

Constraint Aggregation in Non-Linear Programming models for Nesting problems

Pedro Rocha, A. Miguel Gomes, Rui Rodrigues, Franklina Toledo, and Marina Andretta

Abstract This paper analyses the impact that aggregating constraints can achieve in the reduction of computational cost of a Non-Linear Programming model for Nesting problems with continuous rotations. This is achieved by aggregating non-overlapping constraints and using spatial partition and hierarchical overlap detection methods. When aggregating constraints there is also an effect of reducing the sensitivity of the solver, which may reduce the quality of the final layout. Analyzing the trade-off between constraints aggregation and the impact on the quality of the final solution is an important issue to handle nesting instances with a large number of pieces. Computational experiments show that aggregating non-overlapping constraints allows the Non-Linear Programming model for Nesting problems to scale well to tackle large size real world problems with continuous rotations.

1 Introduction

The Nesting problem is a complex problem that arises in industries where sets of pieces or space must be efficiently placed or allocated in order to minimize wasted space or wasted raw materials, without overlaps between pieces and fully contained inside a container. It is a 2D Cutting and Packing problem where pieces have non-regular geometries, also known as the Irregular Strip Packing problem.

The Nesting problem has a geometrical and combinatorial component, where the first impacts the approaches required to tackle the second. The selection of an adequate geometrical representation is an important issue to reduce the complexity

Pedro Rocha · A. Miguel Gomes · Rui Rodrigues
INESC-TEC, Faculdade de Engenharia, Universidade do Porto, e-mail: {pro10015, agomes, rui.rodrigues}@fe.up.pt

Franklina Toledo · Marina Andretta
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, e-mail: {fran, andretta}@icmc.usp.br

of the geometric component. The geometric representation should be able to deal with continuous rotations and allow easy overlap computations.

Bennell and Oliveira (2009) discuss several geometrical representations, each one tailored to a specific application: Grid representation for discrete position placement and orthogonal orientations; Polygonal representation (including No-Fit-Polygons) use polygons for continuous position placement and arbitrary discrete orientations; and Phi-Functions by (Stoyan et al., 2004) for continuous position and orientation placement of the pieces, using sets of primary pieces composed by straight lines and arc segments. An alternative representation is to approximate irregular pieces by a Circle Covering representation. Imamichi and Nagamochi (2007) uses a grid to place the circles, Zhang and Zhang (2009) uses a three-step algorithm to approximate pieces by circles, Jones (2013) uses a greedy heuristic to place the circles, and Rocha et al. (2013) presents a circle covering algorithm based on Medial Axis to minimize the number of circles needed to represent a piece.

Regarding Non-Linear Programming models (NLP) used with circle representation of pieces, Imamichi et al. (2009) proposed an Iterated Local Search algorithm where a NLP model is used to remove overlap between pieces. Stoyan et al. (2012) proposes a NLP mathematical model for strip packing supporting circles and non-convex polygons (used in conjunction with the Phi-Functions). Jones (2013) uses global optimization methods able to optimally solve small instances (4 to 6 pieces) using quadratic programming mathematical models. Rocha et al. (2014) explores the usage of a NLP model and several Circle Covering representations to tackle the Nesting problem with continuous rotations.

The use of NLP models in conjunction with a Circle Covering representation allows tackling the Nesting problem with continuous rotations. The main difficulty lies when dealing with large size instances, due to the large number of pieces which cause an exponential growth in the non-overlapping constraints derived from the comparisons between pairs of pieces. In this work we propose a method that aggregates non-overlapping constraints to reduce the computational cost without significant impact on the quality of the layout solutions, which allows NLP models to tackle larger instances. The proposed approach uses a complete circle covering representation present in Rocha et al. (2013) and introduces a pieces based NLP model, derived from the model presented in (Rocha et al., 2014).

2 Non-Linear Programming Model

A mathematical model based on a circle covering representation for the Nesting problem with continuous rotations requires NLP models due to the non-linearity of the distance computation and the trigonometric operations. In the proposed model, each piece is represented by a set of circles with fixed relative positions between each other. Three variables are needed for each piece, two for the placement position on the layout and one for the orientation. The model has a non-overlapping constraint for each pair of circles from different pieces and four Containment con-

straints for each circle, one for each side of the container. The mathematical formulation of the model is the following:

