Transport and Telecommunication, 2025, volume 26, no. 4, 386-392
Transport and Telecommunication Institute, Lauvas 2, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2025-0030

TRUCK SERVICE OPTIMISATION IN PORT AREAS BY COMPARING GENETIC ALGORITHM AND CUCKOO SEARCH ALGORITHM

Vassilios Kappatos, Evangelos D. Spyrou, Aggelos Aggelakakis, Maria Boile

Hellenic Institute of Transport, Centre for Research and Technology Hellas
Thessaloniki, Greece
{vkappatos, espyrou, agaggelak, boile}@certh.gr

Efficient truck service optimization in port areas is essential for minimizing congestion, reducing delays and improving overall logistics efficiency. This study presents a comparative analysis of Genetic Algorithm (GA) and Cuckoo Search Algorithm (CSA) for optimizing truck scheduling and resource allocation in dynamic port environments. GA explores a broad solution space using evolutionary operators, while CSA leverages Lévy flight-based search mechanisms to enhance solution quality and escape local optima. The comparison evaluates both algorithms based on key performance metrics, including waiting times and server utilization. Simulation results indicate that the GA produces smaller waiting times, while the CSA exhibits higher. Moreover, the server utilization of the GA is significantly lower than with the CSA. The findings highlight the strengths and limitations of each method, providing valuable insights into their applicability for real-time truck service optimization in smart port management systems.

Keywords: Trucks, port, queue, cuckoo search, genetic algorithm

1. Introduction

Queueing theory (Erlang, 2009) is a mathematical framework used to analyze the behaviour of waiting lines or queues in various systems, such as transportation (Mai and Cuong, 2021), (Rathore, 2022), (Varghese *et al*., 2021), telecommunications and networking (Afolalu *et al*, 2021), (Tahmasebi *et al*, 2021), (Mas, 2022), healthcare (Ala, 2023), (Peter and Sivasamy, 2021), (Vaghani *et al*, 2024) and manufacturing (Zhang *et al*, 2021), (May *et al*., 2021). Among its many models, the M/M/c queue (Goodarzi *et al*., 2022), (Gross *et al*., 2011) is a widely studied and applied system that characterizes a service facility with c parallel servers, where arrivals follow a Poisson process with rate λ and service times are exponentially distributed with rate μ. The notation M/M/c reflects this Markovian property of both arrivals and service times, combined with the number of servers. This model is particularly suitable for systems like ports (Neagoe *et al*., 2021), where trucks arrive randomly and multiple service bays operate simultaneously to unload or load goods. By providing analytical expressions for key performance metrics, such as waiting time, queue length and server utilization, the M/M/c queue helps decision makers optimize resource allocation and improve operational efficiency.

The primary challenge in an M/M/c queue system lies in balancing service efficiency against cost constraints. Adding more servers (c) reduces the waiting time for customers, but increases operational costs. On the contrary, fewer servers may result in long queues and delays, impacting service quality. A critical performance metric in this system is the average waiting time in the queue (Qq), which depends on the traffic intensity ($\rho = \lambda$) and the probability of an idle system (P0). The objective of the M/M/c queue formulation is to minimize Wq, by determining the optimal number of servers (c), ensuring that the system remains stable ($\rho < 1$), while meeting service level expectations. This optimization problem serves as the foundation for many real world applications, including port operations, hospital staffing and call centre management.

The key challenge in port operations is finding the right balance between the number of service bays and the demand for service. While increasing the number of bays can reduce truck waiting times, it also incurs higher operational costs. In contrast, limiting the number of bays might lead to longer queues and delays, impacting the throughput and reliability of the port. The M/M/c queue framework enables decision makers to address this trade-off by determining the optimal number of service bays required to meet service level goals without exceeding budget constraints. This approach ensures that resources are allocated efficiently, enhancing the overall productivity of the port, while minimizing the waiting time and operational costs associated with truck services.

