Final Report

MSCI 703, Prof. Elhedhli

Ali Rafieepouralavialavijeh, 20900871

Problem statement:

Cutting and packing problems have numerous applications, spanning from the direct use of the models (loading cargo into ships, vehicles, containers) to a more abstract use of the models (scheduling problems, budgeting, generation of valid inequalities) [1]. Due to many applications, different variants of the issues have been developed based on the additional constraints present in the concrete application. However, in this project, the formulation proposed by [2] is considered and defined as follows.

[Main Problem]:

$$\min h$$

$$s.t.$$

$$\sum_{s=x,y,z}(p_{ij}^s + p_{ji}^s) \geq 1, \qquad \forall j > i \qquad (1)$$

$$p_{ij}^s + p_{ji}^s \leq 1, \qquad \forall j > i, s \qquad (2)$$

$$c_i^s + l_i^s \leq c_j^s + L^s\left(1 - p_{ij}^s\right), \qquad \forall i \neq j, s \qquad (3)$$

$$0 \leq c_i^s \leq L^s - l_i^s, \qquad \forall i \neq j, s \qquad (4)$$

$$c_i^z + l_i^z \leq h, \qquad \forall i \qquad (5)$$

$$p_{ij}^s \in \{0,1\}, \qquad \forall i \neq j, s \qquad (6)$$

$$h \geq 0 \qquad (7)$$

The three-dimensional bin-packing problem is still one of the most challenging optimization problems to solve, despite its wide range of applications,. Currently, medium to large-size instances is only translated heuristically and remain out of reach of exact methods. Various approaches have been proposed in the literature because of the typically long computational times of exact methods, including random search, guided local search, prototype column generation, extreme point, finite enumeration, and tabu search heuristics.

I will first propose a Lagrangian relaxation [3] approach presented with the sub-gradient method and cutting-plane method along with a heuristic algorithm to produce feasible solutions. Also, to show the applicability of metaheuristic algorithms, genetic algorithms, and simulated annealing algorithms are applied to several randomly generated instances.

Lagrangian relaxation:

Cutting-plane method:

Here, constraints (3) are the coupling constraints between the problems decision variables and, as such, are relaxed and multiplied by their Lagrange multiplier μ_{ijs}, which as a result lead to the following formulations:

[SP1]:

$$\min \sum_i \sum_j \sum_s \mu_{ijs}(L^s(p_{ij}^s))$$

$s.t.$

$$\sum\nolimits_{s=x,y,z}(p_{ij}^s + p_{ji}^s) \geq 1, \qquad \forall j > i \qquad (1)$$

$$p_{ij}^s + p_{ji}^s \leq 1, \qquad \forall j > i, s \qquad (2)$$

$$p_{ij}^s \in \{0,1\}, \qquad \forall i \neq j, s \qquad (6)$$

[SP2]:

$$\min h + \sum_i \sum_j \sum_s \mu_{ijs}(c_i^s - c_j^s)$$

$s.t.$

$$0 \leq c_i^s \leq L^s - l_i^s, \qquad \forall i \neq j, s \qquad (4)$$

$$c_i^z + l_i^z \leq h, \qquad \forall i \qquad (5)$$

$$h \geq 0 \qquad (7)$$

The lower bound for our problem can now be defined as follows:

$$LB = Z_{sp1} + Z_{sp2} + \sum_i \sum_j \sum_s \mu_{ijs}(l_i^s - L^s)$$

Below, the Lagrangian Dual Problem formulation for our problem is presented:

[LDP]:

$$max_{\mu_{ijs}}^{Z_{sp1}+Z_{sp2}+\sum_i \sum_j \sum_s \mu_{ijs}(l_i^s - L^s)}$$

Below, the Master Problem is defined, which iteratively solves the subproblem to add cuts to its problem and find optimal Lagrange multipliers:

[MP]:

$$max_{\mu_{ijs}}^{\theta_1+\theta_2+\sum_i \sum_j \sum_s \mu_{ijs}(l_i^s - L^s)}$$

$s.t.$

$$\theta_1 - \sum_i \sum_j \sum_s \mu_{ijs}(L^s(p_{ij}^s)) \leq 0$$

$$\theta_2 - h + \sum_i \sum_j \sum_s \mu_{ijs}(c_i^s - c_j^s) \leq 0$$

$$\mu_{ijs} \geq 0$$

Solving this problem iteratively results in the unboundedness of the master problem in early steps. As such, a predefined upper bound is set for each μ_{ijs} so the following results associated with the LR method are computed with a 25000 upper bound, resulting from trial and error.

