Advanced Process Systems Engineering
Spring 2019
Final project

Romeo Valentin (*romeov*)
Yunshan Liu (*yunshanl*)

May 7th, 2019

Abstract

Generalized Disjunctive Programming is a modeling formulation similar to Mixed Integer Nonlinear Programming, but additionally allowing 'choices' between different constraints. In this project, we investigate the usage of GDP in modeling of strip-packing problems. We consider the effect of different formulations for the objective function and consider different model formulations, including packing 2D and 3D problems as well as rotating and fixed boxes.

1.1 Introduction

The strip-packing problem is a certain type of optimization problem where the goal is to fit a certain set of 'boxes' into a given space, where the space again is constrained by orthogonal boundaries. In this case in particular we will the boxes as rectangles for the two-dimensional and cuboids for the three-dimensional case.

Traditionally, strip-packing problems are formulated either as Mixed-Integer Linear Programs (MILP) or Mixed-Integer Nonlinear Programs (MINLP). For this project, we will investigate how we can instead use Generalized Disjunctive Programming (GDP) to model the problem in a more intuitive way. Additionally, for different problem formulations different objective functions can be considered. We will present different penalty functions like the $\|\cdot\|_1$-norm, the $\|\cdot\|_2$-norm and the $\|\cdot\|_\infty$-norm and the effects on solution and solving time.

This project is based on a basic GDP strip-packing formulation presented by Grossmann and Trespalacios [1]. This formulation only presents a simple two-dimensional

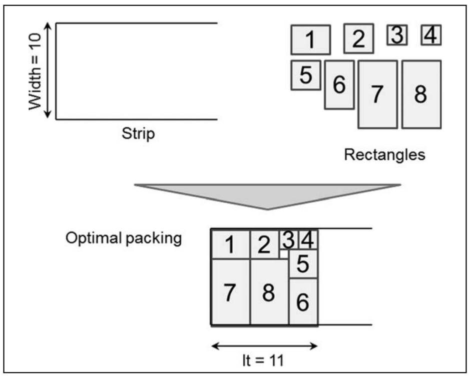


Figure 1.1: An example 2D single objective strip-packing problem, taken from [1].

example with a single objective direction. The formulation is

$$\begin{aligned}
&\min_{x,y} lt \\
s.t.\quad & lt \geq x_i + L_i && i \in N \\
&\begin{bmatrix} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{bmatrix} \veebar \begin{bmatrix} Y_{ij}^2 \\ x_j + L_j \leq x_i \end{bmatrix} \cdots \\
\veebar & \begin{bmatrix} Y_{ij}^3 \\ y_i + H_i \geq y_j \end{bmatrix} \veebar \begin{bmatrix} Y_{ij}^4 \\ y_j + H_j \geq y_i \end{bmatrix} && i,j \in N, i < j \\
& 0 \leq y_i \leq W - H_i && i \in N \\
& Y_{ij}^1 \veebar Y_{ij}^2 \veebar Y_{ij}^3 \veebar Y_{ij}^4 && i,j \in N, i < j \\
& lt, x_i, y_i \in \mathbb{R}^1, Y_{ij} \in \{True, False\} && i,j \in N, i < j,
\end{aligned} \tag{2D single objective}$$

where (x_i, y_i) denote the position of the lower left corner of each rectangle, W denotes the space height and (L_i, H_i) denote the width and height of each rectangle. An illustration can be seen in fig. 1.1. Note in particular that we only optimize in a single direction x, since we fix the direction y with $y_{max} = W$. Consider now a case in which we don't fix the height, instead optimizing in two directions at once. The formulation becomes

$$\begin{aligned}
&\min_{x,y} \left\| \begin{bmatrix} lt \\ ht \end{bmatrix} \right\|_p \\
s.t.\quad & lt \geq x_i + L_i && i \in N \\
& ht \geq y_i + H_i && i \in N \\
&\begin{bmatrix} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{bmatrix} \veebar \begin{bmatrix} Y_{ij}^2 \\ x_j + L_j \leq x_i \end{bmatrix} \cdots \\
\veebar & \begin{bmatrix} Y_{ij}^3 \\ y_i + H_i \geq y_j \end{bmatrix} \veebar \begin{bmatrix} Y_{ij}^4 \\ y_j + H_j \geq y_i \end{bmatrix} && i,j \in N, i < j \\
& Y_{ij}^1 \veebar Y_{ij}^2 \veebar Y_{ij}^3 \veebar Y_{ij}^4 && i,j \in N, i < j \\
& lt, x_i, y_i \in \mathbb{R}^1, Y_{ij} \in \{True, False\} && i,j \in N, i < j.
\end{aligned} \tag{2D multi objective}$$

Notice that the objective function now consists of two terms lt and ht, i.e. the width and height of the combined rectangles. In trade, we relax the height constraint $0 \leq y_i \leq W - H_i$. Choices for the value p in the objective function usually include $p = 1$ (absolute value norm), $p = 2$ (euclidean norm) or $p = \infty$ (max norm). We consider all three cases and show the effects. Notice that if we choose $p = 1$ or $p = \infty$, the formulation becomes linear (see section 1.3), whereas $p = 2$ yields a nonlinear formulation.

