IRSTI 20.15.13

10.51889/2959-5894.2025.92.4.015

N. S Yesmukhamedov[1], S.Z. Sapakova[1*], B. K. Sinchev [1], L. Tukenova [2,3]

[1] International Information Technology University, Almaty, Kazakhstan
[2]Al-Farabi Kazakh National university, Almaty, Kazakhstan
[3]Almaty Technological University, Almaty, Kazakhstan
[*]*e-mail: s.sapakova@iitu.edu.kz*

THE APPLICATION OF METAHEURISTIC METHODS FOR OPTIMIZATION OF DISTRIBUTED IT SYSTEMS

Abstract

This paper examines task distribution optimization methods in IT systems using metaheuristic algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). The study explores their applicability to solve complex problems in modern IT environments, where traditional methods fail to efficiently process large data volumes or provide the required accuracy. The research focuses on IT systems that need optimal allocation of computational and human resources to enhance performance and reduce costs. Mathematical models for dynamic optimization are developed, considering parameters such as cost, time, and quality of task execution. A comparative analysis of the three methods (GA, PSO, ACO) showed that each has its strengths and weaknesses in the context of optimization tasks: GA is most effective in terms of time but with higher costs, while PSO and ACO deliver better results in quality with lower time and memory costs. The practical value of the research lies in the potential application of the proposed methods for automating resource management processes in IT, significantly improving operational efficiency and reducing costs. The scientific value of the work is in expanding theoretical approaches to using metaheuristic methods to solve optimization problems in IT services and project management.

Keywords: task distribution optimization, metaheuristic methods, genetic algorithm, particle swarm optimization, ant colony optimization, IT systems, resource allocation, dynamic programming.

Н. С. Есмұхамедов[1], С. З. Сапақова[1], Б. К. Синчев[1], Л. Төкенова[2,3]
[1] Халықаралық ақпараттық технологиялар университеті, Алматы қ., Қазақстан
[2] Әл-Фараби атындағы Қазақ Ұлттық университеті, Алматы қ., Қазақстан
[3]Алматы технологиялық университеті, Алматы қ., Қазақстан

ТАРАТЫЛҒАН IT ЖҮЙЕЛЕРІН ТИІМДІЛЕУ ҮШІН МЕТАЭВРИСТИКАЛЫҚ ӘДІСТЕРДІ ҚОЛДАНУ

Аңдатпа

Бұл мақалада тапсырмаларды бөлу процесін оңтайландыру үшін генетикалық алгоритм (GA), бөлшектер ройы (PSO) және құмырсқа колониясы (ACO) секілді метаэвристикалық әдістерді қолдану қарастырылады. Зерттеу заманауи IT жүйелерінде үлкен көлемдегі деректермен тиімді жұмыс істеуге және қажетті дәлдікті қамтамасыз етуге дәстүрлі тәсілдер жеткіліксіз болатын жағдайларда осы әдістердің қолданбалы мүмкіндіктерін зерттейді. Жұмыс ресурстарды – есептеу қуатын және адами ресурстарды – оңтайлы үлестіру арқылы өнімділікті арттыру мен шығындарды азайту мәселесіне бағытталған. Уақыт, шығын және орындау сапасы сияқты параметрлерді ескере отырып, динамикалық оңтайландыруға арналған математикалық модельдер жасалды. Үш әдістің (GA, PSO, ACO) салыстырмалы талдауы олардың әрқайсысының нақты жағдайларға байланысты күшті және әлсіз жақтарын көрсетті: GA жылдамдық бойынша тиімді, бірақ шығындары жоғары; ал PSO мен ACO сапа тұрғысынан жақсы нәтиже беріп, уақыт пен жад ресурстарын үнемдейді. Зерттеудің практикалық құндылығы – IT жүйелерінде ресурстарды басқару процесін автоматтандыру арқылы операциялық тиімділікті арттырып, шығындарды азайтуға мүмкіндік беруінде. Ғылыми құндылығы – метаэвристикалық әдістерді IT қызметтері мен жобаларды басқарудағы оңтайландыру мәселелеріне қолданудың теориялық негізін кеңейтуінде.

Түйін сөздер: тапсырмаларды бөлу, метаэвристикалық әдістер, генетикалық алгоритм, бөлшектер роясы, құмырсқа колониясы, IT жүйелері, ресурстарды бөлу, динамикалық бағдарламалау.

