Environment. Technology. Resources. Rezekne, Latvia
Proceedings of the 16th International Scientific and Practical Conference. 2025. Volume II, 221-224

Comparative Analysis of Heuristic Algorithms for Optimizing the Distribution of Medical Equipment in a Crisis

Iryna Kyrychenko
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
iryna.kyrychenko@nure.ua

Artem Oliinyk
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
artem.oliinyk@nure.ua

Kirill Smelyakov
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
kyrylo.smelyakov@nure.ua

Loreta Savulioniene
Faculty of Electronics and Informatics
Vilniaus Kolegija
Vilnius, Lithuania
l.savulioniene@viko.lt

Paulius Sakalys
Faculty of Electronics and Informatics
Vilniaus Kolegija
Vilnius, Lithuania
p.sakalys@eif.viko.lt

***Abstract*— The article presents a comparative analysis of the effectiveness of several heuristic algorithms used to optimize the distribution of medical equipment in a crisis. The key approaches, including the genetic algorithm (GA), the particle swarm optimization algorithm (PSO), and the ant colony optimization algorithm (ACO), are considered, and the performance and accuracy of the problem are investigated. Based on the experiments, the advantages and limitations of each method are determined, and practical recommendations are formulated for choosing the appropriate algorithm, considering the scale of the problem and limited resources. The study's results improve the reliability and efficiency of decision-making in healthcare systems during emergencies.**

Keywords*— *Crisis conditions, heuristic algorithms, medical equipment, performance assessment.

I. Introduction

Healthcare systems, especially in times of crisis (e.g., epidemics or natural disasters), are under significant strain, leading to acute shortages of medical resources and equipment. During such periods, it is crucial to ensure the prompt and efficient distribution of limited ventilators, diagnostic systems, and other devices, as patient survival rates and the overall functioning of hospitals depend on them. Studies have shown that optimizing the allocation of medical resources during crises is a key factor in reducing mortality and improving treatment efficiency [1]. Planning and coordinating the supply of such equipment has many limitations: heterogeneity of needs in different regions, different technical conditions, storage reserves, and the risk of a sharp increase in demand due to emergencies.

Classical optimization methods, such as linear programming or deterministic algorithms, can only guarantee the search for a precisely optimal solution when the problem is relatively small and the conditions are stable. In crises, where demand parameters change simultaneously, technical failures occur, and forecasts of the number of patients are uncertain, adaptive solutions are needed that can quickly adjust to new circumstances. That is why heuristic algorithms are becoming increasingly popular: they do not guarantee optimality but can find high-quality solutions quickly and easily change the search trajectory in an unstable environment [2, 3]. In particular, GA and PSO are widely used in large-scale resource allocation problems [4, 5], and the ACO sometimes effectively solves combinatorial variants of logistics problems.

This article aims to conduct a comparative analysis of GA, PSO, and ACO in optimizing the distribution of medical equipment for 50 hospitals in five different regions against the background of sharply changing conditions. The study used a synthetic dataset that simulates massive patient flows, various levels of criticality of their conditions, unplanned equipment

Online ISSN 2256-070X
https://doi.org/10.17770/etr2025vol2.8612

failures and sudden increases in demand. The analysis was performed in terms of speed, accuracy, stability of solutions, and the ability of algorithms to adapt to crisis events.

II. Materials and Methods

The study considered three heuristics. The first is the GA, which is based on natural selection simulation. It works with a population of possible solutions that are gradually improved using crossover and mutation operators. The study used Python and the DEAP library, a powerful tool for evolutionary algorithms, to implement GA. This library allows you to model a population of possible solutions and apply crossover, mutation, and selection operators, which are essential for achieving high-quality solutions under complex constraints [6]. The main stages of using the GA in this study include:

- Encoding solutions: Each potential resource allocation is represented as a vector of chromosomes:
  $$x = (x_{11}, x_{12}, \dots, x_{NM}); \quad (1)$$
- Crossover: Combining solutions from the "parent" chromosomes to create new options for resource allocation. For example, homogeneous recombination is used.
- Mutation: Random changes in the distribution of equipment x_{ij} within acceptable limits to avoid local extremes.
- Selection: Tournament selection or roulette strategies are used to determine which solutions are passed on to the next generation.

