Mario R. Eden, Marianthi Ierapetritou and Gavin P. Towler (Editors) Proceedings of the 13th International Symposium on Process Systems Engineering – PSE 2018
July 1-5, 2018, San Diego, California, USA

Product Decomposition in Supply Chain Planning

Braulio Brunaud[a], M. Paz Ochoa[a], Ignacio E. Grossmann[a*]

[a]*Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA*
**grossmann@cmu.edu*

Abstract

Lagrangean decomposition has been used to overcome the difficulties in optimizing large-scale supply chain planning models. Decomposing the problem by time periods has been established as a useful technique. In this paper a novel decomposition scheme by products is presented. The decomposition is based on a reformulation of knapsack constraints in the problem. The new approach also allows for simultaneous decomposition by products and time periods, enabling the generation of a large number of subproblems, with the potential benefit of using parallel computing. The case study shows that product decomposition shows a similar performance than temporal decomposition. Selecting different orders of products and aggregating the linking constraints can improve the efficiency of the algorithm.

Keywords: Supply Chain Planning, Lagrange Decomposition, Product Decomposition.

1. Introduction

Tactical planning of supply chains involves the decision of material flows and inventories throughout a network of manufacturing sites and warehouses to satisfy the demand of each customer. To ensure that best decisions are made for the mid-term, optimization models are employed. These models can be very large and hard to solve. To obtain the optimal solution in reasonable time, decomposition techniques such as Lagrangean decomposition (Guignard and Kim, 1987) can be used. In this type of decomposition, the model is broken into subproblems, and the constraints linking them are dualized, i.e. transferred to the objective function with a penalty term known as Lagrange multiplier. The objective is to find the multipliers that yield the tightest bound for the problem.

Jackson and Grossmann (2003) use temporal decomposition to solve a multi-site, multi-period planning problem. Terrazas-Moreno et al. (2011) compare spatial and temporal decomposition for the same kind of problem. They conclude that the temporal decomposition gives tighter bounds than spatial decomposition for some types of production planning problems. In temporal decomposition, the problem is decomposed by time periods, which are linked by an inventory balance constraint. There is a natural dynamic structure when decomposing by time periods. The solution of a subproblem at a given time period depends only on the inventory at the end of the previous period.

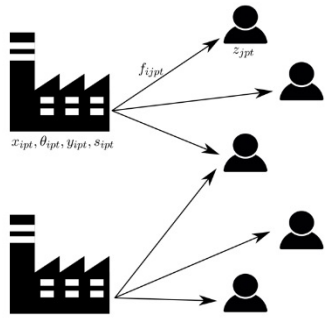
A typical supply chain planning problem has between 6 and 24 periods. At the same time the optimal decisions must be determined for a number of products that ranges typically from 50 to 1,000. However, products do not exhibit the same dynamic structure as time periods; it is not possible to say that one product comes before another. In this paper we show that such a dynamic structure can be exposed through reformulation allowing to decompose the problem by products and apply Lagrangean decomposition. Furthermore, we show that the problem can be decomposed simultaneously by products and time periods. van Elzakker et al. (2014) propose a decomposition by SKU based on heuristics,

different to the exact reformulation followed by Lagrangean decomposition from this paper.

We evaluate each decomposition scheme, and explore the effects of subproblem aggregation, linking constraint aggregation, and order of products through a multi-period multi-product production planning example.

2. Problem Description

Given is a demand forecast for a set of products p ∈ P. It is required to determine the optimal production amounts, inventory levels, and shipments to markets j ∈ J at several multi-product facilities i ∈ I. The objective is to maximize the profit for a finite horizon divided in time periods t ∈ T of length L_t. Since the production rate is different for each product, the production amount is expressed in terms of the number of hours devoted to produce a product in a given period. Fig. 1 describes the problem and the main variables used.



Variables		Parameters	
x_{ipt}	production amount	β_p	price
θ_{ipt}	production time	γ_{ip}	production cost
s_{ipt}	stock	H_i	holding cost
y_{ipt}	production executed	σ_p	setup cost
f_{ijpt}	flow	TC_{ij}	transportation cost
z_{jpt}	Sales	SUT_{ip}	setup time
		R_{ip}	production rate
		D_{jpt}	demand

Figure 1. Problem diagram and nomenclature

The problem is formulated as an MILP from Eqs. (1)-(7).