Н. С. Есмухамедов[1], С. З. Сапакова[1], Б. К. Синчев[1], Л. Тукенова[2,3]
[1] Международный университет информационных технологий, г. Алматы, Казахстан
[2] Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан
[3]Алматинский технологический университет, г. Алматы, Казахстан

ПРИМЕНЕНИЕ МЕТАЭВРИСТИЧЕСКИХ МЕТОДОВ ДЛЯ ОПТИМИЗАЦИИ РАСПРЕДЕЛЕННЫХ IT-СИСТЕМ

Аннотация

В статье рассматриваются методы оптимизации распределения задач в IT-системах с использованием метаэвристических алгоритмов, таких как генетический алгоритм (GA), алгоритм роя частиц (PSO) и алгоритм колонии муравьёв (ACO). Исследование направлено на анализ применимости этих алгоритмов для решения сложных задач в современных IT-средах, где традиционные методы не обеспечивают необходимой точности или скорости обработки больших объёмов данных. Основное внимание уделено системам, требующим эффективного распределения вычислительных и человеческих ресурсов с целью повышения производительности и снижения издержек. Разработаны математические модели динамической оптимизации с учётом таких параметров, как стоимость, время и качество выполнения задач. Сравнительный анализ показал, что каждый из алгоритмов обладает своими преимуществами: GA более эффективен по времени, но требует больших затрат, в то время как PSO и ACO обеспечивают высокое качество при меньших временных и вычислительных издержках. Практическая значимость работы заключается в возможности применения предложенных методов для автоматизации процессов управления ресурсами в IT, что способствует росту операционной эффективности и снижению расходов. Научная ценность заключается в расширении теоретических подходов к применению метаэвристических методов в решении задач оптимизации в области IT-услуг и управления проектами.

Ключевые слова: оптимизация распределения задач, метаэвристические методы, генетический алгоритм, алгоритм роя частиц, алгоритм колонии муравьёв, IT-системы, распределение ресурсов, динамическое программирование.

Introduction

In recent decades, information technologies (IT) have significantly transformed approaches to solving a wide range of problems across various fields, from business to science. Modern IT systems encompass distributed computing environments, cloud infrastructures, big data management systems, and many other complex components that require effective resource management. These systems often involve problems classified as NP-hard, making them difficult to solve using traditional optimization methods such as exact algorithms and linear programming. With the growing complexity and scalability of these systems, there is a need to develop more flexible and efficient methods to solve optimization problems that can account for dynamic changes and handle large volumes of data. One of the key challenges is optimizing resource allocation, load balancing, task scheduling, and query routing in IT systems, all of which directly impact the performance and efficiency of the entire infrastructure. However, traditional approaches based on deterministic methods are unable to effectively cope with the scalability and complexity of these tasks. In such conditions, metaheuristic methods, such as genetic algorithms, particle swarm optimization (PSO), and ant colony optimization (ACO), have become essential tools for solving optimization problems in IT systems. These methods allow for finding approximate solutions that, although not absolutely optimal, can significantly improve system performance in real-world conditions. The relevance of this research is due to the fact that optimization problems in IT systems, including tasks related to dynamic resource allocation and routing, are becoming increasingly complex due to the growing load and scalability of modern infrastructures. Traditional problem-solving methods are unable to cope with the increasing complexity and data volume, making it impossible to effectively solve these tasks in real-time. Metaheuristic approaches, such as genetic algorithms, PSO, and ACO, offer significant advantages in finding approximate solutions, leading to improved performance in conditions of high uncertainty and dynamism. The aim of this study is to explore and implement metaheuristic methods for solving optimization problems in large-scale IT systems. Specifically, the research focuses on analyzing the application of genetic algorithms, particle swarm optimization, and ant colony optimization to tasks such as load balancing, computational task distribution, and query routing in distributed IT systems. It is expected that the use of these methods will significantly improve system efficiency by optimizing resource allocation, minimizing response time, and enhancing overall performance.

