

Review of: "Solving Pallet loading Problem with Real-World Constraints"

Fuqiang Lu¹

¹ Yanshan University

Potential competing interests: No potential competing interests to declare.

The idea of this article is good. However, this article still need to be revised as follows:

1. The Introduction should be improved from two aspects, one is the value of the topic, the author should make an extend to explain the important value and the challenge of the problem; For the second, the main contribution of the paper should be addressed clearly.
2. The Literatures review need to be enhanced, especially on heuristic methods, and articles in recent years. References on pallet loading problem should be included. These articles may be helpful for improving this paper: a hybrid metaheuristic algorithm for the multi-objective location-routing problem in the early post-disaster stage, two-level principal-agent model for schedule risk control of IT outsourcing project based on genetic algorithm, simulated annealing genetic algorithm based schedule risk management of IT outsourcing project, 4pl routing problem using hybrid beetle swarm optimization, colony search optimization algorithm using global optimization.
3. Choose a range of algorithms with different approaches to solving the pallet loading problem. This could include heuristic-based algorithms, metaheuristics, exact algorithms, and any other relevant techniques. A diverse selection will help understand various solution strategies. Generate benchmark instances: Create a set of benchmark instances that represent different scenarios and complexities commonly encountered in logistics. These instances should cover a variety of transport unit sizes, pallet dimensions, and constraints to ensure a comprehensive evaluation.
4. Run each algorithm on the benchmark instances and record their performance according to the defined metrics. This empirical evaluation will provide objective data to compare the algorithms under consistent conditions. Statistical analysis: Perform statistical analysis on the obtained results to identify significant differences between the algorithms. Use appropriate tests like t-tests or ANOVA to determine if certain algorithms outperform others with statistical significance.
5. Using excessive "if" and "else" statements in algorithm descriptions can lead to code that is hard to read, understand, and maintain. It can also make the algorithm more prone to bugs and errors. To improve the clarity and efficiency of the algorithm description, consider the following suggestions: Implement lookup tables: If the algorithm involves repetitive decision-making based on specific inputs, consider using lookup tables or dictionaries to store precomputed results. This approach can improve the algorithm's performance and reduce the need for lengthy condition checks. Avoid deeply nested conditions: Excessive nesting of "if" and "else" statements can make code difficult to follow. Aim to keep the nesting level shallow and consider refactoring the code if it becomes too deeply nested.
6. Please number all of the formulas and figures in the paper.

7. There are only 15 references.