



HAL
open science

Aggregate Planning for Multi-product Assembly Lines with Reconfigurable Cells

Mehmet Uzunozmanoglu, Birger Raa, Veronique Limère, Alexander De Cock,
Yogang Singh, Angel J. Lopez, Sidharta Gautama, Johannes Cottyn

► **To cite this version:**

Mehmet Uzunozmanoglu, Birger Raa, Veronique Limère, Alexander De Cock, Yogang Singh, et al..
Aggregate Planning for Multi-product Assembly Lines with Reconfigurable Cells. IFIP International
Conference on Advances in Production Management Systems (APMS), Sep 2021, Nantes, France.
pp.525-534, 10.1007/978-3-030-85902-2_56 . hal-04117694

HAL Id: hal-04117694

<https://inria.hal.science/hal-04117694v1>

Submitted on 5 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Aggregate planning for multi-product assembly lines with reconfigurable cells

Mehmet Uzunosmanoglu^{1,2}[0000-0002-5894-5221], Birger Raa¹[0000-0003-2127-4513], Veronique Limère^{2,3}[0000-0001-6858-6309], Alexander De Cock⁴[0000-0002-8641-054X], Yogang Singh^{1,2}[0000-0002-7352-8936], Angel J. Lopez^{1,2}[0000-0002-9080-4859], Sidharta Gautama^{1,2}[0000-0001-5628-6974], and Johannes Cottyn^{1,2}[0000-0002-1668-3331]

¹ Dept. of Industrial Systems Engineering and Product Design, Ghent University, Belgium

² ISyE CoreLab, Flanders Make, Belgium, www.FlandersMake.be

³ Dept. of Business Informatics and Operations Management, Ghent University, Belgium

⁴ CodesingS CoreLab, Flanders Make, Belgium, www.FlandersMake.be

Abstract. This paper deals with aggregate planning of Reconfigurable Assembly Lines (RAL). The assembly line considered in this paper consists of hexagonal cells. These have multiple slots where processing modules can be inserted to perform certain operations. In addition, each cell has a single central slot where a central module can be inserted for inter-cellular and intra-cellular transportation of parts. Multiple products with different assembly sequences must be handled over multiple planning periods. An Integer Quadratic Programming (IQP) model is proposed to solve the following problems simultaneously: (i) assigning processing modules and a central module to the cells; (ii) installation of the cells and conveyors between the cells; and (iii) routing products, ensuring that availability of the resources is not exceeded. The assembly line should be reconfigured over time to adapt to possible product functionality and demand changes at minimum reconfiguration, operational and material handling costs while ensuring the demand is met within each period. The IQP model is implemented and solved for an illustrative problem and its extensions using Gurobi.

Keywords: reconfigurable assembly lines · aggregate planning · integer quadratic programming · optimisation models

1 Introduction

Market dynamics have evolved, with increased volatility of demand and variety of product functionalities, challenging the traditional manufacturing methods. Reconfigurable Manufacturing Systems (RMS) are proposed to fill the gap between the efficiency of Dedicated Manufacturing Systems (DMS) and the flexibility of Flexible Manufacturing Systems (FMS) to overcome the challenges with minimum cost [1].

Along with RMS systems, various optimization problems have appeared in the literature. E.g., for Reconfigurable Cellular Manufacturing Systems (RCMS), part grouping, cell formation, and cell loading problems are discussed [2, 3]. For Reconfigurable Manufacturing Lines (RML), the number of workstations, the number of machines per workstation, and the operation assignments need to be decided [4]. Similarly, Reconfigurable Assembly Lines (RAL) face the same problem of determining the system configuration while ensuring the assembly sequence of parts [5]. The most common objective for these optimization problems is to minimize total costs, including e.g., investment, reconfiguration, operational, inventory, etc. Other aspects are often considered as well, such as reconfigurability, operational capability, and reliability [6]. An elaborate overview of the relevant literature is found in [7].