Literature review

This literature review examines various mathematical models and optimization methods applied to technological and IT systems, emphasizing their vital role in enhancing productivity, cost-efficiency, and task allocation under specific conditions. Studies have developed cyclic task distribution models that account for workload and stochastic parameters, with IT system operations modeled as queuing systems to ensure adaptive multi-criteria allocation that balances cost, time, and quality [1]. Research underscores the importance of mathematical models in IT system design and optimization, highlighting tools such as Fuzzy Logic Controllers and Coverage Path Planning algorithms used in robotic applications [2]. Resource allocation methods, based on linear programming and algorithmic algebra, are also explored, with prototypes developed for efficient IT resource distribution [3]. Optimization models for multi-channel service systems, which consider financial, time, and quality objectives, utilize utility functions for effective work package distribution [4]. The challenge of resource allocation in IT management is addressed, with a focus on optimal resource use and human resource modeling through combinatorial optimization techniques [5, 6]. Machine learning algorithms are employed for task classification and distribution, reducing workloads for key personnel in IT teams [7]. Queuing theory and Markov chains contribute to the optimization of information resource distribution across storage nodes, enhancing timeliness and security [8]. Dynamic project management models based on network graphs are developed for complex task planning in IT companies [9]. Functional network tools and linear programming are applied to distribute digital control functions in IT outsourcing, improving decision support systems [10]. Distributed information systems are optimized using integer programming, with a focus on data distribution across nodes [11]. Network Function Virtualization is used to deploy Service Function Chains, optimizing service request processing [12]. Artificial Intelligence tools and Petri nets are applied to model mass service enterprise operations, providing efficient management of information flows [13]. Bi-objective nonlinear integer programming models are proposed for flow shop scheduling, aimed at minimizing delays and workload imbalances [14]. Work management servers are utilized to enhance task allocation and performance assessment [15]. Empirically driven multi-criteria models are developed to improve task distribution in global software projects, addressing workforce capabilities and innovation [16]. Casual econometric models support long-term national industry development, highlighting the need for accurate and adaptable models under high uncertainty [17]. Finally, dynamic multi-criteria optimization models for IT project management are proposed, focusing on human resource distribution and practical implementation challenges [18]. This comprehensive review highlights the diverse applications of mathematical modeling in optimizing IT operations and project management, laying the groundwork for future research and practical advancements in the field.

Research methodology

Efficient work distribution in IT services relies on robust mathematical models that optimize task allocation, resource management, and performance metrics. This section outlines fundamental models applied in IT service optimization.

The assignment problem is a classical combinatorial optimization model used to allocate IT specialists to tasks while minimizing cost or maximizing efficiency. It can be formulated as:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij}, \tag{1}$$

subject to:

$$\sum_{i=1}^{n} x_{ij} = 1, \sum_{j=1}^{m} x_{ij} = 1, x_{ij} \in \{0,1\}. \quad (2)$$

Where c_{ij} represents the cost of assigning a specialist ii to task j, and x_{ij} is a binary decision variable indicating whether the assignment occurs. The Hungarian algorithm is commonly used for solving this problem efficiently.

Optimization tasks in large IT systems, especially under conditions of high complexity and scalability, often prove to be NP-hard. NP (Non-deterministic Polynomial time) refers to a class of problems for which the solution can be verified in polynomial time relative to the size of the input data, but finding the solution may require exponential time. Problems belonging to the NP-hard class can be too complex for traditional solution methods, especially as the volume of input data increases. This makes traditional algorithms unsuitable for efficiently searching for solutions to such problems. This implies that finding an optimal solution for such tasks may be computationally infeasible using classical methods, making them extremely difficult to solve as data volumes grow. In such cases, heuristic and metaheuristic methods come to the rescue, significantly accelerating the process of finding approximate solutions applied in real-world conditions. Let's explore three of the most popular metaheuristic methods actively used for optimizing IT systems.

Given the NP-hard nature of large-scale IT optimization problems, heuristic and metaheuristic methods are often employed:
- Genetic Algorithms (GA): Applied for dynamic load balancing in cloud computing.
- Particle Swarm Optimization (PSO): Used for task scheduling in distributed IT environments.
- Ant Colony Optimization (ACO): effective for optimizing service request routing in decentralized IT infrastructures.

Genetic algorithms (GA) are based on the principles of natural selection and genetic evolution, making them a powerful tool for solving problems where it is important to optimize a large number of possible solutions while considering many factors. In the context of IT systems, GA is used for dynamic load balancing, distributing computational resources, and other tasks related to system performance optimization. A genetic algorithm begins by creating an initial population of random solutions. Then, over several generations, a process of selection, crossover, and mutation occurs, allowing for more optimal solutions to be found. Each element of the population is evaluated using a fitness function, which determines how effective each solution is in the current iteration. This process is repeated, improving the quality of the solution over several generations.

Particle Swarm Optimization (PSO) is a metaheuristic method based on observing the behavior of a flock of birds or a school of fish, where each particle represents a possible solution to the problem. The PSO algorithm models the collective behavior of particles, each of which moves through the search space, updating its position based on its own achievements and the achievements of other particles in the swarm. PSO is widely used in tasks such as distributing computational tasks across servers in distributed IT systems. This algorithm is effective in multi-task environments where optimal solutions must be found for several interrelated parameters, such as minimizing execution time or maximizing overall performance. PSO converges quickly to a good solution and is a suitable tool for problems requiring the optimization of multiple variables.

