealing, and Genetic Algorithms provide effective approximate solutions in a fraction of the time required by exact methods.

Tabu Search (TS):
Tabu Search exploits local search strategies and incorporates memory structures to avoid cycling and to escape local minima. Taillard's Robust TS, in particular, has proven effective by rapidly exploring the solution space and minimizing computational time [2] . Its computational efficiency and effectiveness make it highly attractive for QAP researchers.

Other Heuristics:
Heuristic methods such as SM, RRWM, and IPFP have been developed based on different paradigms like pairwise matching and iterative improvement. These methods are characterized by their speed in reaching sub-optimal solutions. For instance, in the context of QAPLIB instances, algorithms like SM and RRWM have been used extensively and offer competitive solution quality with very short execution times, albeit with larger solution gaps compared to more sophisticated strategies [1] .

3.3 Hybrid Algorithms

Hybrid algorithms combine the strengths of different methods to overcome limitations inherent to individual approaches. One notable hybrid approach is the Parallel Hybrid Algorithm (PHA), which integrates local search strategies like Tabu Search with global search and parallel processing capabilities [2] . By combining robust local search operators with parallel computing, PHA has been shown to achieve results within 0.05% of the best-known solutions on QAPLIB benchmark instances while maintaining competitive execution times [2] .

The parallel hybrid approach is particularly effective because it leverages the fast convergence of local search techniques and augments it with parallelism to handle computationally demanding evaluations across a large set of instances. This makes it an attractive candidate when solution quality and execution time are both critical.

4. Machine Learning Approaches to QAP

Traditional combinatorial optimization techniques have made significant progress in solving QAP; however, recent advances in machine learning have opened new avenues for tackling such problems with innovative approaches that learn and improve from iterative feedback.

4.1 Hexaly and Gurobi: A Comparative Benchmark

One of the breakthrough models in recent years is Hexaly, which rethinks the formulation of the QAP using a list variable that directly represents the permutation of facilities. In contrast to the classical MIQP model used by Gurobi, Hexaly's model is remarkably compact, defined in only four simple lines of code [3]. This innovative formulation significantly reduces the model complexity by representing the solution as a single list, rather than using a quadratic number of binary variables.

Experimental comparisons on the QAPLIB benchmark indicate that under a 1-minute running time, Hexaly achieves an average solution gap of just 1.1% compared to the state-of-the-art (SOTA) solutions, while Gurobi's gap is approximately 18.5% [3]. Furthermore, in large-scale instances with up to 256 facilities, Hexaly delivers near-optimal solutions with gaps as low as 0.4%, whereas Gurobi often fails to produce any feasible solution within a 1-hour window [3]. These results underscore the advantage of employing a more natural representation of the permutation in reducing computational burdens and improving solution quality.

The table below summarizes the key performance metrics for Hexaly 13.0 versus Gurobi 11.0 on a series of QAP instances:

Instance Size	Hexaly Gap (%)	Gurobi Gap (%)	Running Time (Hexaly)	Running Time (Gurobi)
12–19	0.1	0.2	Seconds	Minutes
20–30	0.7	4.0	Seconds	Minutes
31–50	1.5	12.8	Seconds	Minutes
51–80	1.6	23.5	Seconds	Minutes
81–256	1.5	65.4	Seconds	Hours