In this paper, aggregate planning for multi-product assembly lines with reconfigurable hexagonal cells is considered. Instead of workstations with parallel machines as in the traditional RML and RAL, cells are considered that have multiple slots for side modules. Every side module has specific assembly capabilities. Thus, different assembly operations can be performed within a single cell at the same time (if different side modules are assigned to the different slots in a cell), but also the same assembly operation can be performed in different cells at the same time (if there are multiple instances of the same side module and these are assigned to different cells). Furthermore, every cell needs a central module for moving parts and products between its side modules, but also between the cell and the conveyor connecting it to the other cells.

For the aggregate planning of these systems, 3 different problems are solved simultaneously: cell configuration, system configuration, and product routing. Cell configuration focuses on selecting side modules for the hexagonal cells to ensure that all assembly operations for all products can be performed, as well as selecting a central module for each cell. System configuration considers the decision of activating the cells and conveyors as well as positioning them, such that parts can visit the different cells along the assembly line, taking into account specific layout requirements and material handling costs. Product routing, finally, aims to find a routing for each part and product throughout the system. Note that it is possible for different instances of the same product type to follow a different routing, as long as the availability of the resources (side modules, central modules, conveyors) is not exceeded. For this integrated aggregate planning problem, a Quadratic Integer Programming model is developed and validated. The outcomes serve as an input for further detailed scheduling which is beyond the scope of this paper.

The rest of the paper is organized as follows. Section 2 presents the IQP model. An illustrative example is introduced, and multiple instances with different size are solved in Section 3. Finally, conclusions and plans for future work are given in Section 4.

2 Integer Quadratic Programming Model

This section introduces the integer quadratic programming model. Only some variables and constraints are shown for the side and central modules due to the space limitations in the paper. For the other resources, i.e., conveyors, transportation side modules, and the cells, the variables and constraints are similar to the ones shown in this section. Nevertheless, the missing variables and constraints are explained in words.

Sets

- P : Products to be assembled.
- N_p : Sequence of assembly operations for product $p \in P$.
- T : Periods in the planning horizon.
- M : Available side modules.
- $M(i)$: Side modules that can perform assembly operation $i \in N_p$.
- C : Available cell positions.
- S : Slots for side modules in a cell.

Decision variables

- $X_{k,c}^{p,i,t} \geq 0$: number of products p having operation i performed by side module k at cell c in period t .
- $Z_{s,k,c}^t \in \{0, 1\}$: whether or not side module k is assigned to slot s of cell c in period t .
- $ZCm_{k',c}^t \in \{0, 1\}$: whether or not central module k' is assigned to the central slot of cell c in period t .
- $W_{s,k,c}^t \in \{0, 1\}$: whether or not side module k is in slot s of cell c for t and $t + 1$.
- $Y_{s,k,c}^t \in \{0, 1\}$: whether or not side module k is in slot s of cell c for t , but not $t + 1$.
- $Q_{s,k,c}^t \in \{0, 1\}$: whether or not side module k is in slot s of cell c for $t + 1$, but not t .

Parameters

- $d^{p,t}$: Demand for product p in period t .
- a_k : Available processing time of side module k per period.
- $b_k^{p,i}$: Processing time for task i of product p using side module k .
- $o_k^{p,i}$: Processing cost for task i of product p using side module k .
- f_k, g_k : Insertion/removal cost of side module k .
- $o_{k'}^{p,i}$: Cost for moving product p at task i using central module k' .

IQP Model

$$\text{minimize } \sum_{t \in T} \sum_{p \in P} \sum_{i \in N_p} \sum_{k \in M(i)} \sum_{c \in C} o_k^{p,i} \cdot X_{k,c}^{p,i,t} \quad (0.1)$$

$$+ \sum_{t=1}^{|T|-1} \sum_{s \in S} \sum_{c \in C} \sum_{k \in M} (f_k \cdot (W_{s,k,c}^t + Y_{s,k,c}^t + Q_{s,k,c}^t) + g_k \cdot Y_{s,k,c}^t) \quad (0.2)$$