The second technique is the PSO [7]. The main idea of this approach is that each potential solution (particle) moves in the space of possible solutions and adjusts its trajectory based on its own best experience and the collective experience of the entire swarm. This allows the algorithm to quickly adapt to environmental changes and find practical solutions even under challenging constraints. Within PSO, each particle represents a possible option for resource allocation, and the following formula determines its movement:

$$v_{k+1} = w \cdot v_k + c_1 r_1 (p_{\text{best}} - x_k) + c_2 r_2 (g_{\text{best}} - x_k) \quad (2)$$

where w — is the inertia coefficient, which affects the speed of solution update; c_1, c_2 — learning rates, which determine how much a share is guided by its best practices (p_{best}) and the best solution found in the entire swarm (g_{best}), r_1, r_2 — random values that add a stochastic element to the search.

The main advantage of PSO is the ability to efficiently handle large-scale optimization problems while quickly finding solutions to variable conditions, such as fluctuations in demand or resource constraints [8]. Due to this, PSO is widely used in complex distribution systems, allowing for dynamic adaptation of medical resource management even in crises. The Python programming language and the PySwarms library [9] were used to implement PSO, simplifying the management of swarm parameters.

The third approach is the ACO. Its basic idea is that "ants" consistently build equipment distribution by adding types of resources according to a certain probability function that depends on "pheromone traces". These traces accumulate faster on more successful routes (combinations) and then evaporate, leaving the heuristic search for better states in focus. The ACO was implemented in a custom Python script using a matrix representation of the relationships between states and a mechanism for updating pheromones after each iteration [10]. This approach is often practical in complex discrete problems, but its performance for continuous distributions can also be satisfactory, depending on the parameters.

To verify the methods, we needed data that reflected the dynamics of equipment utilization in many hospitals, including sudden failures and fluctuations in demand. Obtaining such a sample from a real healthcare system proved problematic due to ethical and confidentiality restrictions, so synthetically generated scenarios were used.

First of all, 50 hospitals were modelled, evenly divided into five regions (10 per region), where each had its initial stock of four types of equipment: ventilators (5-20 units), Magnetic resonance imaging (MRI) (1-3), ultrasound systems (3-10), and X-ray machines (2-6). This difference reflected the different specialization and capacity of the hospitals. Next, a stream of 10,000 patients was generated, arriving randomly during a 30-day period. Their distribution in time was close to Poisson, and the characteristics included the type of equipment needed, the region of origin, and the level of criticality (from 1 to 5).

Two stages of the crisis were provided for modelling emergency circumstances. Between the tenth and fifteenth days, the demand for lung ventilators increased by 50%: this simulated an exacerbation of the epidemiological situation when the number of patients with severe respiratory pathology increased significantly. Later, on the twentieth to twenty-first day, about 10% of the available MRI machines in some hospitals randomly broke down, reducing diagnostic capacity. All changes were recorded in additional scripts as "events" that appear during the simulation and change the availability of the resource. Using the NumPy and Pandas libraries, the final data stored information about each hospital, all patients, and changes in equipment status for 30 days.

III. Results and Discussion

To ensure correctness, we defined a universal fitness function consisting of an essential target (3) and additional penalties for non-compliance with the constraints (4, 5) [11].

$$Z = \max_{x_{ij}} \sum_{i=1}^{N} \sum_{j=1}^{M} \omega_{ij} x_{ij}, \quad (3)$$

where x_{ij} — is the amount of equipment of type i assigned to hospital j; ω_{ij} — a weighting factor that considers the priority; N, M — number of equipment types and hospitals, respectively.

Several important limitations were taken into account. First, resource constraints that ensure that the total amount of equipment allocated to all hospitals does not exceed the available stock:

$$\sum_{j=1}^{M} x_{ij} \leq R_i \quad \forall i = 1, \dots, N \tag{4}$$

Secondly, the minimum needs of hospitals, which stipulate that each hospital should receive at least the amount of equipment it needs:

$$\sum_{i=1}^{N} x_{ij} \geq D_j \quad \forall j = 1, \dots, M \tag{5}$$

All of the above parameters (cost coefficients, procedure time, patient criticality) were used for all algorithms.

The calculations were performed in Python 3.9 using the DEAP (for GA), PySwarms (for PSO), and our code (for ACO) libraries. As part of the main experiment, each algorithm was executed for 200 search iterations, with all methods starting from different initial solutions five times to account for the random component. At the end of each run, the minimum and average Z values, standard deviation (SD), and time taken were recorded.

Based on the results of five repeated runs of each algorithm, the average values of the metrics were obtained. Table 1 compares the best and average values of the objective function Z, SD, and solution time. Each algorithm allocated the available equipment to minimize the logistics and penalty costs.