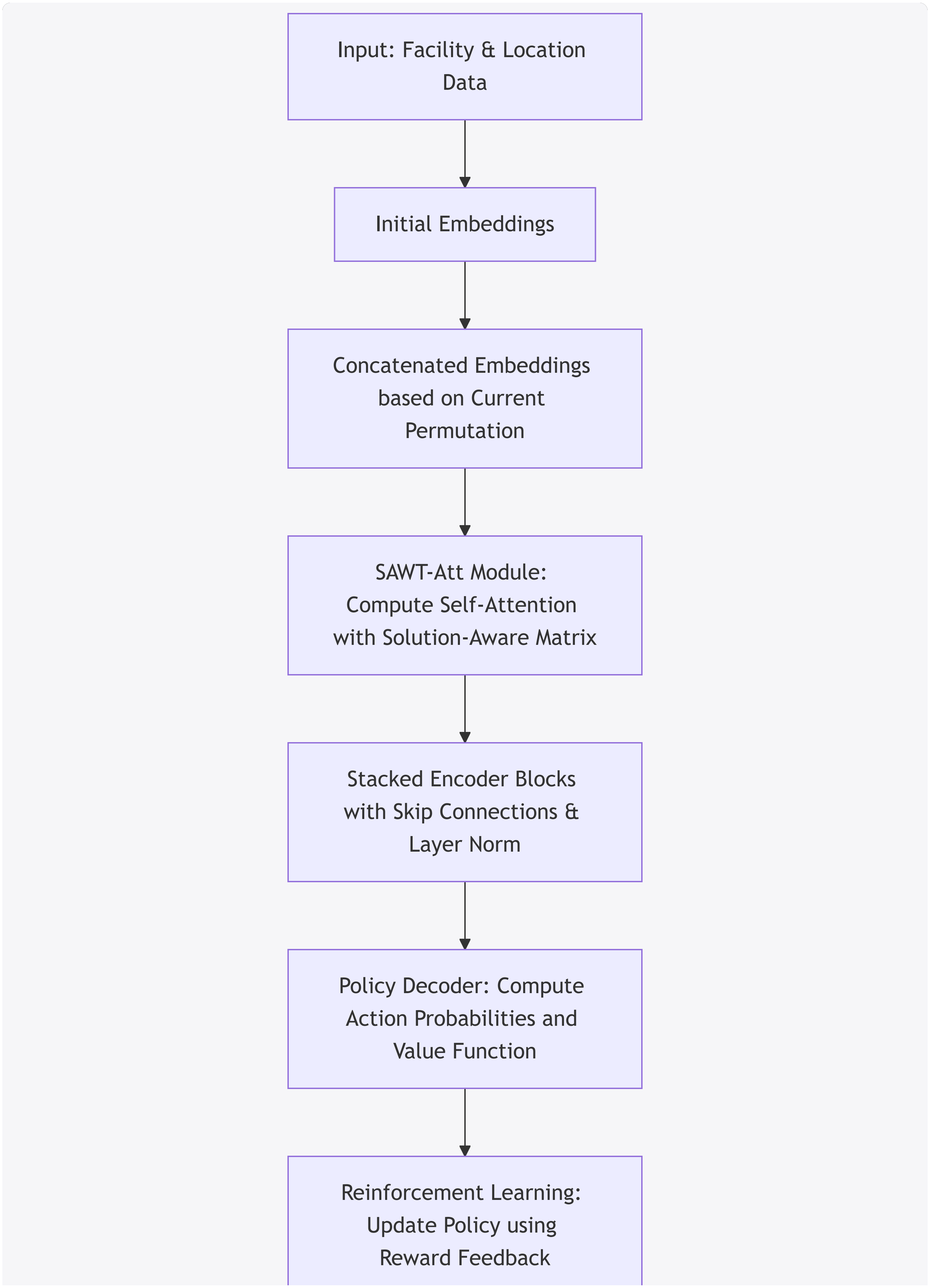
Table 1: Performance Comparison of Hexaly 13.0 and Gurobi 11.0 on QAPLIB Benchmark Instances [3]

4.2 Learning Solution-Aware Transformers (SAWT)

Another innovative approach in the field is the use of reinforcement learning and Transformer architectures to tackle the QAP. The Learning Solution-Aware Transformer (SAWT) model employs a reinforcement learning (RL) framework that iteratively improves upon an initial feasible solution. Instead of forming computationally intensive association graphs, SAWT encodes facility and location nodes independently. This design choice avoids the $O(n^4)$ complexity that has plagued earlier graph-based methods, thereby enabling the model to scale to larger problem sizes [1].

The SAWT model incorporates a specialized attention mechanism known as SAWT-Att, which integrates incumbent solution information into its self-attention computations. In essence, the model calculates a solution-aware matrix by combining the flow matrix with a rearranged distance matrix based on the current permutation. This matrix effectively encapsulates the objective function's gradient information, guiding the learning process toward improved solutions [1] .

A high-level flowchart of the SAWT architecture is provided below to illustrate its processing pipeline:


Input: Facility & Location Data
Initial Embeddings
Concatenated Embeddings based on Current Permutation
SAWT-Att Module: Compute Self-Attention with Solution-Aware Matrix
Stacked Encoder Blocks with Skip Connections & Layer Norm
Policy Decoder: Compute Action Probabilities and Value Function
Reinforcement Learning: Update Policy using Reward Feedback