$$+ \sum_{t \in T} \sum_{k' \in CM} \sum_{c \in C} \left(\sum_{p \in P} \sum_{i \in N_p} \sum_{k \in M(i)} 3 \cdot o_{k'}^{p,i} \cdot X_{k,c}^{p,i,t} \right) \cdot ZCm_{k',c}^t \quad (0.3)$$

subject to

$$\sum_{k \in M} Z_{s,k,c}^t \leq 1 \quad (\forall c \in C, \forall s \in S, \forall t \in T) \quad (1)$$

$$\sum_{k \in M} Z_{3,k,c}^t + \sum_{k \in M} Z_{1,k,c+1}^t \leq 1 \quad (\forall c = 1, \dots, |C| - 1, \forall t \in T) \quad (2)$$

$$\sum_{c \in C} \sum_{s \in S} Z_{s,k,c}^t \leq 1 \quad (\forall k \in M, \forall t \in T) \quad (3)$$

$$2 \cdot Z_{s,k,c}^t + Z_{s,k,c}^{t+1} = 3 \cdot W_{s,k,c}^t + 2 \cdot Y_{s,k,c}^t + Q_{s,k,c}^t \quad (\forall s \in S, \forall k \in M, \forall c \in C, \forall t \in T \setminus \{t_{|T|}\}) \quad (4)$$

$$W_{s,k,c}^t + Y_{s,k,c}^t + Q_{s,k,c}^t \leq 1 \quad (\forall s \in S, \forall k \in M, \forall c \in C, \forall t \in T \setminus \{t_{|T|}\}) \quad (5)$$

$$\sum_{p \in P} \sum_{\{i \in N_p | k \in M(i)\}} X_{k,c}^{p,i,t} \cdot b_k^{p,i} \leq a_k \cdot \sum_{s \in S} Z_{s,k,c}^t \quad (\forall k \in M, \forall c \in C, \forall t \in T) \quad (6)$$

$$\sum_{k \in M(i)} \sum_{c \in C} X_{k,c}^{p,i,t} = d_p^t \quad (\forall p \in P, \forall i \in N_p, \forall t \in T) \quad (7)$$

$$\sum_{c=1}^{c_1} \sum_{k \in M(i)} X_{k,c}^{p,i,t} \geq \sum_{c=1}^{c_1} \sum_{k \in M(i+1)} X_{k,c}^{p,i+1,t} \quad (\forall p \in P, \forall i < |N_p|, \forall c_1 \in C, \forall t \in T) \quad (8)$$

The objective function includes 4 different cost components. (0.1) is the operational cost of the side modules, whereas (0.2) consists of the configuration

and reconfiguration costs of side modules. Note that using the same notation, configuration and reconfiguration costs of central modules, conveyors, and transportation side modules are easily calculated. As the third component, the operational cost of central modules is considered in (0.3), taking into account that every assembly operation results in three movements. Note that this term is quadratic. The fourth cost component, not shown in the model, is the total cost of using the conveyors.

In constraints (1), it is ensured that a slot can be utilized by at most one side module. Similar constraints can be written for the central modules by using appropriate indices. Also, an additional constraint ensures that the 5th slot of a cell is allocated to either a side module or a transportation side module depending on the decision made for the insertion of additional conveyors. Note that it is already assumed to have an operational conveyor between two adjacent cell positions for all the cell position pairs in the problem, as the part source and part exit need to be connected regardless of the number of open cells. But, their capacity might be insufficient to provide inter-cellular transportation. Then, a decision is made for each adjacent cell pair to insert an additional conveyor.

Constraints (2) are the problem-specific overlapping constraints. Some side module slots in adjacent cells are overlapping meaning that side modules cannot be inserted into these slots at the same time. E.g., slots I7 and I8 are overlapping (see Figure 1).

Constraints (3) ensure each side module instance can be inserted into at most one slot during a period. Again, similar constraints can be written for the central modules just by changing the variable.