$$\textit{Max} \quad \sum_i \sum_j \sum_t \beta_p z_{jpt} - \sum_i \sum_p \sum_t \left(\gamma_{ip} x_{ipt} + H_i s_{ipt} + \sigma_p y_{ipt}\right) - \sum_i \sum_j \sum_p \sum_t TC_{ij} f_{ijpt} \tag{1}$$

$$\textit{s.t.} \quad s_{ipt} = s_{ipt-1} + x_{ipt} - \textstyle\sum_j f_{ijpt} \quad \forall ipt \tag{2}$$

$$\theta_{ipt} + SUT_{ip} y_{ipt} \leq L_t \quad \forall ipt \tag{3}$$

$$R_{ip} \theta_{ipt} = x_{ipt} \quad \forall ipt \tag{4}$$

$$\sum_i f_{ijpt} = z_{jpt} \quad \forall jpt \tag{5}$$

$$z_{jpt} \leq D_{jpt} \quad \forall jpt \tag{6}$$

$$x_{ipt}, \theta_{ipt}, s_{ipt}, f_{ijpt}, z_{jpt} \geq 0, y_{ipt} \in \{0,1\} \quad (7)$$

3. Temporal Decomposition

In temporal decomposition, the goal is to divide the problem into individual time periods, which are linked by inventory variables. In Fig. 2 the inventory balance constraint is represented by the nodes, and the inventory continuity constraint ensuring that the final inventory at a given period is equal to the initial in the next period, is represented by the box. The problem can be decomposed dualizing the inventory continuity constraint.

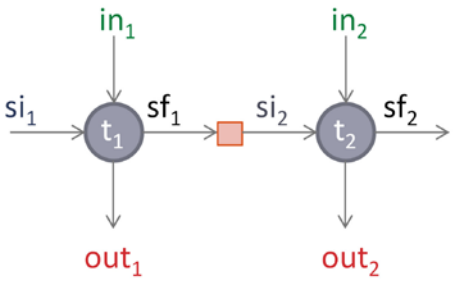


Figure 2. Representation of temporal linking constraints

Following the representation from Fig. 2 the previous model needs to be reformulated including variables to represent the initial (*si*) and final stock (*sf*). With these variables, Eq. (2) is replaced by Eqs. (8) and (9). Eq. (9) is then dualized to decompose the problem.

$$sf_{ipt} = si_{ipt} + x_{ipt} - \sum_j f_{ijpt} \quad \forall ipt \quad (8)$$

$$sf_{ipt} = si_{ipt+1} \quad \forall ipt \quad (9)$$

4. Product Decomposition

Unlike time periods, products do not have the same dynamic structure. They are usually sharing capacity in a knapsack constraint, whether is production, transportation, or inventory capacity. To reformulate the problem and obtain an equivalent dynamic structure, consider a planner that optimizes one product at a time. After the first flows and inventory for the first product have been optimized, the planner needs to know how much capacity is left for the next product. This process is shown in Fig. 3. We reformulate Eq. (3) according to the representation in Fig. 3 to obtain Eqs. (10) and (11).

$$cf_{ipt} = ci_{ipt} - \theta_{ipt} \quad \forall ipt \quad (10)$$

$$cf_{ipt} = ci_{i(p+1)t} \quad \forall ipt \quad (11)$$

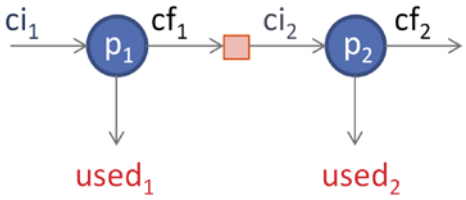


Figure 3. Product decomposition representation

The production capacity is a local variable for each product. The initial capacity for the first product is the entire capacity L_t. Note that in order to write down Eq. (11) an arbitrary order is given to the products. When Eq. (11) is dualized, the problem is decomposed into subproblems for each product.

Note that once the problem is decomposed, each subproblem is still a multi-period problem. Thus, it can also be decomposed by time periods following the reformulation from the previous section. This leads to a simultaneous decomposition by products and time periods where a subproblem can be as small as containing a single product and a single time period, opening the door to generating thousands of subproblems if desired.

5. Case Study

The computational performance of the proposed decomposition schemes is assessed through their application on the problem described in Section 2. The problem studied has 6 manufacturing sites and 10 markets. The number of products and time periods was set to 6 and 20, keeping the number of products equal to the number of time periods to ensure a fair comparison between product and temporal decomposition. Each time period represents 720 hours. The MILP formulations are modeled in JuMP and solved with Gurobi 7.5 in an Intel i7 machine with 16 Gb of RAM. Problem sizes are shown in Table 1.

Table 1. Case study problem sizes

Problem Size	Constraints	Variables	Binaries
6	2,232	4,032	216
20	24,800	44,800	2,400

5.1. Decomposition Schemes Comparison

Product decomposition (P), temporal decomposition (T), and simultaneous product and temporal decomposition (PT) were compared using two options to initialize the Lagrange multipliers: (1) zero ($\lambda = 0$), and (2) the multipliers obtained from solving the LP relaxation (λ = LP). The results for the wall clock time and optimality gap for 50 iterations are presented in Table 2.