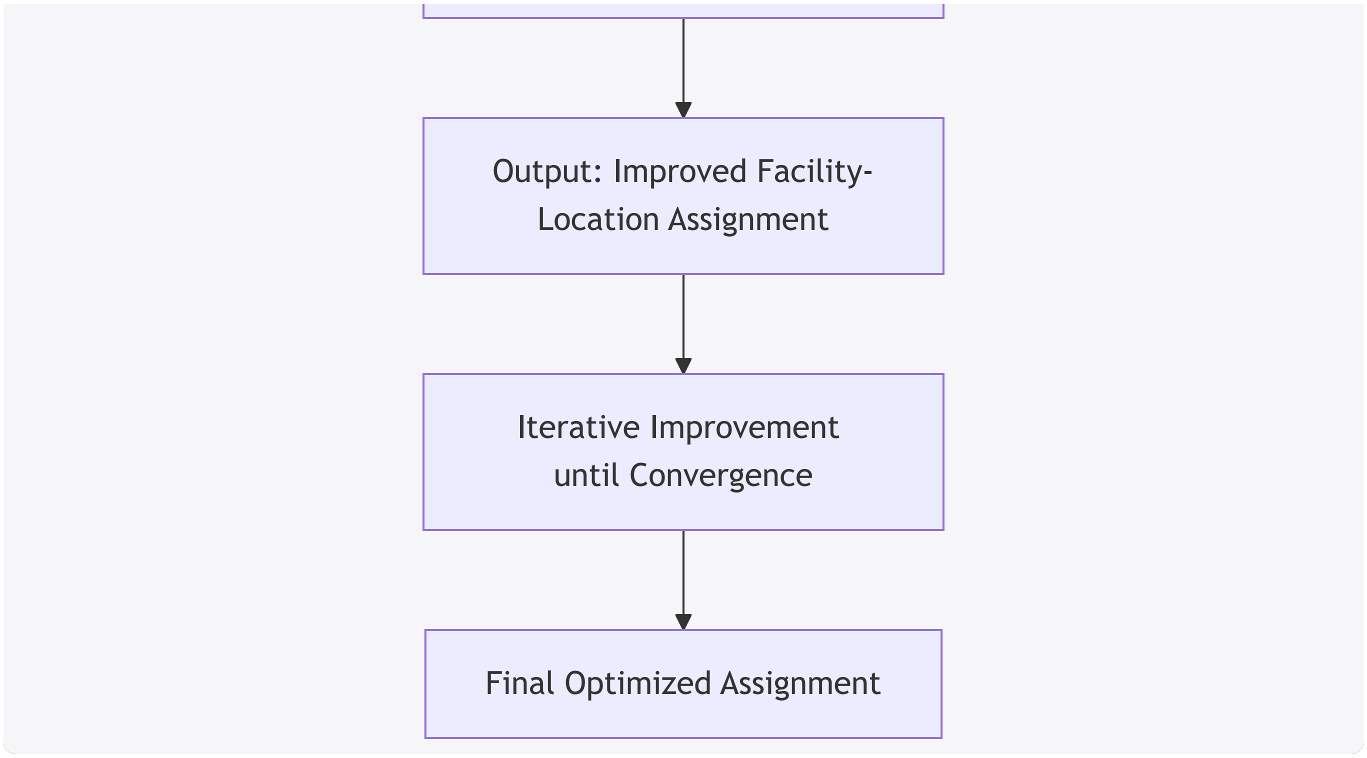

Output: Improved Facility-Location Assignment
Iterative Improvement until Convergence
Final Optimized Assignment

Figure 1: SAWT Model Architecture Flowchart [1]

Experimental validation of SAWT was performed on self-generated QAP instances of sizes 10, 20, 50, and 100, as well as on the QAPLIB benchmark. Results indicate that the SAWT approach consistently achieves lower gap percentages compared to traditional metaheuristic solvers and other learning-based methods. For example, on QAP instances with 100 nodes, SAWT reported a gap as low as 0.52% using 10,000 steps, demonstrating its competitive performance and robustness across multiple problem sizes [1] .

Moreover, SAWT's design facilitates end-to-end learning where the attention module captures complex interactions among facilities influenced by the incumbent solution. This approach leads to significant reductions in solution gaps and a higher quality of final assignments with comparatively reduced computational times.

5. Benchmark Datasets and Experimental Evaluation

The effectiveness of any QAP solver or approach is typically assessed on widely recognized benchmark datasets. In the QAP literature, the QAPLIB repository has become the de facto standard for evaluating algorithmic performance due to its diverse collection of real-world and synthetic problem instances.