Constraints (4) identify which of the three reconfiguration scenarios involving insertion and removal of side modules applies. These constraints can also be written for all the other resources by just changing the variables appropriately. Related to these constraints, constraints (5) select one of these scenarios for each resource. These constraints are again duplicated for all the resources. These constraints help define the resource reconfiguration costs included in the objective function (0.2).

Constraints (6) ensures that the availability of the side modules is not exceeded if the side modules are inserted into the slots.

Constraints (7) impose that the demand for any product must be met in any period by performing all of its assembly operations. Since it is allowed to perform the same assembly operation in different cells and by using different side modules, the sum of the number of times an assembly operation is performed by any of these side modules must be equal to the period's demand. Along with this, constraints (8) ensure that parts can be assembled in different cells but only as long as they follow a one-directional flow, i.e., if assembly operation i of product p is performed at a certain cell, then operation $i + 1$ of that same product p can only be performed within the same cell or successor cells.

In addition, the following constraints are included in the model, but not shown here due to space limitations: constraints to ensure that an assembly operation of a product can only be performed in a cell that has a central module

that can handle the parts or sub-assemblies involved in that assembly operation; constraints to ensure that the total number of movements a central module performs does not exceed its availability; constraints to ensure that the additional conveyor between cells 2 and 3 can only be inserted if the additional conveyor between cells 1 and 2 is already inserted; and constraints to make sure that if an additional conveyor between two cells is inserted, an additional transportation side module must be inserted to those cells as well. Finally, two sets of constraints are added for ensuring that the capacity of conveyors is not exceeded and if necessary, additional conveyors are inserted.

3 Illustrative Problem and Results

In this section, an illustrative problem is introduced and the results are discussed.

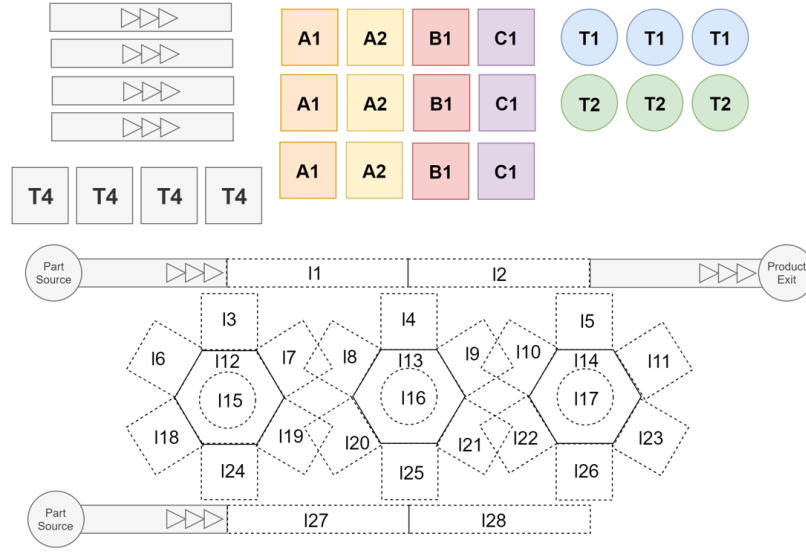
Problem setup The example consists of a layout with 3 possible cell locations in each of which the side and central module slots are numbered as seen in Figure 1. Also, the possible conveyor positions between the cells as well as 4 conveyor instances are shown. In addition, 5 types of side modules including a single type of transportation side module T4, and 2 types of central module instances are shown. The side modules perform different assembly operations, while the transportation side module T4 is used along with the central modules for moving the parts between the cells and conveyors. Note that there are some overlapping side module slots on adjacent cells which means that the overlapping side modules cannot be used at the same time, e.g., I7 on cell 1 and I8 on cell 2.

In Table 1, the resource specifications are given. ID is the name of the resource type where the letter shows the assembly operation name and the number is the variant ID, e.g., A1 and A2 are two different resource types that perform the same assembly operation A. Instances is the number of available resources of that type. Note that in the example, all of the assembly operations are performed by assembling two parts. Skill time is calculated as minutes, and skill cost is incurred each time the resource is used. Insertion and removal costs are assumed to be symmetrical, and they are used to calculate the reconfiguration cost.