Furthermore, we can allow for rotation of the boxes as well. As we will see in section 1.2, the packing problem often leaves a lot of empty space because packages can't be rotated but instead are fixed with their given dimensions. We will introduce a way that we can model rotation using disjunctions.

Finally, we can transfer the ideas to a three-dimensional structure by adding additional disjuncts to each disjunction. Further details on the three-dimensional structure will be given in section 1.4.

1.2 The two dimensional strip-packing problem with rotation

Consider again eq. (2D single objective) and notice that in this formulation, W, L_i and H_i are parameters provided as input. The problem with this formulation is that it will often leave a lot of unused space as we can see in fig. 1.2 (a). We can solve this problem by allowing for rotation of the boxes, just like one would do in real life. To model rotation, we let L_i and H_i now be variables instead parameters and $\hat{L}_i$ and $\hat{H}_i$ be the provided parameters. In order to assign the new variables to the parameters, we can use a disjunction

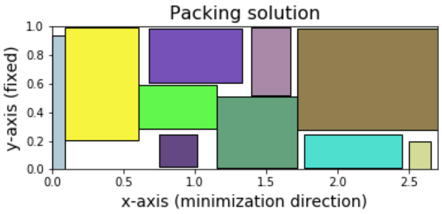
$$\begin{bmatrix} Y_{i1} \\ H_i = \hat{H}_i \\ L_i = \hat{L}_i \end{bmatrix} \veebar \begin{bmatrix} Y_{i2} \\ H_i = \hat{L}_i \\ L_i = \hat{H}_i \end{bmatrix} \quad \forall i \in N$$

so that for each box with index i we can choose to rotate it by 90° in respect to the input parameters. This fairly simple addition can reduce the unused space by a lot, as seen in section 1.2 (b). Just by visual inspection it is clear that there is almost no way the solution could be improved in any way because the unused spaces are very small.

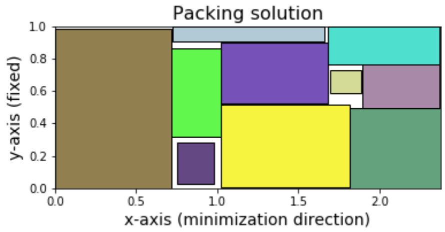
1.3 Transforming $\|\cdot\|_1$- and $\|\cdot\|_\infty$-norm to linear constraints

Unlike the euclidean norm $\|y\|_2 = \sqrt{\sum_l (y_l)^2}$, which is inherently nonlinear, the 1-norm and the ∞-norm in the objective function can be reformulated as a linear program by introducing auxiliary variables and a set of corresponding constraints.

Let's first consider the objective min $\|y\|_1$. Recall the definition of the 1-norm $\|y\|_1 = \sum_{1 \leq j \leq n} |y_j|$. To minimize each element individually, n auxiliary variables ξ_j are introduced such that $|y_j| \leq \xi_j$. After adding these constraints to the problem, the objective function can be rewritten as min $\sum \xi_j$. Finally, each constraint $|y_j| \leq \xi_j$ can be written as a set of two linear inequalities $-\xi_j \leq y_j \wedge y_j \leq \xi_j$. The new problem



(a) Two-dimensional strip packing problem without rotation.



(b) Two-dimensional strip packing problem with rotation.

Figure 1.2: Comparison between strip packing solutions with and without rotation. As we can see, **(a)** does not seem very tight whereas **(b)** has almost no more free space.