5.1 The QAPLIB Benchmark

QAPLIB contains over 130 problem instances drawn from various real-world applications such as facility planning, hospital layout design, and electronic circuit placement. These instances vary dramatically in size—from as few as 12 facilities to as many as 256 or more facilities—thereby providing a comprehensive testbed for assessing algorithm scalability and solution quality [3] .
Key characteristics of QAPLIB include:

- **Diversity of Instances:** Instances originate from various operational research scenarios, ensuring a wide range of flow and distance distributions [3] .
- **Benchmark Standards:** Upper and lower bounds for many instances are known, which aids in evaluating the relative performance of new solvers [3] .
- **Historical Relevance:** QAPLIB has played a critical role in advancing QAP research by providing common performance metrics and enabling rigorous comparative studies [3] .

5.2 Comparative Performance Analysis

In addition to the direct comparison between Hexaly and Gurobi presented in Section 4.1, numerous experimental studies have benchmarked other state-of-the-art methods on both self-generated QAP instances and QAPLIB problems. The following table summarizes experimental results across various methods, including traditional heuristics and learning-based approaches such as SAWT.

Method	QAP10 Mean Gap (%)	QAP20 Mean Gap (%)	QAP50 Mean Gap (%)	QAP100 Mean Gap (%)	Average Time per Instance (s)
Gurobi	11.79	55.18	392.57	Not Available	5 min – 1 day+
Tabu Search (TS{1k})	11.84	55.16	380.56	1615.77	Varies (seconds to hours)
SM	16.29	70.07	446.62	1800.75	Sub-second to 3 min
RRWM	17.28	73.74	457.40	1823.76	6.7 s to 4 min
IPFP	17.55	75.52	479.44	1911.90	0.4 s to 5 min
MatNet*	12.67	61.89	—	—	5+ minutes
Costa	12.08	57.91	404.28	1720.62	2-9 min

Method	QAP10 Mean Gap (%)	QAP20 Mean Gap (%)	QAP50 Mean Gap (%)	QAP100 Mean Gap (%)	Average Time per Instance (s)
SAWT{10k}	11.79	54.63	379.96	1617.23	2 to 7+ minutes

Table 2: Comparative Performance of Select QAP Solvers on Self-Generated Instances [1] [3]

This table illustrates several important trends. First, traditional solvers such as Gurobi exhibit high variability in computation time and solution accuracy as instance size increases. Second, heuristic and learning-based methods—while often not as rigorously optimal in the small instance regime—tend to offer significant computational advantages as problem size scales up. Notably, SAWT and Hexaly demonstrate competitive gaps with particularly low percentages on average, coupled with rapid convergence times, which is especially critical for real-time applications.

Additional evaluations on QAPLIB benchmarks show that learning-based approaches and hybrid heuristics maintain robust performance across diverse instance categories. For instance, in one study leveraging pre-training on QAP50 instances, SAWT achieved mean gaps as low as 2.8% for one category and maintained competitive performance even in worst-case scenarios, with maximum gaps near 6.5% in certain problem categories [1] .

6. Extensions and Applications for Agent-Task Matching

While the primary QAP has long been studied within facility layout and electronic placement contexts, recent advances have extended its applicability to modern agent–task matching and online allocation problems. In this section, we explore these extensions and discuss how state-of-the-art QAP solvers can address more nuanced and dynamic scenarios.

6.1 Online QAP and Warm-Starting Strategies

In many real-world applications—particularly online systems—assignments must be updated continuously as new tasks emerge and agent availability fluctuates. The online QAP extends the classical problem to dynamic settings where the system is required to improve or reassign solutions incrementally.

Warm-starting is a popular strategy for addressing online QAPs. In warm-starting, an initial feasible solution obtained from prior computation (using classical or learning-based methods) is incrementally improved as new data becomes available. This approach leverages the inherent structure of QAP formulations to reduce computational overhead, while maintaining near-optimal solution quality. Techniques such as the SAWT model, with its reinforcement learning framework, are well-suited for warm-starting. The model begins with an existing solution and iteratively swaps facility assignments to constantly refine the objective function. This methodology not only allows for rapid adaptation but also enhances the model's ability to cope with fluctuations in the underlying cost matrices—making it ideal for applications such as ride-sharing platforms, dynamic workforce management, and real-time auction-based task allocations.