As can be seen in Table 2, there are 3 different assembly operations A, B, and C that are necessary to assemble 4 different products that each have a different sequence of assembly operations to be performed. Furthermore, the demand for the products in 2 periods is given.

Note that at the beginning of the first period, it is assumed that there is no prior configuration for both the cells and the system (referred to as ‘greenfield’). At the beginning of the second period, however, it is possible that a reconfiguration occurs depending on the configuration used during the first period (referred to as ‘brownfield’).

Results The IQP model is implemented and solved with Gurobi for Python version 3.7. The program ran on a PC with an Intel(R) Core i7-10610U CPU @ 1.80GHz and 16 GB RAM capacity.


Fig. 1. Assembly system layout and available resources.

ID	Instances	Skill Time	Skill Cost	Migration Cost	
				Insert	Remove
Cell	3	-	-	400	400
A1	3	02:00	10	100	100
A2	3	04:00	20	50	50
B1	3	03:00	10	100	100
C1	3	*	*	100	100
T1	2	00:24	10	150	150
T2	2	01:00	30	150	150
T4	6	-	-	100	100
T5	4	02:00	10	100	100

Table 1. Resource specifications.

ID	Parts	N_p	Demand
P1	5,5,4	{C, A}	{2, 8}
P2	1,2,3,4	{A, B, A}	{2, 3}
P3	1,2,3,5,5,4	{A, B, C, C, A}	{6, 2}
P4	2,6	{C}	{6, 4}

Table 2. Products, parts, assembly sequence, and demand.

For the instance with 4 products, 3 possible cell locations, and 2 time periods, it is found that the configuration shown in the Figure 2 is the only configuration that is used during both periods. Therefore, no reconfiguration is necessary.

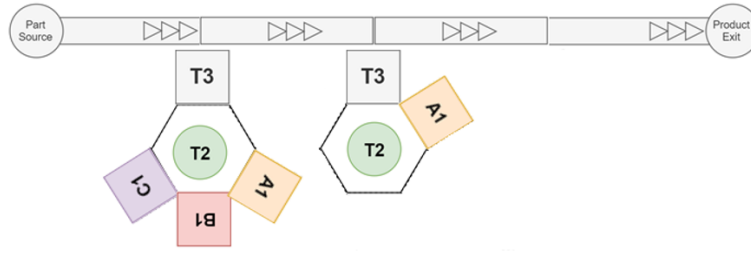


Fig. 2. Resulting configuration

To test the model with different instances, the illustrative problem is extended by increasing the number of periods and products. First, the number of periods is increased by generating new demand for each product randomly. However, this is performed in a way that the resource availability is sufficient to meet the product demands as the resource pool is kept the same during the experiments. For this reason, the random demands are drawn from $U(0, 10)$ where U is a discrete uniform distribution.

Secondly, the number of products is increased by generating random demands as described previously, and by generating random-length assembly operation sequences. To generate these sequences, first, the length is randomly defined by drawing a number from $U(1, 5)$ which ensures consistency with the products considered in the illustrative problem. Then, the sequences are filled randomly with the three assembly operations A, B, and C.

All the instances shown in Table 3 are generated and solved while the random seed is fixed. The first nine instances are generated by increasing the number of periods, and keeping the number of products fixed. Next eight instances are generated by increasing the number of products, and keeping the number of periods fixed.

As it is seen in the first nine rows in Table 3, the number of periods affects the solution time dramatically. Increasing it from 2 to 4 increases the solution time by a factor of 13, whereas the increase from 4 to 8 increases the solution time by a factor of 22. In addition, depending on the definition of the periods, e.g., hours, days, months, 10 periods can be considered as a short time horizon for the planning problem considered in this paper. Therefore, it is easy to see that for the problem instances where the periods represent short time intervals, and where it is necessary to generate a plan for a long time horizon, the problem can not be solved in an acceptable time with the current solution approach.