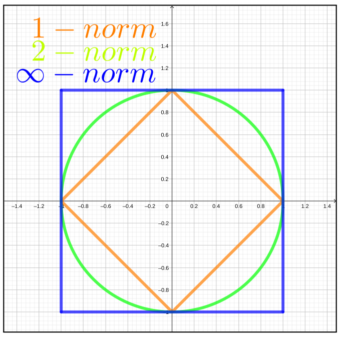
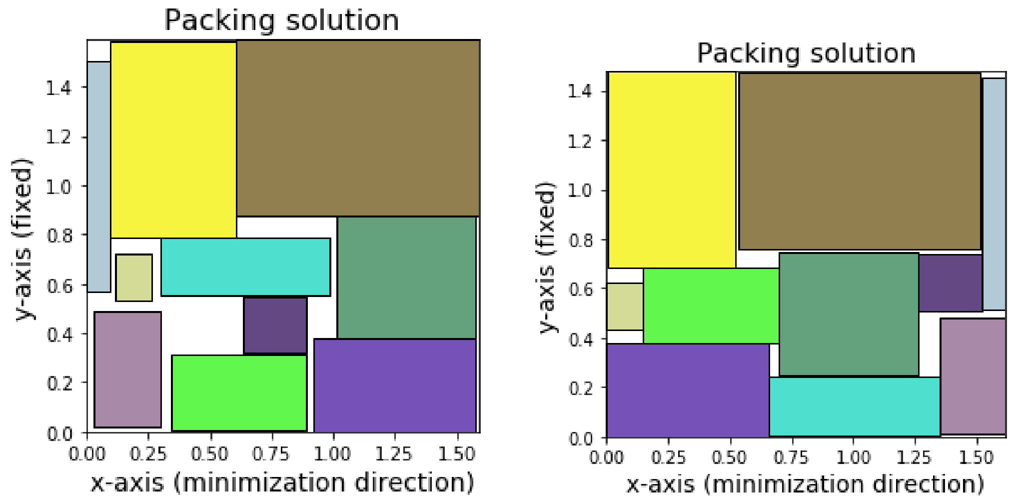


Figure 1.3: Comparison of $\|\cdot\|_p = 1$ for different values for p in a simple two-dimensional case.

formulation is

$$\begin{aligned} \min \quad & \sum \xi_j \\ s.t. \quad & -\xi_1 \leq y_1 \\ & y_1 \leq \xi_1 \\ & \vdots \\ & -\xi_n \leq y_n \\ & y_n \leq \xi_n \end{aligned}$$

in addition to any other constraints that might be given. This formulation requires an additional N continuous variables and $2N$ additional constraints with N the number of variables in the norm. The advantage of the formulation is that the resulting problem is linear and can be solved with a MILP solver (in contrast to the euclidean norm that yields a MINLP). Additionally, all variables are optimized at the same time (in contrast to the ∞-norm, that only minimizes the maximum absolute value one at a time).



(a) Two-dimensional strip packing problem with the $\|\cdot\|_1$ norm.

(b) Two-dimensional strip packing problem with the $\|\cdot\|_\infty$ norm.

Figure 1.4: Comparison between strip packing solutions with multiple objective directions and different norms. We can see that for this problem the $\|\cdot\|_\infty$ norm seems the yield the more reasonable result.

Next, we discuss the reformulation of min $\|y\|_\infty = \min \max_i \{|y_1|, \ldots, |y_m|\}$. Again an auxiliary variable is introduced, but this time it is only one variable ξ that bounds each element y_i with $|y_i| \leq \xi$. Thus, by minimizing ξ, only the maximum absolute value is minimized. The problem formulation becomes

$$
\begin{aligned}
& \min \xi \\
s.t. \quad & -\xi \leq y_1 \\
& y_1 \leq \xi \\
& \vdots \\
& -\xi \leq y_n \\
& y_n \leq \xi
\end{aligned}
$$

in addition to any other constraints. This formulation only introduces a single additional variable (as supposed to N for the 1-norm) and also introduces $2N$ additional constraints.

Figure 1.3 shows a simple comparison of the different norms. It is easily apparent that the 1- and ∞-norm can be modeled by linear constraints, but any other value p in between can not. Furthermore, the ∞-norm only penalizes the largest variable whereas the other norms reduce all variables at the same time.

1.4 Three dimensional problem

In real life, we usually come across problems in three instead of two dimensions. In this particular case, it is easy to imagine a problem involving boxes that need to be packed into a tight space, for example for a delivery service or a container ship.

In order to extend the formulation to the third dimension, we can just add another space variable D_i and the corresponding disjuncts for placement. The model formulation becomes

$$\begin{aligned}
&\min_{x,y} lt && \\
s.t.\quad & lt \geq x_i + L_i && i \in N \\
& \begin{bmatrix} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{bmatrix} \veebar \begin{bmatrix} Y_{ij}^2 \\ x_j + L_j \leq x_i \end{bmatrix} \dots && \\
& \veebar \begin{bmatrix} Y_{ij}^3 \\ y_i + H_i \geq y_j \end{bmatrix} \veebar \begin{bmatrix} Y_{ij}^4 \\ y_j + H_j \geq y_i \end{bmatrix} \dots && \\
& \veebar \begin{bmatrix} Y_{ij}^5 \\ z_i + D_i \geq z_j \end{bmatrix} \veebar \begin{bmatrix} Y_{ij}^6 \\ z_j + D_j \geq z_i \end{bmatrix} && i,j \in N, i < j \\
& 0 \leq y_i \leq W - H_i && i \in N \\
& 0 \leq z_i \leq V - D_i && i \in N \\
& Y_{ij}^1 \veebar Y_{ij}^2 \veebar Y_{ij}^3 \veebar Y_{ij}^4 \veebar Y_{ij}^5 \veebar Y_{ij}^6 && i,j \in N, i < j \\
& lt, x_i, y_i, z_i \in \mathbb{R}^1, Y_{ij} \in \{True, False\} && i,j \in N, i < j,
\end{aligned} \tag{3D single objective}$$