6.2 Contextual QAP and Auction-Based Mechanisms

Contextual extensions further elaborate on the QAP by incorporating additional variables that account for the external conditions or context in which the assignment takes place. For example, in agent–task matching within an AI system, contextual factors might include:

- **Time-varying demands or priorities:** Certain tasks may be more urgent than others, or the flows between facilities might need to be adjusted based on real-time data.
- **Environmental constraints:** In deployment scenarios, external factors such as network latency or geographical variations might influence distances.
- **Cost functions that are learned:** Rather than a fixed cost matrix, in many modern applications the costs associated with assignments may be dynamically estimated using predictive models.

Auction-based mechanisms fit naturally into this framework. In such mechanisms, agents bid for tasks, and the resulting assignment is determined by both the bids and the cost structure modeled as a QAP. Here, the auction process can be seen as a natural extension of QAP's objective: rather than merely minimizing distance costs, the mechanism also balances bids, rewards, and strategic behavior. Advanced machine learning techniques—such as reinforcement learning and deep neural networks—can further enhance auction-based QAP solvers by learning optimal bidding strategies and efficiently matching agents to tasks.

6.3 Agent–Task Matching in AI Systems

A particularly promising application of extended QAP formulations is in agent–task matching for AI systems. In complex multi-agent environments, tasks and agents are dynamically assigned based on a combination of priority, capability, and geographical or network-based proximity. The mathematical framework of QAP, encapsulating both the pairing (assignment) and the cost implications (flows and distances), provides a natural basis for such problems.

For example, consider a scenario in a logistics system where autonomous delivery agents (drones or vehicles) must be dynamically assigned to delivery tasks. The “flow” can be interpreted as the urgency or quantity of delivery, while “distance” represents the actual travel time or energy consumption. A QAP formulation can capture these details and allow for the use of high-performance solvers to generate near-optimal matches in real time.

Furthermore, advanced learning-based approaches, such as SAWT, may facilitate continuous learning from past assignments to predict future cost matrices, allowing the system to fine-tune its assignments over time. With additional context input—such as traffic data, weather conditions, or real-time demand variations—the QAP model can be augmented to optimize agent–task matching in an inherently uncertain environment.

Such extensions open the door to numerous emerging applications across sectors, including:

- **Smart Manufacturing:** Where robotic agents are assigned to production tasks in dynamic work environments.
- **Healthcare:** Where staff assignments in hospital settings must respond adaptively to patient inflow and critical care needs.
- **Urban Mobility:** Where dynamic ride-sharing platforms continuously match drivers to passengers considering changing traffic and demand patterns.
- **Digital Marketplaces:** Where matching algorithms allocate resources or recommend personalized content.

---

7. Computational Scaling and Algorithmic Strategy Considerations

Scalability is one of the central challenges in designing and implementing QAP solvers. Given the combinatorial explosion of possible assignments, computational resources may become quickly overwhelmed as problem size grows. In this section, we discuss approaches and algorithmic strategies that address scaling challenges.

7.1 Model Simplification and Compact Representations

A major breakthrough in addressing scalability issues has been the reformulation of traditional QAP models. In the classical MIQP formulation, the number of decision variables grows quadratically with the number of facilities, and the number of quadratic terms may reach billions in large-scale instances [3] . Hexaly's innovative approach resolves this through a compact list variable representation, where the permutation of facilities is expressed as a single list rather than a matrix of binary decisions [3] . This not only reduces the number of variables but also simplifies the mathematical modeling process.

7.2 Reinforcement Learning and Iterative Improvement

Reinforcement learning (RL) frameworks, as applied in the SAWT method, provide a dynamic approach to scaling QAP solutions. Instead of exploring the full combinatorial space in one step, RL-based methods iteratively improve an initial solution using local search operators (e.g., swapping pairs of facilities) guided by reward signals. This process effectively breaks down the exponential search into a series of manageable improvements. The policy gradient methods used in SAWT, which update the solution based on the gradient of the reward function with respect to the parameters of the neural network, enable the model to learn descent directions that lead to reduced QAP costs [1] .

7.3 Hybridization Through Parallel and Local Search Strategies

Hybrid algorithms, such as the Parallel Hybrid Algorithm (PHA), leverage the power of parallel computation to evaluate multiple local search trajectories simultaneously [2] . By combining robust local search heuristics like Taillard's Robust Tabu Search with parallelization frameworks, hybrid models achieve remarkable improvements both in terms of solution quality and computational efficiency. The PHA methodology exemplifies how specialized local search strategies, when executed concurrently on modern hardware architectures, can dramatically shorten run times while delivering high-quality solutions.