Increasing the number of products affects the solution time dramatically as well. Along with the fact that the model size increases with the number of products, it is observed that making such an aggregate plan including reconfiguration is harder with resources fixed while the total demand is increasing. Therefore, this is reflected in the solution time as well. Likewise, the problem becomes in-

ID	P	T	CPU Time (s)
1	4	2	0.94
2	4	3	3.53
3	4	4	12.32
4	4	5	16.29
5	4	6	81.35
6	4	7	183.27
7	4	8	270.56
8	4	9	454.16
9	4	10	556.60
10	5	2	1.71
11	6	2	4.08
12	7	2	7.18
13	8	2	45.23
14	9	2	76.31
15	10	2	124.14
16	11	2	258.36
17	12	2	-

Table 3. Results

feasible when there are more products than the resources can process, as can be seen in row 17.

4 Conclusions and Future Work

In this paper, aggregate planning of a RAL with hexagonal reconfigurable cells is considered. The aggregate planning is performed by solving three problems simultaneously: cell configuration, system configuration, and product routing. The demand for multiple products across multiple periods is met with minimum operational and reconfiguration costs while considering the limited availability of all the resources in the RAL. An IQP model is proposed and solved with Gurobi for an illustrative example. Small problem instances are solved to optimality in limited time using Gurobi. However, for larger instances, this is no longer the case. Therefore, (meta)heuristic approaches will be developed as the next step.

Along with the development of other solution methods, the model will be generalized further. This involves considering automatic guided vehicles (AGVs) or other material handling systems for inter-cellular transportation, and extending the number of cell positions to more than three. In addition, as opposed to a fixed assembly operation sequence for each product type considered in this paper, a flexible sequence of operations (with precedence constraints) is often possible in practice. Therefore, it is planned for the future to adopt this flexibility by allowing alternative sequences in the model. Finally, more extensive computational experiments will be conducted.

Acknowledgement

This research was funded by the Flanders Make SBO project AssemblyRecon.

References

1. Koren, Y., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel.: “Reconfigurable Manufacturing Systems.” In *Manufacturing Technologies for Machines of the Future*, 627–665. Springer, Berlin (1999)
2. Yu, J.-M., H.-H. Doh, H.-W. Kim, J.-S. Kim, D.-H. Lee, and S.-H. Nam.: “Iterative Algorithms for Part Grouping and Loading in Cellular Reconfigurable Manufacturing Systems.” *Journal of the Operational Research Society* 63 (12): 1635–1644 (2012)
3. Eguia, I., J. C. Molina, S. Lozano, and J. Racero.: “Cell Design and Multi-period Machine Loading in Cellular Reconfigurable Manufacturing Systems with Alternative Routing.” *International Journal of Production Research* 55 (10): 2775–2790 (2016)
4. Dou, J., X. Dai, and Z. Meng.: “Optimisation for Multi-part Flow-line Configuration of Reconfigurable Manufacturing System Using GA.” *International Journal of Production Research* 48 (14): 4071–4100 (2009)
5. Bryan, A., S. Jack Hu, and Y. Koren.: “Assembly System Reconfiguration Planning.” *Journal of Manufacturing Science and Engineering* 135 (4): 041005 (13 pages) (2013). <https://doi.org/10.1115/1.4024288>
6. Ashraf, M., and F. Hasan.: “Configuration Selection for a Reconfigurable Manufacturing Flow Line Involving Part Production with Operation Constraints.” *The International Journal of Advanced Manufacturing Technology* 98 (5–8): 2137–2156 (2018)
7. Yelles-Chaouche, A., Gurevsky, E., Brahimi, N., & Dolgui, A.: “Reconfigurable Manufacturing Systems From an Optimisation Perspective: A Focused Review of Literature.” *International Journal Of Production Research*, (1-19) (2020). <https://doi.org/10.1080/00207543.2020.1813913>