This paper conducts a comparative study of the Genetic Algorithm (GA) (Dawid, 1999) and the Cuckoo Search Algorithm (CSA) (Yang, 2020) for optimizing truck scheduling and resource allocation in dynamic port environments. GA employs evolutionary operators to explore a wide solution space, whereas CSA utilizes Lévy flight-based search strategies to improve solution quality and avoid local optima. The performance of both algorithms is assessed using key metrics, such as truck waiting time and server utilization. Simulation results show that GA achieves shorter waiting times, while CSA tends to result in higher waiting durations. Om the other hand, CSA demonstrates more efficient server utilization compared to GA. These results underscore the trade-offs between the two approaches and offer practical insights into their suitability for real-time optimization in smart port management systems.

2. Related work

The subject of proposing solutions to optimize truck services operating in ports has been a focus of research in recent years. Various methods can be found in the international literature, utilizing modern technological tools, such as machine and deep reinforcement learning, specialized algorithm types and the use of neural networks.

One proposed methodology aimed at reducing traffic congestion and truck service delays at ports, includes the implementation of a dynamic Truck Appointment System, Machine Learning through the use of the Decision Tree algorithm that collects and processes data, incorporating smart technologies (GPS, RFID) and the development of a Discrete Event Simulation model. This methodology led to a 90.4% reduction in waiting time, as well as a significant decrease in queuing at port terminals (da Silva *et al.*, 2023).

Another methodological approach focuses on optimizing the scheduling of Yard Trucks in container terminals. Specifically, the objective is to reduce empty travel distances and simultaneously minimize the total time required for loading and unloading containers from vessels. This method develops a multi-objective mathematical model based on two contemporary techniques: the fuzzy membership function and the Pareto optimal solution method, followed by the application of a heuristic-adaptive version of a genetic algorithm (Hu *et al.*, 2019).

Further research effort includes the development of a dynamic truck routing system for optimizing container management in ports, aiming to reduce waiting times and increase productivity. This system utilizes Deep Reinforcement Learning (DRL), applying a spatial-attention-based algorithm trained through a neural network to make real-time routing decisions. The system is implemented due to a high-precision simulation developed within a customized Real2Sim framework that enables the integration of empirical knowledge (Jin *et al.*, 2024).

Deep Reinforcement Learning can also be applied in another innovative approach that combines it with genetic programming and hyper-heuristic methods. This methodology, known as DRL-GPEHH (Deep Reinforcement Learning Assisted Genetic Programming Ensemble Hyper-Heuristics), mainly aims to improve the effectiveness of dynamic truck routing in container terminals, addressing the complexity and uncertainty of real-world operating conditions (Chen *et al.*, 2024).

Another proposed methodology, called Variable Neighborhood Tabu Search (VNTS), uses a Variable Neighbourhood Search (VNS) algorithm, in combination with a Tabu List, an iterative local search and the parallelization of neighbour- hood solution generation process. This methodology aims to minimize the total distance traveled by trucks, taking into account constraints, such as time windows and open periodic vehicle routing, to reduce operation times and transport costs (Matijević *et al.*, 2023).

A different methodological approach aiming to enhance the operational efficiency of a terminal through timely truck scheduling, may incorporate Markov Decision Process (MDP) and a cooperative fleet scheduling algorithm, based on Deep Double Q-Networks (DDQN). The proposed methodological framework includes MDP and DDQN, the development of a terminal simulation model, the creation of an appropriate mathematical truck scheduling model and the representation of real terminal operations using graph theory (Cheng *et al.*, 2025).

The final proposed methodology is based on the application of a genetic algorithm to achieve optimal scheduling of truck operations within a container terminal (arrival, loading/unloading, departure). The algorithm is composed of an initial encoding scheme, a fitness function and a few genetic operators that ensure the viability of the proposed schedule. This methodology aims to minimize the maximum completion time of truck-related operations within the terminal, ultimately enhancing the efficiency of truck operations in the port and reducing their waiting times (Ng *et al.*, 2007).

3. M/M/c queue formulation for trucks in port area

The M/M/c queue represents a queueing system with the following characteristics of Poisson arrival process with rate λ (trucks per hour), exponential service time with rate μ (services per hour per server) and c parallel servers.

The traffic intensity is given by

$$\rho = \frac{\lambda}{c\mu}. \tag{1}$$

Here, ρ must satisfy $\rho < 1$ for the system to be stable.