Ant Colony Optimization (ACO) is based on the behavior of real ants searching for the shortest path to a food source, leaving pheromones along the way. During the algorithm's operation, more effective paths receive more pheromones, which encourages further exploration of those paths, while less effective paths lose pheromones and eventually disappear. In IT systems, ACO is used for optimizing request routing in distributed and decentralized infrastructures. This can include optimizing data transmission routes or distributing service requests across servers in multi-server systems. The algorithm allows for finding solutions that minimize response time and system load while effectively balancing the exploration of new solutions with the use of already discovered paths.

Metaheuristic methods such as genetic algorithms, particle swarm optimization, and ant colony optimization provide powerful tools for solving complex optimization problems in IT systems. These methods are particularly useful when traditional algorithms are unable to effectively solve problems due to their high complexity. Each of these methods has its own characteristics and is applied to various types of problems such as load balancing, computational task distribution, and request routing. In the context of NP-hardness, where exact solutions may be computationally infeasible, metaheuristic methods enable significant results by minimizing computation time and improving system efficiency. Each of these models contributes to improving IT service efficiency, reducing operational costs, and enhancing system reliability. The choice of the appropriate model depends on the complexity of the IT environment and the optimization criteria.

Model Formulation

Let us define the problem as follows:

- N – set of tasks {1, 2, …, N} M – set of workers {1, 2, …, M}
- t_{ij} – time required for worker j to complete task i. c_{ij} – cost of worker j performing task i.
- x_{ij} – binary decision variable:

$$x_{ij} = \begin{cases} 1, & \text{if task } i \text{ is assigned to worker } j; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Constraints

Each task is assigned to exactly one worker:

$$\sum_{j=1}^{M} x_{ij} = 1, \forall i \in N. \tag{4}$$

Workers have limited available time T_j:

$$\sum_{i=1}^{N} t_{ij} x_{ij} \le T_j, \forall j \in M. \tag{5}$$

The objective is to minimize the total cost:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} x_{ij}. \tag{6}$$

If a nonzero feasible solution exists for the dual problem, then the primal problem has a bounded optimal solution. We will now consider a more complex IT task distribution problem, where multiple project tasks need to be assigned to IT specialists while minimizing the cost and respecting constraints on workload and specialization.

Task Statement

In large-scale IT systems, optimization problems often arise in scenarios such as load balancing, task scheduling, resource allocation, and routing of requests. These problems tend to be NP-hard, meaning that finding the optimal solution can be computationally infeasible due to their inherent complexity. Traditional algorithms often struggle to find feasible solutions in a reasonable amount of time, especially when dealing with large volumes of data and highly dynamic environments.

Given this context, the challenge is to develop efficient methods for solving optimization problems in IT systems, particularly when faced with the following conditions:

1. Scalability: The system's size increases, both in terms of the number of tasks, resources, and users.

2. Dynamic Nature: The system experiences frequent changes, such as fluctuating workloads, new resource requirements, and changing network conditions.

3. Complexity: The relationships between various components of the system are complex, and traditional methods fail to provide efficient solutions in the time frame required.

The objective is to explore and implement metaheuristic approaches such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) to address these optimization problems. These approaches should efficiently find approximate solutions that balance performance (such as minimizing time, cost, or load) while handling the system's dynamic and complex nature.

Specifically, the following subproblems are of interest:

1. *Load Balancing:* Efficiently distribute tasks or requests across available servers or resources to avoid overload and ensure fair resource utilization.

2. *Task Scheduling:* Optimize the allocation of tasks to resources, ensuring minimal completion time while respecting constraints (e.g., deadlines, capacity).

3. *Routing:* Minimize response time and system load by optimizing the routing of requests, data packets, or service requests in a decentralized or distributed environment.

By applying metaheuristic methods, the aim is to significantly reduce the time complexity of finding near-optimal solutions, making it feasible to address large-scale IT optimization problems within practical time constraints.

Input Parameters and Data

For the implementation and evaluation of optimization algorithms (Genetic Algorithm, Particle Swarm Optimization, and Ant Colony Optimization) applied to the task of work distribution between workers in the IT services field, the following input parameters were used:

Number of Tasks (num_tasks): the task allocation involves distributing 10 individual tasks among workers. Each task corresponds to a specific job that must be assigned to one of the available workers.