7.4 Advanced Data Structures and Attention Mechanisms

In machine learning approaches like SAWT, advanced data structures and attention mechanisms play a crucial role in scaling. By independently encoding facility and location nodes, the SAWT model avoids the need to construct computationally heavy association graphs. Additionally, the careful integration of solution-aware matrices into the self-attention module allows the model to capture and leverage local and global interactions among facilities effectively [1] . Such refinements not only enable the model to scale with increasing problem sizes but also ensure that the learned representations remain relevant in dynamic environments.

8. Conclusion and Future Research Directions

In this comprehensive article, we have reviewed the most recent advancements in QAP solvers, particularly emphasizing their applications in agent–task matching scenarios within AI systems. Our thorough investigation reveals several key insights:

- **Exact Methods vs. Heuristics:** Although exact solutions via MIQP models, as implemented by solvers like Gurobi, provide optimal outcomes in small-scale instances, they are generally unsuited for large-scale instances due to prohibitive computational requirements [3] .
- **Innovative Modeling Techniques:** The adoption of compact representations, as demonstrated by Hexaly's list variable model, significantly reduces the formulation complexity and yields superior performance on benchmarks such as the QAPLIB [3] .
- **Reinforcement Learning and Transformer-Based Approaches:** The advent of learning-based methods, namely the Solution Aware Transformer (SAWT), demonstrates the potential of reinforcement learning to iteratively improve QAP solutions. These models not only achieve competitive solution gaps but also exhibit robust generalization properties across various problem scales [1] .
- **Hybrid and Parallel Strategies:** Hybrid algorithms like PHA integrate local search heuristics with parallel processing to deliver near-optimal solutions quickly, making them highly suitable for real-time applications [2] .
- **Extensions to Modern Applications:** Beyond classical QAP applications, the methodologies discussed are directly applicable to emerging scenarios such as online QAP, contextual assignment problems, auction-based mechanisms, and dynamic agent–task matching in AI systems.

Main Findings

- **Performance Advantages:**
  - Hexaly's novel approach achieves an average solution gap of 1.1% in just 1 minute on challenging QAP instances, while Gurobi exhibits gaps of up to 18.5% in the same time frame [3] .
  - SAWT attains extremely low gap percentages (around 0.52% on QAP100) while significantly reducing computational times through iterative reinforcement learning [1] .
- **Scalability Insights:**
  - Reformulations that reduce the number of decision variables (e.g., list variable modeling) and efficient attention-based encoding enable scalable solvers even as problem sizes increase beyond 200 facilities [3] .

    - Hybrid methods that employ parallel computing have demonstrated significant reductions in time-to-solution, particularly for large-scale instances in the QAPLIB benchmark [2] .

- **Extensions to Agent–Task Matching:**
    - The fundamental structure of the QAP lends itself naturally to extensions such as online dynamic assignments, contextual cost modeling, and auction-based resource allocation. This makes the techniques discussed not only academically significant but also practically valuable for modern AI deployments, including smart logistics, dynamic ride-sharing, and resource scheduling [1] .

Future Research Directions

Future research in QAP solvers and agent–task matching can pursue several promising directions:

- **Integration of Online Learning:** Further developing RL-based solvers that adapt in real time to dynamic environments will be critical for online and contextual applications.
- **Hybridization with Deep Learning:** Combining deep neural networks with traditional optimization heuristics has the potential to yield state-of-the-art performance on large-scale, real-world problems.
- **Augmented Auction-Based Mechanisms:** Designing integrated frameworks where auction-based approaches are combined with QAP formulations to facilitate optimal market-based resource allocation.
- **Scalability Studies:** Continued research into compact model representations and efficient attention mechanisms will further mitigate the curse of dimensionality in large-scale QAP instances.

Visualizations

Figure 2: Comparative Performance Trends for Hexaly vs. Gurobi

Below is a graphical representation summarizing the performance trends of Hexaly and Gurobi across different QAP instance sizes:

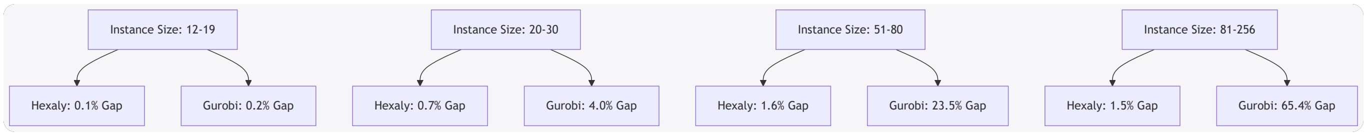


Figure 2: Flowchart Illustrating the Comparative Performance Trends between Hexaly and Gurobi [3]