Subgradient method:

To find the Lagrangean multipliers that yield the sharpest bound, we need to solve the Lagrangean dual problem. However, since it is non-differentiable at points where it has multiple optimal solutions subgradient method could be helpful. In this method, the procedure starts with an initial solution μ_0, then iteratively, update this solution, and step by step, gets closer to a solution. For that, there is a need for a direction that is calculated through the subgradient method. The final solution gives a lower bound to the minimization problem and an upper bound to the maximization problem.

Lagrangian Heuristic:

Given the responses achieved from the Lagrangian relaxation (LR) method are infeasible to the main problem, in this section a heuristic method is proposed to find both a feasible solution and an upper bound to the main problem.

Heuristic Solution Methodology:

Given the LR solution available, we can start to search its neighborhood, starting with our current infeasible solution. The objective function of this search is to make sure boxes have no overlap with each other and height is minimized. (1) We start by clustering our boxes into a minimum number of clusters using the K-means and Elbow method. There are two objectives for this. First, the motivation behind clustering the boxes is to add boxes that are similar iteratively. Second, solving instances with a smaller number of boxes is more manageable than solving many of them together. At the cost of losing some flexibility, by clustering we could iteratively solve the 3DBPP in a timelier manner compared to the exact methods. (2) For the local search algorithm, Simulated annealing is applied to the problem. This is because the objective is to produce a reasonable upper bound to the main issue, and we already have a starting point (solution of the LR method). Below, three slightly different approaches to producing an upper bound are defined.

<u>Approach 1:</u>

1) Preprocess the data.
    - *Find the optimal number of clusters for all existing boxes using the Elbow method*
    - *Find the groups of boxes using the K-means method*
2) For each of the groups found:
    - *Start the SA local search algorithm and find the solutions for each group*
    - *Stack each of the bins found so far to make an upper bound*

<u>Approach 2:</u>

1) Preprocess the data.
    - *Find the optimal number of clusters for all existing boxes, using the Elbow method*
    - *Find the groups of boxes using the K-means method*
2) For each of the groups found:
    - *Start the SA local search algorithm and find the solutions for each group fixing the position of other previous groups over time.*

<u>Approach 3:</u>

1) Considering all the boxes:
   - *Start the SA local search algorithm to minimize the overlap between the boxes*

Computational Results:

Given the mathematical formulations after splitting the main problem into two subproblems, the result of the master problem could become unbounded; as such, based on the specifics of the situation, certain bounds are put on the Lagrange multipliers to mitigate this issue. In this part, computational results are provided to compare the performance and quality of the lower and upper bounds found so far.

All instances are randomly generated with a maximum box size of 59 on each of the x, y, or z directions. The bin, or in other words, the pallet, has a size of 59*59*infinity.

Computational results show the inability of [Approach 3] in converging to zero overlaps between the boxes when the number of the boxes grows; thus, the results are shown in the table below. The initial heat is set to be 250000, and the heat reduction is set to be geometrical based on trial and error. The justification of the SA algorithm and its computational results will be provided in the solution methodologies section.

Since the SA algorithm fails to provide a feasible solution for more significant instances, their results are omitted. The subgradient method converges very fast in this problem; most of them take between 3-4 steps maximum to converge.

Number of boxes	**Lower bound cutting-plane method**	**Lower bound subgradiant method**	**Exact objective value**	**Upper bound (K-Means+SA)**	**Upper bound (K-Means+Exact)**
5	59	59	143	219	**-**
10	59	59	200	379	235
15	59	59	314	614	349
20	59	59	Couldn’t be solved in a timely manner	770	666

Figures 1 to 4 below demonstrate how each of the heuristic methodologies is compared with exact methods and Lagrangian relaxation solution. This example is done with ten randomly generated boxes.