TABLE 1 COMPARATIVE RESULTS OF HEURISTIC ALGORITHMS

Algorithm	Performance Metrics			
	Best Z	***Average Z***	***SD (Z)***	***Average time, s***
GA	4820	4915	48,2	15,7
PSO	4765	4790	20,1	12,3
ACO	4880	4930	55,5	16,4

These results show that PSO, on average, finds the solution closest to the minimum value of Z and takes slightly less time. This indicates its high performance and ability to adapt well to dynamic scenarios. Although GA is inferior to PSO regarding the best achieved Z, the difference is insignificant. The GA also shows relatively stable solutions but sometimes requires more iterations to achieve better results. The ACO in the considered environment showed a slightly larger spread (SD 55.5), which may indicate more frequent "stuckness" in local minima, especially regarding the continuous nature of the distribution of quantitative resources, rather than a purely combinatorial routing problem.

A line graph is shown in Fig. 1 to compare how the value of Z changes during each of the five runs.

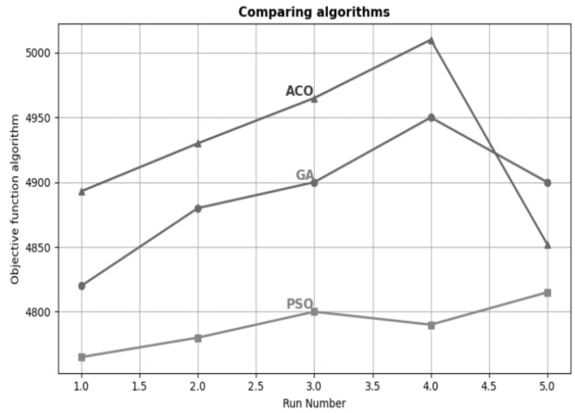

Fig. 1. Comparing algorithms.

It can be seen that PSO generally produces lower values of Z compared to GA and ACO, and the scatter of the ACO results confirms its greater sensitivity to variations in the initial decisions.

To test how quickly the algorithms adapt to changes, we monitored them during sudden changes in equipment availability. It turned out that PSO, after about 20 additional iterations after the "failures" and a doubling of the demand for ventilators, again found solutions very close to the minimum Z. GA reacted a little slower: compensating for the system failures took 30-35 iterations. The ACO turned out to be even more inert: its period of "re-adaptation" reached almost 50 iterations, but it also came to an acceptable level.

The overall result of the comparative analysis confirms that neither GA, PSO, nor ACO provide a universally best solution for all scenarios. The choice of algorithm often depends on how quickly the answer is needed and how critical the exact minimization is. If the goal is emergency resource management and flexible decision restructuring, then PSO seems more attractive, especially at the initial stage of the crisis. If it is possible to increase the number of iterations, GA can match PSO in terms of results. At the same time, ACOs can be used at stages where it is crucial to optimize the supply order or where there is a complex combination of discrete choices.

IV. CONCLUSIONS

The article compares three heuristic algorithms (GA, PSO, and ACO) in the example of medical equipment distribution in a dynamic crisis. The proposed synthetic dataset (50 hospitals with uneven initial stock, 10 thousand patients with different levels of criticality, and two waves of crisis events) allows us to check how the algorithms cope with the growing demand for ventilators and the failure of a part of the MRI. The results show that PSO reaches an acceptable distribution level the fastest, GA provides a similar level of accuracy but is slower to respond to sudden changes, and ACO periodically gets stuck on local minimal solutions. At the same time, each approach has advantages: GA explores the solution space powerfully, PSO can adapt to changing conditions quickly, and ACO is sometimes more effective for combinatorial constraints.

In addition to traditional optimization methods, recent studies show that a promising area of research is the development of hybrid methods, when PSO performs the initial approximation, and the refinement is performed by a GA and data indexing in large systems [12, 13]. It is also advisable to introduce mechanisms for dynamically adjusting parameters (population size, inertia coefficients, pheromone intensity) depending on the system's current state. Actual practice shows that crises can develop rapidly, requiring constant updating of resource allocation plans and flexible strategies for their implementation [14-16]. Using heuristic algorithms in such conditions allows for more efficient control of equipment and timely response to critical situations.