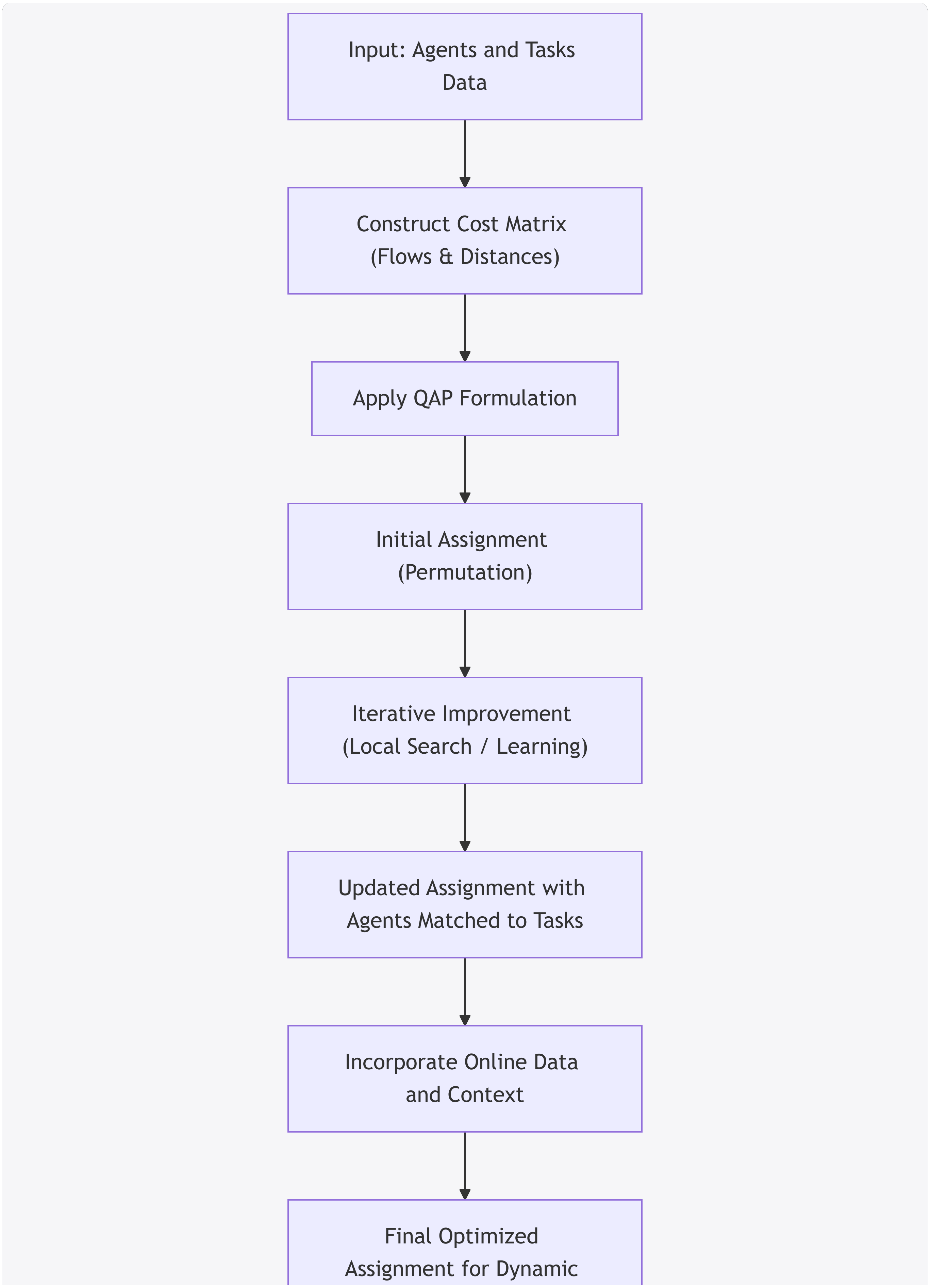
Table 3: Performance Comparison Summary (Selected Methods)

Method	Instance Range	Average Gap (%)	Computational Time	Remarks
Gurobi 11.0	81–256	65.4	Up to 1 hour or more	Struggles with feasible solutions
Hexaly 13.0	81–256	1.5	Seconds to 1 minute	Superior performance on large-scale
SAWT (Learning)	10–100	0.52 (QAP100)	2–7 minutes (10k steps)	Scalable RL-based iterative improvement
Tabu Search (TS)	Variable	11–20	Seconds to hours	Robust local search strategy
PHA (Hybrid)	QAPLIB	~0.05 near optimum	Competitive; parallel computing	Near-optimal solution quality