Even though the performance of the product decomposition was superior in the examples presented, in other cases the temporal decomposition obtained a lower gap. It is not possible to conclude that one decomposition scheme is superior to the other. The simultaneous product and temporal decomposition does not result in an increased performance for small problems, because with a larger number of subproblems there is a larger number of multipliers that need to optimized.

Table 2. Comparison of decomposition schemes for 50 iterations

	λ = 0			λ = LP		
	P	T	PT	P	T	PT
Size = 6						
Time (s)	65	144	4	288	12	17
Gap (%)	6.8	6.1	19	2.7	12	13
Size = 20						
Time (s)	106	905	18	132	131	152
Gap (%)	8.8	18	62	40	76	146

5.2. Aggregation of linking constraints

Since the difficulty of the Lagrangean decomposition algorithm lies in finding the optimal set of multipliers giving the tightest bound, the algorithm can benefit from reducing the number of multipliers. One way to accomplish this while keeping the size of the subproblems, is to aggregate the linking constraints before dualization. For example, Eq. (9) linking initial and final inventories can be replaced by the surrogate constraint from Eq. (12).

$$\sum_i sf_{ipt} = \sum_i si_{ipt+1} \quad \forall pt \tag{12}$$

The results of applying this aggregation for the three decomposition schemes is presented in Table 3. The column header indicates the summation indices of the aggregated constraint. For example, (ip) indicates sum Eq. (9) over sites and products.

Table 3. Effect of linking constraint aggregation

	P				T				PT	
	none	i	t	It	none	i	p	ip	none	ip/it
Size = 6										
Time (s)	288	6	2476	6	9	48	62	27	17	17
Gap (%)	2.8	0.0	1.28	0.0	12	0.6	0.3	2.4	13	9.2
Size = 20										
Time (s)	132	1,072	114	300	131	65	156	49	151	160
Gap (%)	40	20	24	41	76	124	42	2.8	146	57

The results show that the aggregation can help in improving the performance of the algorithm. The improvements are problem dependent and require experimentation.

5.3. Order of products

As mentioned before, in order to decompose the problem by products, an arbitrary order is given to the set. This effect was explored by solving the problem of size 6 for 1,000 random orderings of the products. The gap obtained after 10 iterations is presented in the histogram in Fig. 4. With most orderings, the gap obtained is less than 10%. However, there is a significant number of orderings that yield a large gap with this algorithm

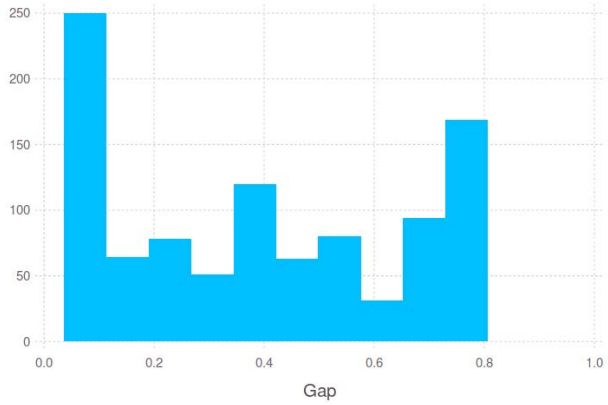


Figure 4: Effect of ordering of products

6. Conclusions

Product decomposition is a novel decomposition scheme that can potentially lead to improved results when applied to Lagrangean decomposition. Decomposing the problem by products is not consistently better or worse than decomposing by time periods. This is explained by the mathematical equivalency of both reformulations. This novel decomposition scheme is then presented as an alternative to modelers to experiment during the implementation of Lagrangean decomposition. Simultaneous product and temporal decomposition allows to generate a larger number of subproblems and take more advantage of parallel computing. The number of subproblems used is also a matter of experimentation in the implementation phase of the algorithm.

Aggregating linking constraints to reduce the number of multipliers can also lead to improved performance and quality of the bounds obtained. This offers additional options to optimize the performance of the decomposition algorithms. The order of products is another lever that can be adjusted to improve the obtained results.

In summary, product decomposition is a novel approach that gives several parameters to optimize the performance of decomposition algorithms, and obtain better results.

References

Guignard, M., & Kim, S. (1987). Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical programming*, *39*(2), 215-228.

Van Elzakker, M. A. H., Zondervan, E., Raikar, N. B., Hoogland, H., & Grossmann, I. E. (2014). An SKU decomposition algorithm for the tactical planning in the FMCG industry. *Computers & Chemical Engineering*, *62*, 80-95.

Jackson, J. R., & Grossmann, I. E. (2003). Temporal decomposition scheme for nonlinear multisite production planning and distribution models. *Industrial & engineering chemistry research*, *42*(13), 3045-3055.

Terrazas-Moreno, S., Trotter, P. A., & Grossmann, I. E. (2011). Temporal and spatial Lagrangean decompositions in multi-site, multi-period production planning problems with sequence-dependent changeovers. *Computers & Chemical Engineering*, *35*(12), 2913-2928.