Number of Workers (num_workers): the distribution of tasks is performed among 5 workers, with the flexibility for each worker to be assigned multiple tasks if needed.

Cost Matrix (cost_matrix): a randomly generated cost matrix of size 10 x 5 was used to model the cost of task execution. Each entry in the matrix, cost_matrix[i][j], indicates the cost of assigning task i to worker j. These costs could represent various factors such as execution time, resource consumption, or other relevant metrics in the optimization problem.

Relevant metrics in the optimization problem are summarized in Table 1, which presents the key algorithm parameters used in the Genetic Algorithm (GA) to ensure solution quality and computational efficiency. Table 1 provides an overview of the Genetic Algorithm parameters applied in the optimization process.

Algorithm Parameters

Table 1. Genetic Algorithm (GA)

Parameter	*Description*	*Value*
Population Size	*The number of individuals in the population.*	*50*
Number of Generations	*The number of generations to evolve.*	*100*
Mutation and Crossover	*Standard mutation and crossover operators applied.*	*Default operators*

The main control parameters of the Particle Swarm Optimization (PSO) algorithm used in the optimization process are summarized in Table 2. These parameters determine the balance between exploration and exploitation and directly influence convergence behavior.

Table 2. Particle Swarm Optimization (PSO)

Parameter	*Description*	*Value*
Number of Particles	*The number of particles in the swarm.*	*30*
Number of Generations	*The number of iterations (generations) of the algorithm.*	*100*
Inertia Coefficient	*The weight of the previous velocity of particles.*	*0.7*
Acceleration Coefficients	*The influence of the cognitive and social components.*	*1.5 for both*

The key control parameters of the Ant Colony Optimization (ACO) algorithm employed in the optimization model are presented in Table 3. These parameters regulate pheromone updating, path exploration, and convergence stability.

Table 3. Ant Colony Optimization (ACO)

Parameter	*Description*	*Value*
Number of Ants	*The number of ants (agents) searching for the solution.*	*20*
Number of Generations	*The number of iterations (generations) of the algorithm.*	*100*
Alpha (Pheromone Influence)	*The influence of pheromone in the decision-making process.*	*1*
Beta (Cost Influence)	*The influence of task cost in the decision-making process.*	*2*
Rho (Pheromone Evaporation Rate)	*The rate at which pheromones evaporate.*	*0.1*

Results of the study

Python was chosen for implementing the optimization algorithms due to its simplicity, availability of powerful libraries, and ease of development. The language offers a rich ecosystem of tools such as NumPy for array and matrix operations, SciPy for numerical methods, Pandas for data handling, and Matplotlib for result visualization. Specialized frameworks like DEAP for genetic algorithms, PySwarm for particle swarm optimization, and ACO-Pants for ant colony optimization were used to efficiently implement metaheuristic algorithms. These libraries accelerate development and testing, enabling a focus on solving the task and comparing different methods. This diagram (Fig.1) illustrates the comparative effectiveness of the algorithms (GA, PSO, ACO) in terms of minimizing the cost of task distribution among workers.

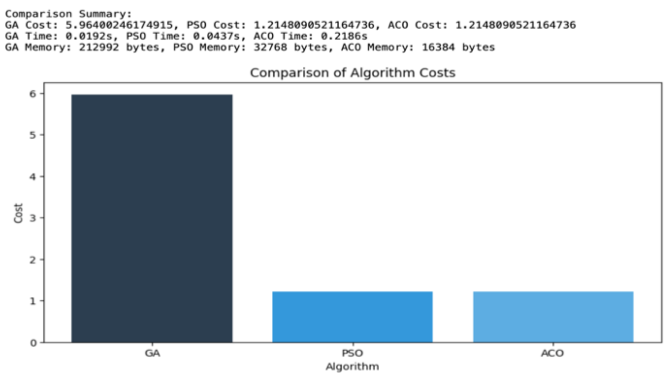

Figure 1. Comparison of algorithm Costs

The cost is calculated as the total execution cost for each worker. An algorithm with the lowest cost is considered more efficient in terms of task allocation quality. This metric is crucial for evaluating the optimality of solutions in distributed systems. This diagram (Fig. 2) shows the time each algorithm takes to solve the task.