where D_i denotes the size in z-direction and V denotes the maximum length in z-direction. We can see that only two additional disjuncts and one global constraint per box have been added, so in general the problem size doesn't increase exponentially.

In order to add rotation to the three-dimensional model, we can again consider L_i, H_i and D_i to be variables, assigned to the parameters $\hat{L}_i$, $\hat{H}_i$ and $\hat{D}_i$, and add the disjunction

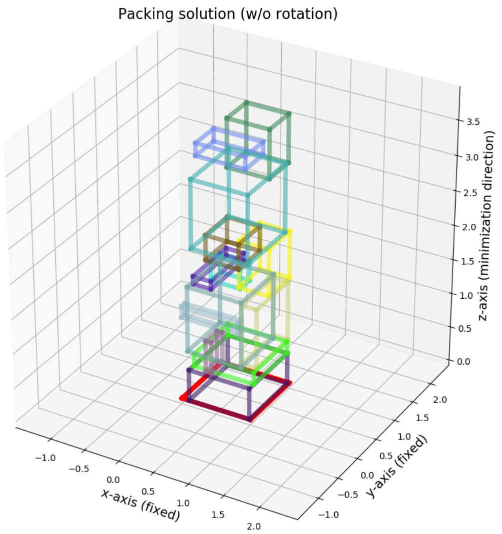
$$\begin{bmatrix} Y_{i1} \\ H_i = \hat{H}_i \\ L_i = \hat{L}_i \\ D_i = \hat{D}_i \end{bmatrix} \veebar \begin{bmatrix} Y_{i2} \\ H_i = \hat{L}_i \\ L_i = \hat{H}_i \\ D_i = \hat{D}_i \end{bmatrix} \veebar \begin{bmatrix} Y_{i3} \\ H_i = \hat{D}_i \\ L_i = \hat{L}_i \\ D_i = \hat{H}_i \end{bmatrix} \veebar \begin{bmatrix} Y_{i4} \\ H_i = \hat{H}_i \\ L_i = \hat{D}_i \\ D_i = \hat{L}_i \end{bmatrix} \veebar \begin{bmatrix} Y_{i5} \\ H_i = \hat{D}_i \\ L_i = \hat{H}_i \\ D_i = \hat{L}_i \end{bmatrix} \veebar \begin{bmatrix} Y_{i6} \\ H_i = \hat{L}_i \\ L_i = \hat{D}_i \\ D_i = \hat{H}_i \end{bmatrix} \quad \forall i \in N$$

for each of the boxes. As we can see the number of disjuncts in each disjunctions is the number of permutations in the dimensions, i.e. $(dim)!$. This means the problem can get significantly larger with more dimensions.

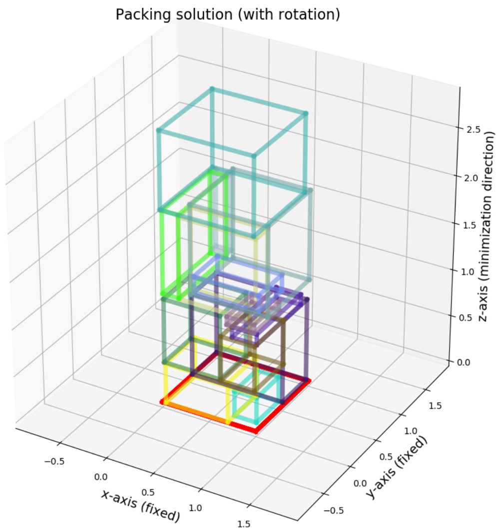
A visualization of the solution with and without rotation can be seen in fig. 1.5.

References

[1] Ignacio E Grossmann and Francisco Trespalacios. "Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming". *AIChE Journal* 59.9 (2013), pp. 3276–3295.



(a) Three-dimensional strip packing problem without rotation.



(b) Three-dimensional strip packing problem with rotation.

Figure 1.5: Comparison between strip packing solutions with and without rotation. Again, **(a)** does have a lot of space, especially for the slim boxes, whereas **(b)** has almost no more free space.