The probability of Zero Trucks in the System (P_0) is given by,

$$P_0 = \left[\sum_{n=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \cdot \frac{1}{1-\rho}\right]^{-1}. \tag{2}$$

The average number of trucks in the queue (L_q) is given by,

$$L_q = \frac{\left(\frac{\lambda}{\mu}\right)^c \cdot \rho}{c! \cdot (1-\rho)^2} \cdot P_0. \tag{3}$$

The average waiting time in the queue (W_q) is given by,

$$W_q = \frac{L_q}{\lambda}. \tag{4}$$

The goal is to minimize the average waiting time in the queue (W_q) by optimizing the number of servers c. The optimization problem is formulated as:

$$\min_c W_q = \frac{\left(\frac{\lambda}{\mu}\right)^c \cdot \rho}{c! \cdot (1-\rho)^2} \cdot P_0 \cdot \frac{1}{\lambda}, \quad subject\ to\ \rho < 1. \tag{5}$$

4. Stochastic methods for optimization

Two stochastic methods are used to solve the optimization problem are a Genetic Algorithm and Cuckoo Search Algorithm.

4.1. Genetic algorithm (GA)

The GA is an evolutionary optimization method inspired by natural selection. It works with the following steps:

- **Step 1 – Initialization:** Generate a population of M initial candidate values for c
- **Step 2 – Fitness Evaluation:** Compute the fitness of each candidate c as:

$$Fitness\ (c) = \frac{1}{W_q}. \tag{6}$$

- **Step 3 – Selection:** Select parents for crossover based on their fitness scores.
- **Step 4 – Crossover:** Combine parent solutions to produce offspring:

$$c_{offspring} = \frac{c_{parrent1} + c_{parrent2}}{2}, \tag{7}$$

(rounded to the nearest integer within the range of c).

- **Step 5 – Mutation:** Introduce variability by randomly altering a small percentage of offspring values.
- **Step 6 – Replacement:** Form a new population by combining parents and offspring, retaining the most fit candidates.
- **Termination:** Repeat steps 2–6 for a predefined number of generations or until convergence.
- **Output:** Return the c with the highest fitness.

4.2. Cuckoo Search Algorithm (CSA)

The Cuckoo Search Algorithm, inspired by the brood parasitism behavior of cuckoos, uses Lévy flights for exploration. The steps are:

- **Step 1 – Initialization:** Generate an initial population of M nest solutions (c values).
- **Step 2 – Fitness Evaluation:** Evaluate the fitness of each c as:

$$Fitness\ (c) = \frac{1}{W_q}. \tag{8}$$

- **Step 3 – New Solutions via Lévy Flight:** Replace some nest solutions with new ones based on Lévy flights:

$$c_{new} \quad = c_{current} + \alpha L\acute{e}vy(\beta). \tag{9}$$

Here, α is the step size, and Lévy(β) follows a Lévy distribution.

- **Step 4 – Selection:** Retain the best solutions based on fitness.
- **Termination:** Repeat steps 2–4 for a predefined number of iterations.
- **Output:** Return the c with the highest fitness.

5. Results

The port is modeled as a system, where ships or trucks arrive randomly and are served by a fixed number of servers. Service speed changes over time due to operational adjustments and optimization algorithms simulate how managers improve efficiency to reduce waiting times. The results show how these adjustments affect both congestion levels and the overall use of port capacity over time.

The simulation model involves optimizing the service rate (μ) of a truck appointment system using GA and CSA. The arrival rate (λ) of trucks is fixed at 10 trucks per hour and the system is equipped with c = 4 servers. The service rate μ is optimized over 250 iterations. Initially, the GA uses a population of 10 random service rates between 2.5 and 4.5, and through crossover and mutation, it attempts to minimize the waiting time (W_q) by selecting the best 5 individuals from each generation. The mutation is performed with a probability of 0.2 and a random value within a range of ±0.2 is added to the selected solutions. In the CSA, the initial population consists of 10 random values between 2.5 and 4.5 for μ. The algorithm explores new solutions using a Levy flight step size, controlled by a random scaling factor and the best nests (solutions) are selected, after each iteration to minimize the waiting time.