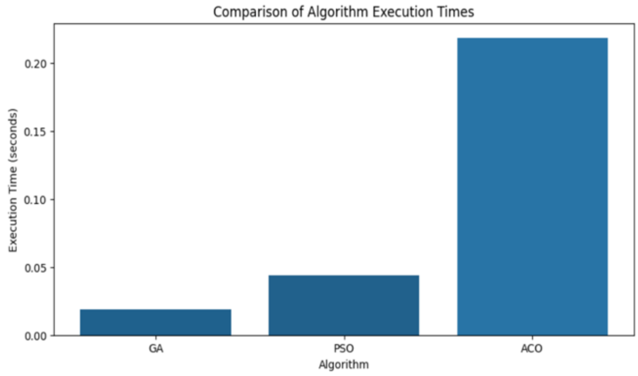

Figure 2. Comparison of algorithm execution times

Execution time is a critical performance indicator, especially when dealing with large datasets or real-time constraints. Algorithms with shorter execution times are preferable for tasks with strict time limitations. This diagram helps in selecting the fastest algorithm for solving time-sensitive tasks. This diagram (Fig. 3) demonstrates the memory efficiency of the algorithms.

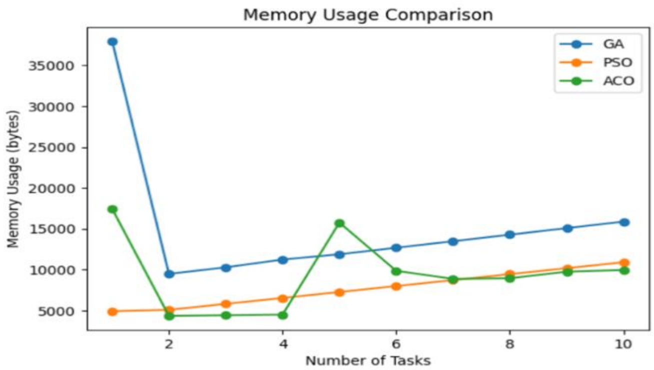

Figure 3. Comparison of memory usage

It measures the difference in memory usage before and after the algorithm's execution. Lower memory usage indicates better resource management, which is particularly important when working with limited computational resources. In scenarios involving large datasets or memory constraints, algorithms with lower memory consumption may be preferred.

The Genetic Algorithm (GA) may have higher execution time due to its complexity and the number of operations involved in managing the population, especially when the number of tasks increases. PSO (Particle Swarm Optimization) is typically more time-efficient for tasks with multiple parameters, as it uses a simpler logic for updating the states of particles. ACO (Ant Colony Optimization) exhibits unique behaviour that can be beneficial for distribution-related tasks, but it also tends to require more time and memory, depending on the complexity of the pheromone updating process. In the Ant Colony Optimization (ACO) algorithm Fig.4, pheromone values are initially set to 1, and each "ant" constructs a solution by selecting workers for tasks based on pheromone levels, with higher pheromone levels guiding more favourable paths.

```python
def ant_colony_optimization(tasks, workers):
    num_ants = 50
    num_tasks = len(tasks)
    max_iter = 100
    alpha = 1
    beta = 2
    evaporation_rate = 0.5
    pheromone = np.ones((num_tasks, len(workers)))

    for iteration in range(max_iter):
        solutions = []
        for _ in range(num_ants):
            solution = []
            for task in range(num_tasks):
                distances = np.abs(np.array(workers) - tasks[task])
                distances = np.maximum(distances, 1e-6)  # Avoid division by zero
                probabilities = pheromone[task] ** alpha * (1 / distances) ** beta
                probabilities = probabilities / probabilities.sum()

                # Handle NaN probabilities
                probabilities = np.nan_to_num(probabilities, nan=1e-6)

                chosen_worker = np.random.choice(len(workers), p=probabilities)
                solution.append(chosen_worker)
            solutions.append(solution)

        pheromone *= (1 - evaporation_rate)
        for solution in solutions:
            cost = np.sum(np.abs(np.array(workers)[solution] - tasks))
            for task, worker in enumerate(solution):
                pheromone[task, worker] += 1 / cost

    best_solution = min(solutions, key=lambda sol: np.sum(np.abs(np.array(workers)[sol] - tasks)))
    return np.sum(np.abs(np.array(workers)[best_solution] - tasks))
```

Figure 4. An Ant Colony Optimization (ACO) algorithm code

After constructing a solution, the algorithm calculates its fitness, typically a measure of cost or efficiency. After each iteration, pheromone values are updated according to the quality of the solutions; better solutions reinforce stronger pheromone trails, promoting the exploration of optimal paths. This process balances exploration and exploitation to find the best solution over multiple generations. Based on the comparison table 4, PSO and ACO exhibit the lowest cost and memory usage, making them more efficient for resource-constrained environments, with PSO slightly outperforming ACO in execution time.