$$\begin{aligned}
& \text{minimize} && l && (1) \\
& \text{subject to:} && (R_{k_i} + R_{h_j})^2 - (x_k + \cos(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} - \\
& && \quad - x_h - \cos(A_{h_{0,j}} + \theta_h) \times D_{h_{0,j}})^2 - (y_k + \\
& && \quad + \sin(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} - y_h - \\
& && \quad - \sin(A_{h_{0,j}} + \theta_h) \times D_{h_{0,j}})^2 \leq 0, \\
& && \quad \forall i \in C_k, \forall j \in C_h, \forall k, h \in \mathbb{N}, k \neq h && (2) \\
& && x_k + \cos(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} + R_k - l \leq 0, \quad \forall i \in C_k, \forall k \in \mathbb{N} && (3) \\
& && R_k - x_k - \cos(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} \leq 0, \quad \forall i \in C_k, \forall k \in \mathbb{N} && (4) \\
& && y_k + \sin(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} + R_k - W \leq 0, \quad \forall i \in C_k, \forall k \in \mathbb{N} && (5) \\
& && R_k - y_k - \sin(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} \leq 0, \quad \forall i \in C_k, \forall k \in \mathbb{N} && (6) \\
& && x_k, y_k, \theta_k, l \in R && (7)
\end{aligned}$$

The Objective Function (1) aims to minimize the length of the strip and is represented by the auxiliary variable l . Each piece k is composed by a set of C_k circles and is represented by three variables: x_k and y_k defines the position of piece k of piece k on the layout and θ_k defines its orientation. The number of variables grow linearly with the number of pieces. The non-overlapping constraint (2) compare the distance between each pair of circles i, j from pieces k, h and the sum of radius of both circles R_{k_i} and R_{h_j} . The containment constraints (3–6) ensure that each circle does not exceed the admissible placement region. The number of non-overlapping constraints has a factorial growth, while the number of containment constraints grow linearly. Finally, the variable domains are defined in (7).

3 Aggregating Non-Overlapping Constraints

The NLP model presented in the previous section has difficulties in solving large instances due to high computational cost caused by the large number of non-overlapping constraints, which is caused by the increase in the number of pieces and in the number of circles necessary to represent them. To overcome this difficulty we propose to aggregate all non-overlapping constraints in a single summation constraint, where each summation term derives from constraint (2).

The aggregation of non-overlapping constraints allows discarding a large number of summation terms between distant pairs of pieces, by considering their terms zero (8). This is possible because the original non-overlapping constraint (2) will return a negative value. This also have the additional benefit of reducing the in-

ternal computational cost of the solver, when compared to the original model with independent non-overlapping constraints. Each summation term represents the difference between the squared sum of the radius of both circles R_{k_i} and R_{h_j} (9) and the squared distance between each pair of circles i, j from pieces k, h (10–11).

$$\sum_{i=0}^{C_k} \sum_{j=0}^{C_h} \sum_{k=0}^N \sum_{h=0}^N \{\max[0, \text{NOVLP}_R - (\text{NOVLP}_X + \text{NOVLP}_Y)]\}^2 \leq 0, \quad (8)$$

$$\forall i \in C_k, \forall j \in C_h, \forall k, h \in \mathbb{N}, k \neq h, x_{k_i}, y_{k_i}, \theta_k, l \in R$$

$$\text{NOVLP}_R = (R_{k_i} + R_{h_j})^2 \quad (9)$$

$$\text{NOVLP}_X = (x_{k_i} + \cos(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} - x_{h_j} - \cos(A_{h_{0,j}} + \theta_h) \times D_{h_{0,j}})^2 \quad (10)$$

$$\text{NOVLP}_Y = (y_{k_i} + \sin(A_{k_{0,i}} + \theta_k) \times D_{k_{0,i}} - y_{h_j} - \sin(A_{h_{0,j}} + \theta_h) \times D_{h_{0,j}})^2 \quad (11)$$

Summation terms are discarded by using spatial partition and hierarchical overlapping detection. Spatial partition divides the space into regions, and only compares pieces from the same or adjacent regions. The hierarchical overlap detection uses a three level representation, where the first level uses an orthogonal bounding box, the second level uses a minimum enclosing circle (MEC), and the last uses the circle covering. Comparisons start by using the basic piece representations and when it cannot discard overlap, it checks overlap with higher quality representations.

4 Results and Discussion

To evaluate the aggregation of non-overlapping constraints a set of five nesting instances were selected from the ESICUP website (<http://www.fe.up.pt/esicup>), especially suited for continuous rotations. Instance *poly1a* is the least complex one with only 15 pieces. Instances *poly2a* and *poly3a* are multiples of *poly1a* with respectively 30 and 45 pieces (i.e., 2 and 3 times the pieces of *poly1a*). Instances *poly2b* and *poly3b* have respectively 30 and 45 pieces, but only 15 are the pieces of *poly1a* and the remaining ones are new pieces. The algorithm proposed by Rocha et al. (2013) was used to obtain the circle coverings.

The computational experiments were performed on a computer with two Intel Xeon E5-5690 processors at 3.46Ghz, with 48Gb Ram at 1333Mhz, running Ubuntu 12.04 LTS x86-64, and using a single-thread. The selected Non-Linear solver is Algencan v2.37 (Andreani et al., 2007, 2008) (<http://www.ime.usp.br/egbirgin/tango/>), which is a non-linear solver based on the Augmented Lagrangian multipliers method.