Both algorithms track server utilization during the optimization process. The utilization is calculated as $\boldsymbol{Utilization} = \frac{\boldsymbol{\lambda}}{\boldsymbol{c\mu}}$ where λ is the arrival rate and c is the number of servers. The goal of the simulation is to decrease the waiting time W_q over iterations, while maximizing the server utilization, ensuring that the system operates efficiently. The GA performs this by refining the service rates through genetic operations, while the CSA employs an exploration-exploitation strategy to adjust service rates. The results are visualized through two plots: one comparing the waiting times of the two algorithms and the other showing the server utilization over iterations. Both algorithms aim to achieve higher utilization with lower waiting times, by adjusting the service rate μ within the defined boundaries.

In Figure 1, the reader can observe the waiting times of both algorithms over the iterations of the optimization process. Initially, both algorithms exhibit high waiting times, which decrease rapidly in the early iterations as the algorithms explore and converge toward optimal solutions. After this rapid decrease, the waiting times gradually level off to much lower values, approaching zero by the end of the 250 iterations, indicating that the system is becoming more efficient over time.

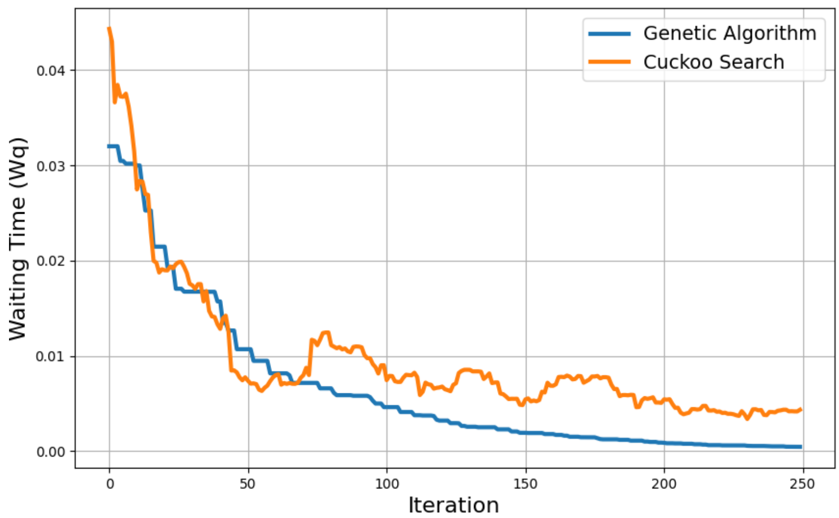

Figure 1. Waiting time over iterations

The GA shows a smoother decrease in the waiting time (W_q) as the optimization progresses. This suggests that the GA maintains a more steady improvement in the system's performance. In contrast, the CSA exhibits higher waiting times as the iterations proceed, suggesting that despite its exploration-exploitation mechanism, the CSA may not be converging to the optimal service rate as efficiently as the GA. The fluctuation in waiting times for the CSA could indicate that it is struggling to find the optimal balance between exploration and exploitation.

In Figure 2, the reader can observe the server utilization of both algorithms. The GA appears to decrease to very low utilization, suggesting underutilization of resources. This implies that as the GA optimizes the service rate (μ) over time, it leads to a configuration, where the system is not efficiently utilizing the available servers. This could be due to the GA converging to a suboptimal service rate that does not fully capitalize on the system's capacity.