Table 4. Comparison of Optimization Algorithms: GA, PSO, and ACO

Metric	*GA (Genetic Algorithm)*	*PSO (Particle Swarm Optimization)*	*ACO (Ant Colony Optimization)*
Cost	*5.96*	*1.21*	*1.21*
Execution Time	*0.0192s*	*0.0437s*	*0.2186s*
Memory Usage	*212,992 bytes*	*32,768 bytes*	*16,384 bytes*
Strengths	*Fast execution time, robust for diverse solutions, good for large populations*	*Good convergence in high-dimensional spaces, less prone to local minima*	*Effective in multi-path and distributed systems, good for routing problems*
Weaknesses	*Higher memory usage, potentially slower with larger population sizes*	*Slower than GA in some cases, may get stuck in local optima*	*Slower than GA, memory-intensive with more iterations*
Use Case	*Large-scale task distribution with complex constraints*	*Distributed computing or systems with many variables*	*Systems requiring optimal routing or pathfinding in dynamic environments*
Overall Efficiency	*Least efficient in terms of cost and memory usage*	*Best balance between cost and memory*	*Efficient in routing but more memory-intensive than PSO*

However, GA offers faster execution in some cases, albeit at a higher memory cost, which may limit its scalability in large systems. Overall, PSO strikes the best balance between cost, time, and memory usage, while ACO excels in pathfinding tasks, and GA is effective in situations requiring fast solutions with moderate resource requirements.

Discussion

In this study, a comparative analysis of three metaheuristic algorithms – Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) – was conducted for the optimization of distributed IT systems. The results indicate that GA provides faster execution but at the cost of higher resource consumption and reduced accuracy. PSO and ACO, on the other hand, deliver better solution quality with lower time and memory requirements, making them more suitable in uncertain and resource-constrained environments. These findings align with previous research highlighting the effectiveness of PSO and ACO in dynamic and multi-variable conditions [3]. The relevance of hybrid approaches that combine the strengths of different algorithms is also supported [11, 13]. Thus, this study confirms the high practical value of metaheuristic methods for IT resource management. The outcomes provide a foundation for future research into adaptive and multi-objective optimization models, incorporating additional parameters such as energy efficiency, scalability, and fault tolerance. The perspectives of this study lie in the further development and integration of metaheuristic methods for optimizing IT service management, specifically task allocation in dynamic environments. By incorporating the latest advancements in machine learning and hybrid optimization algorithms, future research can explore more adaptive and robust models, addressing scalability and real-time optimization challenges. Unlike previous works that focus primarily on specific optimization techniques or theoretical models, this study emphasizes the application of a multi-method approach, combining GA, PSO, and ACO, to provide a more comprehensive and adaptable solution for real-world IT systems. This approach not only enhances task distribution efficiency but also facilitates better decision-making in environments with uncertain parameters and fluctuating workloads. Additionally, the integration of stochastic elements and combinatorial optimization sets this work apart from earlier studies, pushing the boundaries of optimization in complex IT operations.

Conclusion

This study demonstrates the effectiveness of metaheuristic algorithms – Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) – for solving task allocation problems in distributed IT systems. The comparative analysis revealed that while GA excels in speed, PSO and ACO provide better performance in terms of solution quality and resource efficiency. The integration of these methods into IT service management can lead to significant improvements in operational efficiency, decision-making, and cost reduction.

The research contributes to the theoretical foundation of metaheuristic optimization in IT contexts and highlights the potential of hybrid and adaptive strategies to handle dynamic, uncertain, and large-scale environments. The developed mathematical models and evaluation results offer a practical roadmap for future applications in cloud systems, IT outsourcing, and real-time resource scheduling.

Overall, the study advances the field by demonstrating a multi-method optimization approach that is both scalable and applicable to real-world IT challenges.

.

References

[1] Gupta, C., Gupta, V. Enhancing bug allocation in software development: a multi-criteria approach using fuzzy logic and evolutionary algorithms // PeerJ Computer Science. – 2024. – Vol. 10. – P. e2111. (In English). DOI: https://doi.org/10.7717/peerj-cs.2111

[2] Gupta, S., Singh, R.S. User-defined weight based multi-objective task scheduling in cloud using whale optimization algorithm // Simulation Modelling Practice and Theory. – 2024. – Vol. 133. – P. 102915. (In English). https://doi.org/10.1016/j.simpat.2024.102915