The selected solver converges to a local minimum, requiring multiple starting points to explore different regions of the solution space. A total of 30 initial solutions were generated by randomly placing the pieces on a grid, in a non-overlapping configuration with random rotations.

Two model variants were tested, one with the model (1–7) where the non-overlapping constraints are considered independent, and the other where the non-overlapping constraints are aggregated in a single constraint (8), respectively denoted as *independent* and *aggregated*. Table 1 summarizes the results obtained, where the first column identifies the instance, the second one shows the number of pieces for each instance, and the third column identifies the model variant. The next two columns presents the minimum and average objective function (layout length) achieved by each model variant for the 30 initial solutions. The last column shows the average running time, in seconds, of each variant.

Table 1 Model variants results.

Instance	Number of pieces	Model variant	Obj. Function (l)		Avg. Time (s)
			Min.	Avg.	
<i>poly1a</i>	15	<i>independent</i>	16.39	17.33	24.47
		<i>aggregated</i>	16.04	17.63	9.00
<i>poly2a</i>	30	<i>independent</i>	30.30	32.25	635.70
		<i>aggregated</i>	30.94	32.84	111.10
<i>poly2b</i>	30	<i>independent</i>	33.83	35.64	709.17
		<i>aggregated</i>	33.64	35.78	121.40
<i>poly3a</i>	45	<i>independent</i>	45.70	47.33	3104.67
		<i>aggregated</i>	46.67	47.96	249.57
<i>poly3b</i>	45	<i>independent</i>	45.00	47.02	2736.03
		<i>aggregated</i>	46.37	48.03	266.47

The results show that both variants have distinct behaviours in what concerns solution quality and computational time. Aggregating non-overlapping constraints clearly allows the computational times to be much lower, allowing this variant to scale well for instances with more pieces. The average computational time of the aggregated variant is less than 10% of the variant with the original model for the larger instances. This is achieved at expenses of solution quality which in average decreases for the aggregated variant in all instances (the biggest decrease is 2.1% for *poly3b*). The minimum layout length shows a different behaviour, with the aggregated variant achieving better results for *poly1a* and *poly2b*, and worst results for the remaining instances (the biggest decrease is 3.0% for *poly3b*).

5 Conclusions

The main conclusion of this work is that aggregating non-overlapping constraints proved to be an effective and efficient technic to handle the computational complexity of larger instances. The downside of this approach is a small reduction in the layout quality. If the objective is to obtain the best layout length, the best option to use is the variant with independent non-overlapping constraints. However, if the

aim is to solve instances with a large number of pieces in a reasonable computational time, the the best choice is to use the model variant with aggregated non-overlapping constraints at expenses of a slight decrease in the solution quality.

Acknowledgements This research was funded by ERDF through the Programme COMPETE and by the Portuguese Government through FCT – Foundation for Science and Technology (PTDC/SEN-TRA/121715/2010 – StableCargo).

References

- R. Andreani, E Birgin, J.M. Martínez, and M.L. Schuverdt. On augmented lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18:1286–1309, 2007.
- R. Andreani, E Birgin, J.M. Martínez, and M.L. Schuverdt. Augmented lagrangian methods under the constant positive linear dependence constraint qualification. *Mathematical Programming*, 111:5–32, 2008.
- J.A. Bennell and J.F. Oliveira. A tutorial in irregular shape packing problems. *Journal of the Operational Research Society*, 60:93–105, 2009.
- T. Imamichi and H. Nagamochi. A multi-sphere scheme for 2d and 3d packing problems. *Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*, 4638:207–211, 2007.
- T. Imamichi, M. Yagiura, and H. Nagamochi. An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. *Discrete Optimization*, 6(4):345–361, 2009.
- Donald R. Jones. A fully general, exact algorithm for nesting irregular shapes. *Journal of Global Optimization*, pages 1–38, 2013. ISSN 0925-5001.
- P. Rocha, A.M. Gomes, R. Rodrigues, and F.M.B. Toledo. Circle covering using medial axis. *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems*, 11:402–407, 2013.
- P. Rocha, A.M. Gomes, R. Rodrigues, F.M.B. Toledo, and M. Andretta. Circle covering representation for nesting problems with continuous rotations. *accepted for publication in the 19th IFAC World Congress*, August 2014.
- Y. Stoyan, G. Scheithauer, N. Gil, and T. Romanova. Phi functions for complex 2d-objects. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 2(1):69–84, 2004.
- Y. Stoyan, M.V. Zlotnik, and A.M. Chugay. Solving an optimization packing problem of circles and non-convex polygons with rotations into a multiply connected region. *Journal of the Operational Research Society*, 63(3):379–391, 2012.
- W. Zhang and Q. Zhang. Finite-circle method for component approximation and packing design optimization. *Engineering Optimization*, 41:971–987, 2009.