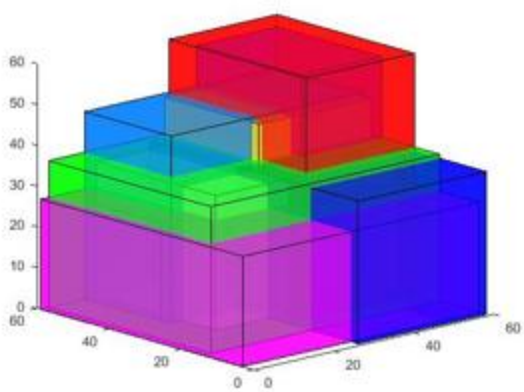

Figure 1 Lagrangian relaxation result

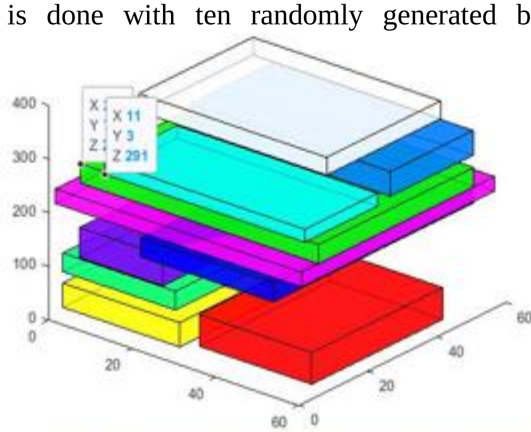

Figure 2 Illustration of the first heuristic method with k-means and SA algorithm

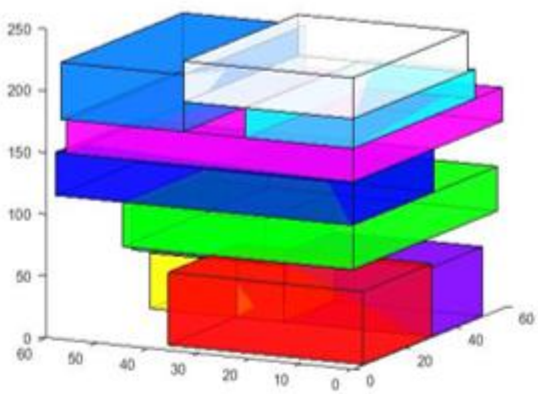

Figure 3 Illustration of the second heuristic method with k-means and exact solution for each cluster

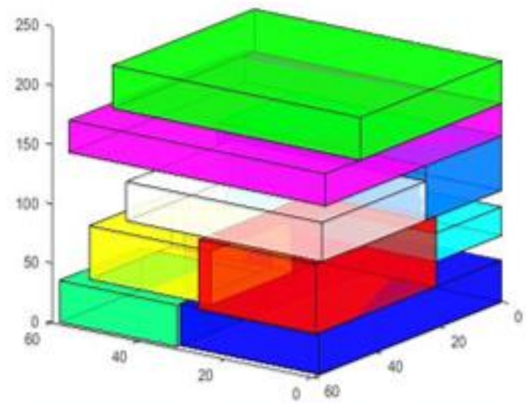

Figure 4 Illustration of an exact solution to the 10-boxes problem

Other solution methodologies:

Genetic algorithm:

A genetic algorithm (GA) is a method for solving constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution [4]. For this project, I have used various breeding strategies to find an optimal solution with various hyper-parameter instructions. These hyper-parameters included the number of iterations, population size, mutation rate, and elite population size. I propose three different mating strategies to maintain the boxes are within the pallet boundary, and the algorithm randomly chooses which one to use in the process. [For more details, you can refer to “mating function” in the codes attached]

Through trial-and-error, these parameters are set to be 100 population size, 1000 iteration numbers, mutation rate 10%, and elite population size of 30. The figure below shows the Genetic Algorithm running for a 25-instance sample with 10000 iterations, a population size of 250, and an elite size of 50. The table below shows some computational results as well.