REFERENCES

[1] F. Parker, H. Sawczuk, F. Ganjkhanloo, F. Ahmadi, and K. Ghobadi, “Optimal Resource and Demand Redistribution for Healthcare Systems Under Stress from COVID-19,” arXiv:2011.03528 [cs, math], Nov. 2020, Available: https://arxiv.org/abs/2011.03528

[2] S. E. Griffis, J. E. Bell та D. J. Closs, “Metaheuristics in Logistics and Supply Chain Management”, Journal of Business Logistics, vol. 33, no. 2, pp. 90–106, Jun. 2012, doi: https://doi.org/10.1111/j.0000-0000.2012.01042.x.

[3] A. Lameesa, M. Hoque, M. SB Alam, S. F. Ahmed, and A. H. Gandomi, “Role of metaheuristic algorithms in healthcare: A comprehensive investigation across clinical diagnosis, medical imaging, operations management, and public health”, Journal of Computational Design and Engineering, vol. 11, no. 3, pp. 223–247, Jun. 2024, doi: https://doi.org/10.1093/jcde/qwae046.

[4] S. M. Almufti, A. Yahya Zebari, and H. Khalid Omer, “A comparative study of particle swarm optimization and genetic algorithm,” Journal of Advanced Computer Science & Technology, vol. 8, no. 2, p. 40, Oct. 2019, doi: https://doi.org/10.14419/jacst.v8i2.29401.

[5] Y. -J. Gong et al., "An Efficient Resource Allocation Scheme Using Particle Swarm Optimization," in IEEE Transactions on Evolutionary Computation, vol. 16, no. 6, pp. 801-816, Dec. 2012, doi: 10.1109/TEVC.2012.2185052.

[6] J. Kim and S. Yoo, “Software review: DEAP (Distributed Evolutionary Algorithm in Python) library,” Genetic Programming and Evolvable Machines, vol. 20, no. 1, pp. 139–142, Nov. 2018, doi: https://doi.org/10.1007/s10710-018-9341-4.

[7] W. F. Abd-El-Wahed, A. A. Mousa, and M. A. El-Shorbagy, “Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems,” Journal of Computational and Applied Mathematics, vol. 235, no. 5, pp. 1446–1453, Jan. 2011, doi: https://doi.org/10.1016/j.cam.2010.08.030.

[8] Z. Ghaffari, M. M. Nasiri, A. Bozorgi-Amiri, and A. Rahbari, “Emergency supply chain scheduling problem with multiple resources in disaster relief operations,” Transportmetrica A: Transport Science, vol. 16, no. 3, pp. 930–956, Jan. 2020, doi: https://doi.org/10.1080/23249935.2020.1720858.

[9] L. James V. Miranda, “PySwarms: a research toolkit for Particle Swarm Optimization in Python,” The Journal of Open Source Software, vol. 3, no. 21, p. 433, Jan. 2018, doi: https://doi.org/10.21105/joss.00433.

[10] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 26, no. 1, pp. 29-41, Feb. 1996, doi: 10.1109/3477.484436.

[11] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," in IEEE Transactions on Evolutionary Computation, vol. 7, no. 5, pp. 445-455, Oct. 2003, doi: 10.1109/TEVC.2003.817236.

[12] I. Kyrychenko, G. Tereshchenko, K. Smelyakov, “Optimized Indexing Method in a Hybrid Image Storage Model for Efficient Storage and Access in Big Data Environments,” 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), pp. 1–4, Oct. 2024, doi: https://doi.org/10.1109/tcset64720.2024.10755763.

[13] A. F. Ali and M. A. Tawhid, “A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems”, Ain Shams Engineering Journal, vol. 8, no. 2, pp. 191–206, Jun. 2017, doi: https://doi.org/10.1016/j.asej.2016.07.008

[14] A. B. Eisman, B. Kim, R. G. Salloum, C. J. Shuman, and R. E. Glasgow, “Advancing rapid adaptation for urgent public health crises: Using implementation science to facilitate effective and efficient responses,” vol. 10, Aug. 2022, doi: https://doi.org/10.3389/fpubh.2022.959567.

[15] K. Smelyakov, D. Tovchyrechko, I. Ruban, A. Chupryna and O. Ponomarenko, "Local Feature Detectors Performance Analysis on Digital Image," 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 644-648, doi: 10.1109/PICST47496.2019.9061331.

[16] J. Braithwaite, E. Hollnagel, and G. S. Hunte, “Methods and Solutions,” in Resilient Health Care, 1th ed. London: CRC Press, 2016, ch.9, pp.186-236.