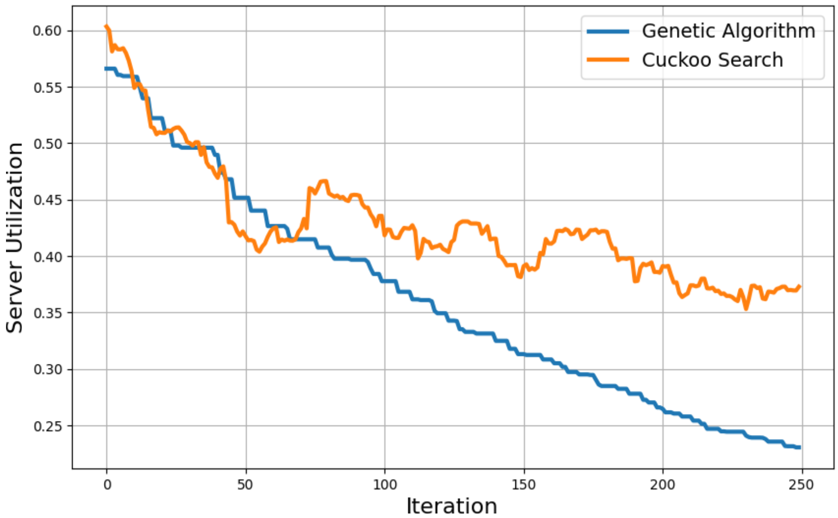

Figure 2. Server utilisation over iterations

On the other hand, the CSA exhibits higher utilization throughout the optimization process. The CSA's exploration-exploitation mechanism allows it to more effectively search for the optimal service rate that maximizes server utilization, while minimizing waiting time. As a result, the CSA is able to maintain a higher utilization of the servers, ensuring better resource allocation and overall system efficiency compared to the GA.

These findings highlight a trade-off between the optimization of waiting times and resource utilization, with the GA focusing more on minimizing waiting times, potentially at the expense of server utilization and the CSA providing a better balance between these two objectives.

6. Conclusions

This study presented a comparative evaluation of the Genetic Algorithm (GA) and Cuckoo Search Algorithm (CSA) for optimizing truck scheduling and resource allocation in dynamic port environments. Both algorithms demonstrated the capability to reduce waiting times over the course of the optimization process. The GA showed a consistent and smoother reduction in waiting times, indicating steady convergence toward efficient solutions. In contrast, the CSA exhibited greater fluctuations and higher waiting times, suggesting challenges in effectively balancing exploration and exploitation.

However, the comparative analysis of server utilization revealed a key trade-off. The GA, while minimizing waiting times, led to significant under-utilization of server resources, indicating potential inefficiencies in system throughput. Conversely, the CSA maintained higher levels of server utilization, reflecting its strength in optimizing resource allocation, albeit with less success in minimizing waiting times.

These results underscore a fundamental performance trade-off between minimizing queue delays and maximizing server efficiency. The GA excels in reducing waiting times, but may compromise resource utilization, while the CSA offers a more balanced approach by sustaining better server utilization at the cost of slightly higher waiting times. These insights are critical for smart port management systems that must prioritize, either swift service or efficient resource use depending on real-time operational goals.

Declaration of Generative AI and AI-assisted technologies in the writing process:

- During the preparation of this manuscript the author(s) used ChatGPT in order to improve and rephrase the text. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