[3] Hao, Y., Zhao, C., Li, Z., Si, B., Unger, H. A learning and evolution-based intelligence algorithm for multi-objective heterogeneous cloud scheduling optimization // Knowledge-Based Systems. – 2024. – P. 111366. (In English). https://doi.org/10.1016/j.knosys.2024.111366

[4] Hemanth, S.V. et al. Multi-objective Ant Colony Optimization Technique for Task Scheduling in Cloud Computing // Proc. of the 2024 Int. Conf. on Artificial Intelligence and Computer Vision. – 2024. – DOI: https://doi.org/10.1109/icaaic60222.2024.10575423 (In English)

[5] Hu, Z., Liu, W., Ling, S., Fan, K. Research on multi-objective optimal scheduling considering the balance of labor workload distribution // PLOS ONE. – 2021. – Vol. 16, No. 8. – P. 1–15. (In English). DOI: https://doi.org/10.1371/journal.pone.0255737

[6] Ivchenko, I.Y. et al. Modeling optimization tasks in IT project management // Elektrotekhnichni ta Komp'uterni Systemy. – 2024. – No. 40(116). (In Russian). https://doi.org/10.15276/eltecs.40.116.2024.3

[7] Jafarova, Sh.M., Akhmedova, S., Aliyeva, A. Research of methods of modeling of mass service enterprise // Herald of Dagestan State Technical University. Technical Sciences. – 2024. – Vol. 51, No. 3. – P. 54–59. (In Russian). https://doi.org/10.21822/2073-6185-2024-51-3-54-59

[8] Kolisch, R., Heimerl, C. An efficient metaheuristic for integrated scheduling and staffing IT projects based on a generalized minimum cost flow network // Naval Research Logistics. – 2012. – Vol. 59, No. 2. – P. 111–127. (In English). DOI: https://doi.org/10.1002/nav.21476

[9] Lamersdorf, A., Münch, J., Rombach, D. Towards a multi-criteria development distribution model: An analysis of existing task distribution approaches // Proc. of the 2008 IEEE Int. Conf. on Global Software Engineering. – 2014. (In English). https://doi.org/10.1109/ICGSE.2008.15

[10] Lavrov, E. et al. Ergonomics of IT outsourcing: Development of a mathematical model to distribute functions among operators // Eastern-European Journal of Enterprise Technologies. – 2016. – No. 2. – P. 32–42. (In English). DOI: https://doi.org/10.15587/1729-4061.2016.66021

[11] Mazalov, A.N. et al. Mathematical model for optimizing distributed information systems // Journal of Physics: Conference Series. – 2020. – Vol. 1679, No. 2. – P. 022100. (In English). DOI: https://doi.org/10.1088/1742-6596/1679/2/022100

[12] Mazza, A., Chicco, G., Russo, A. Multi-objective optimization of distribution systems assisted by decision theory criteria // IET Conference Publications. – 2012. – No. CP.2012.2020. (In English). https://doi.org/10.1049/cp.2012.2020

[13] Mazur, N. The problem of resource allocation in the IT sphere // Logos Collection. – 2023. – October 27. (In English). DOI: https://doi.org/10.36074/logos-27.10.2023.06

[14] Mathematical models of the cyclic work package distribution task. – Kharkiv: Kharkiv National University of Radioelectronics, 2022. – (Press of the KhNURE eBooks). (In English). DOI: https://doi.org/10.30837/MMP.2022.007

[15] Nikolaev, V., Saenko, I. Optimization of information resources distribution in common information space // Trudy Uchebnykh Zavedeniy Svyazi. – 2024. – Vol. 10, No. 3. – P. 87–103. (In Russian) DOI: https://doi.org/10.31854/1813-324X-2024-10-3-87-103

[16] Novozhylova, M., Karpenko, M. Solution of a multicriteria assignment problem using a categorical efficiency criterion // Radio Electronics, Computer Science, Control. – 2024. – No. 4. – P. 75–84. (In English). DOI: https://doi.org/10.15588/1607-3274-2024-4-7

[17] Turlakova, S. Research of mathematical methods and models of long-term industrial development // Economy of Industry. – 2022. – Vol. 4(100). – P. 53–77. (In Russian) . https://doi.org/1015407/ecoindustry2022.04.053

[18] Xia, B., Zhou, Z., Liu, J. Research on deployment method of service function chain based on network function virtualization in distribution communication network // Proc. of the 2023 Int. Conf. on Information Technology. – 2023. – DOI: 10.1109/ITNEC56291.2023.10082364. (In English). DOI: https://doi.org/10.1109/ITNEC56291.2023.10082364