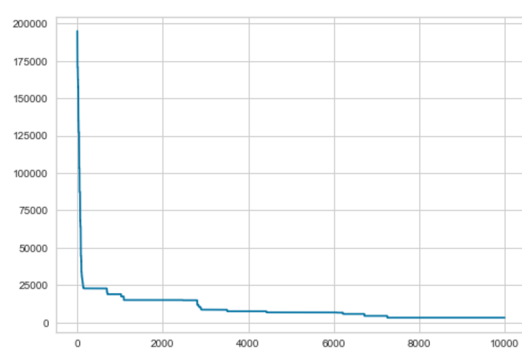

Figure 5 10000 iterations of GA and the cost of overlap in each of the time slots are depicted

Number of instances	Genetic algorithm best solution	Number of iterations	Time elapsed
5	0	12	1.61 s
10	~2500	1000	163.24 s
15	568.72	1000	300.83 s
20	3945.92	1000	559.06 s

Simulated Annealing algorithm:

Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in an ample search space for an optimization problem. SA is used both as a general solution methodology. As a part of the heuristics proposed in the project, the Figures below show the fluctuations of the optimization process and the best possible solutions found over time. Besides, the following table shows computational results for different randomly generated instances.

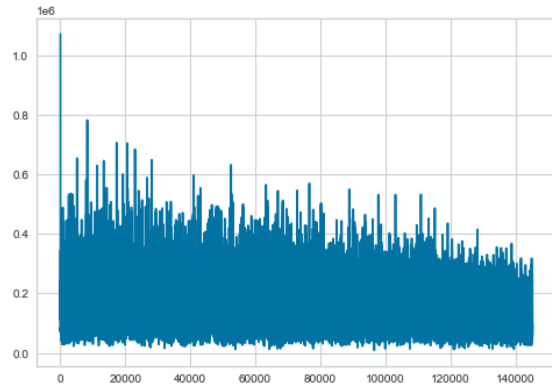

Figure 6 Fluctuations in the SA algorithm solutions

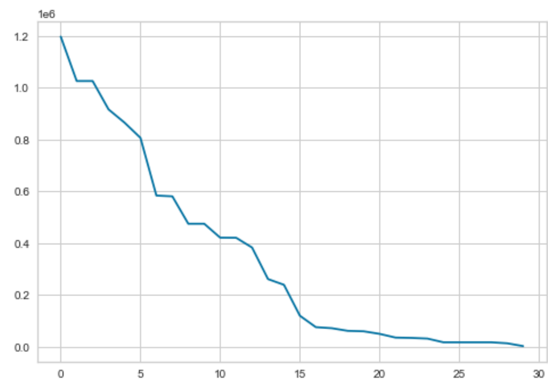

Figure 7 Fluctuations in the best solutions found in the SA

Number of boxes	Time	Best penalty after the random restart	Random restart results	Number of iterations
5	3.3449 s	0	-	10298
10	1564.83 s	2430	[2430, 9155, 7290, 7264, 5975]	1.75 million
15	3083.42 s	26963	[34238, 30858, 26963, 34669, 35717]	1.75 million
20	8297.05 s	55556	[61689, 59100, 57757, 60310, 55556]	1.75 million

Conclusion:

In this report, I have tried many methodologies, including Lagrange relaxation (cutting plane and subgradient method), Lagrangian heuristic (3 heuristics are proposed, and the results are demonstrated), Simulated Annealing, and Genetic Algorithm. Many instances are solved and depicted to show the applicability of each of the methods. Computational results show the SA algorithm doing approximately similar to GA but taking a couple of times more time to converge to optimal or a good solution after many iterations. Also, it is found that neither SA nor GA algorithms can find practical solutions unless heuristics are added to them to direct and guide the search.

Bibliography

[1] D. P. M. Z. Oluf Faroe, "Guided local search for the three-dimensional bin-packing problem," *Informs journal on computing,* vol. 15, no. 3, pp. 268-283, 2003.

[2] S. E. B. C. Y. Fatma Gzara, "The Pallet Loading Problem: Three-dimensional bin packing with practical constraints,," *European Journal of Operational Research,* pp. Pages 1062-1074, 2020.

[3] M. L. Fisher, "An Applications Oriented Guide to Lagrangian Relaxation," *Interfaces,* vol. 15, no. 2, pp. 10-21, 1985.

[4] L. D. Whitley, "Genetic algorithm," *Evolutionary algorithms and neural networks,* pp. 43-55, 2019.