1. Afolalu, S., Ikumapayi, O., Abdulkareem, A., Emetere, M., Adejumo, O. (2021) A short review on queuing theory as a deterministic tool in sustainable telecommunication system. *Materials Today: Proceedings*, 44(469), 2884–2888. DOI:10.1016/j.matpr.2021.01.092.
2. Ala, A., Yazdani, M., Ahmadi, M., Poorianasab, A., Yousefi Nejad Attari, M. (2023) An efficient healthcare chain design for resolving the patient scheduling problem: queuing theory and MILP-ASA optimization approach. *Annals of Operations Research*, 328(1), 3–33. DOI:10.1007/s10479-023-05287-5.
3. Chen, X., Bai, R., Qu, R., Dong, J. (2024) Deep reinforcement learning assisted genetic programming ensemble hyper-heuristics for dynamic scheduling of container port trucks. *IEEE Transactions on Evolutionary Computation* [Preprint]. DOI:10.1109/TEVC.2024.3381042.
4. Cheng, S., Liu, Q., Jin, H., Zhang, R. (2025) Collaborative optimization of truck scheduling in container terminals using graph theory and DDQN. *Scientific Reports*, 15(1), 6950. DOI:10.1038/s41598-025-91140-7.
5. Dawid, H. (1999) Genetic algorithms. *Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models*, 41–69. DOI: 10.1007/978-3-642-18142-9.
6. Erlang, A.K. (1909) The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik* [Preprint].
7. Goodarzi, A.H., Diabat, E., Jabbarzadeh, A., Paquet, M. (2022) An M/M/c queue model for vehicle routing problem in multi-door cross-docking environments. *Computers & Operations Research*, 138, 105513. DOI:10.1016/j.cor.2021.105513.
8. Gross, D., Shortle, J.F., Thompson, J.M., Harris, C.M. (2011) *Fundamentals of queueing theory*. John wiley & sons.
9. Hu, X., Guo, J. and Zhang, Y. (2019) Optimal strategies for the yard truck scheduling in container terminal with the consideration of container clusters. *Computers & Industrial Engineering*, 137, 106083. DOI: 10.1016/j.cie.2019.106083.
10. Jin, J., Cui, T., Bai, R., Qu, R. (2024) Container port truck dispatching optimization using Real2Sim based deep reinforcement learning. *European Journal of Operational Research*, 315(1), 161–175. DOI: 10.1016/j.ejor.2023.11.038.
11. Mai, N.T. and Cuong, D.M. (2021) The application of queueing theory in the parking lot: a literature review. In: *International conference on emerging challenges: Business transformation and circular economy (ICECH 2021).* Atlantis Press, 190–203. DOI:10.2991/aebmr.k.211119.020.
12. Mas, L, Vilaplana, J., Mateo Fornes, J., Jordi & Solsona, F. (2022) A queuing theory model for fog computing. *The Journal of Supercomputing*, 78(8), 11138–11155. DOI:10.1007/s11227-022-04328-3.
13. Matijević, L., DJurasević, M. and Jakobović, D. (2023) A Variable Neighborhood Search Method with a Tabu List and Local Search for Optimizing Routing in Trucks in Maritime Ports. *Mathematics*, 11(17), 3740. DOI: 10.3390/math11173740.
14. May, M.C., Fischer, M.D., Albers, A., Mayerhofer, F. (2021) Queue length forecasting in complex manufacturing job shops. *Forecasting*, 3(2), 322–338. DOI:10.3390/forecast3020021.
15. Neagoe, M., Hvolby, H.-H. and Turner, P. (2021) Why are we still queuing? Exploring landside congestion factors in Australian bulk cargo port terminals. *Maritime Transport Research*, 2, 100036.
16. Ng, W., Mak, K. and Zhang, Y. (2007) Scheduling trucks in container terminals using a genetic algorithm. *Engineering optimization*, 39(1), 33–47. DOI:10.1080/03052150600917128.
17. Peter, P.O. and Sivasamy, R. (2021) Queueing theory techniques and its real applications to health care systems–Outpatient visits. *International Journal of Healthcare Management* [Preprint]. May 201914(2). DOI:10.1080/20479700.2019.1616890.
18. Rathore, R. (2022) A study on application of stochastic queuing models for control of congestion and crowding. *International Journal for Global Academic & Scientific Research*, 1(1), 1–6. DOI: 10.55938/ijgasr.v1i1.6.
19. da Silva, M.R.F., Agostino, I.R.S. and Frazzon, E.M. (2023) Integration of machine learning and simulation for dynamic rescheduling in truck appointment systems. *Simulation Modelling Practice and Theory*, 125, 102747. DOI:10.1016/j.simpat.2023.102747.

20. Tahmasebi, A., Salahi, A. and Pourmina, M.A. (2021) Improvement of software-defined network performance using queueing theory: a survey. *Majlesi J. Telecommun. Devices*, 10(1), 33–43. DOI:10.52547/mjtd.10.1.33.
21. Vaghani, K., Thakkar, V., Vaghasiya, S., Thaker, J., Bhise, A. (2024) Implementation of Queuing Theory in Emergency Departments. In: Proceedings of 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, March 2024. IEEE, pp. 1–6. DOI:10.1109/IATMSI60426.2024.10503130.
22. Varghese, V., Verghese, V. and Chandran, A. (2021) Application of Queuing Theory in transportation. *Intern. J. Eng. Res. Techn*, 9, 55–58. DOI: 10.17577/IJERTCONV9IS06010.
23. Yang, X.-S. (2020) Nature-inspired optimization algorithms: Challenges and open problems. *Journal of Computational Science*, 46. DOI: 10.1016/j.jocs.2020.101104.
24. Zhang, H.-Y., Chen, Q., Smith, J., Mao, N., Liao, Y., Xi, Sh. (2021) Queueing network models for intelligent manufacturing units with dual-resource constraints. *Computers & Operations Research*, 129, 105213. DOI:10.1016/j.cor.2021.105213.