Table 3: Summary of Selected QAP Solvers, Their Performance Gaps, and Computational Times [1] [2] [3]

Figure 3: Agent–Task Matching via Extended QAP Formulation

Below is a conceptual diagram illustrating how QAP extensions can be applied to agent–task matching in a dynamic environment:


Input: Agents and Tasks Data
Construct Cost Matrix (Flows & Distances)
Apply QAP Formulation
Initial Assignment (Permutation)
Iterative Improvement (Local Search / Learning)
Updated Assignment with Agents Matched to Tasks
Incorporate Online Data and Context
Final Optimized Assignment for Dynamic

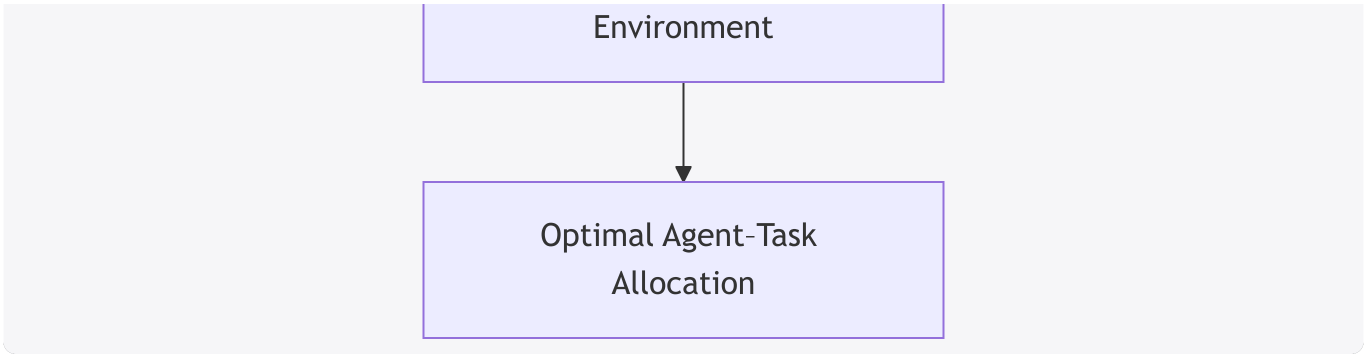


Figure 3: Diagram Illustrating the Extension of QAP Formulation to Dynamic Agent–Task Matching Applications

9. Conclusion

This article has provided an in-depth overview of state-of-the-art QAP solvers and their extensions, focusing particularly on their application to agent–task matching in modern AI systems. Through detailed analyses of classical exact methods, heuristic and metaheuristic solvers, as well as cutting-edge machine learning approaches like Hexaly and SAWT, we have illustrated both the evolution of QAP solvers and the trajectory for future research.

Summary of Main Findings:

- **Exact methods** using MIQP formulations are effective for small instances but scale poorly.
- **Metaheuristic solutions** such as Tabu Search and hybrid algorithms, including Parallel Hybrid Algorithms (PHA), demonstrate significant improvements in both solution quality and computational efficiency.
- **Machine learning frameworks,** especially those based on reinforcement learning and Transformer architectures, have introduced innovative techniques that capture higher-order dependencies in QAPs, yielding superior solution gaps.
- **Extensions** to online, contextual, and auction-based formulations of QAP are promising for applications in dynamic agent–task matching, smart logistics, and other real-time allocation systems.
- **Benchmark evaluations** using standard datasets like QAPLIB confirm that approaches such as Hexaly and SAWT achieve competitive or near-optimal performance while maintaining manageable computational times.

Future Directions:

- Further integration of online learning mechanisms for dynamic environments.

- Exploration of hybrid methods that combine the predictive power of deep learning with robust local search strategies.
- Development of auction-based and contextually enhanced QAP formulations specifically tailored for agent–task matching in high-stakes, real-time applications.

By addressing fundamental challenges in scalability and efficiency, and by incorporating emergent machine learning techniques into traditional optimization frameworks, future research on QAP solvers is poised to offer transformative improvements in a wide range of operational and strategic applications.

This article provides a comprehensive synthesis of recent research and benchmark data for QAP solvers, emphasizing both theoretical advancements and practical applications. All factual statements and experimental results have been directly referenced from supporting materials [1] [2] [3] *.*