



# A Decision Support Methodology for the Design of Reconfigurable Assembly Systems

M. Colledani, A. Yemane, Giovanni Lugaresi, N. Frigerio, G. Borzi, A. Bassi,  
D. Callegaro

## ► To cite this version:

M. Colledani, A. Yemane, Giovanni Lugaresi, N. Frigerio, G. Borzi, et al.. A Decision Support Methodology for the Design of Reconfigurable Assembly Systems. IFAC-PapersOnLine, 2018, 51 (11), pp.108-115. 10.1016/j.ifacol.2018.08.243 . hal-03880618

**HAL Id: hal-03880618**

**<https://hal.science/hal-03880618>**

Submitted on 1 Dec 2022

**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.

# A Decision Support Methodology for the Design of Reconfigurable Assembly Systems

M. Colledani, A. Yemane<sup>1</sup>, G. Lugaresi, N. Frigerio<sup>\*</sup>  
G. Borzi, A. Bassi, D. Callegaro<sup>\*\*</sup>

<sup>\*</sup> *Dipartimento di Meccanica, Politecnico di Milano  
P.za Leonardo da Vinci 32, 20133 Milan, Italy*

<sup>1</sup> *Corresponding author: antenehteferi.yemane@polimi.it*

<sup>\*\*</sup> *Enginsoft S.p.a.*

*Via Stezzano, 87, 24126 Bergamo, Italy*

**Abstract:** The design phase of assembly systems under uncertain product-mix and product volume settings is challenging for both manufacturers and researchers. One of the design paradigms to deal with uncertain and dynamic environments is through reconfigurable system designs. This paper presents the formulation and a solution approach for supporting the design of assembly systems that are based on reconfigurable and modular resources. The method assists manufacturers in the search of optimal design solutions and future reconfiguration actions. Systems designed this way can efficiently respond to external changes, such as evolving product types and uncertain demand scenarios. The proposed methodology has been used to solve a real use-case in the ReCaM project, and preliminary results demonstrate its potential benefits for industrial settings.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

**Keywords:** Reconfigurable Manufacturing Systems, Modular Production, ReCaM Project, Scenario optimization.

## 1. INTRODUCTION

In response to the current market requirements, such as dynamic production volumes and high variety of products, the rapid adjustment of production capacity and processing functionality of manufacturing systems is an important research topic. Central to this research is the concept of Reconfigurable Manufacturing Systems (RMS), well defined in [Mehrabi et al., 2000] and [Koren et al., 1999]. An RMS is a system designed for rapid change both in its structure and in its hardware and software components, in order to quickly and efficiently adjust its capacity and functionality within a part family in response to sudden market changes. General requirements for the development of the next generation of manufacturing systems are presented in [Bi et al., 2008].

Due to the unpredictable evolution of market requirements, i.e., future demand and product-mix, decisions on how to design the system for today's requirement also need to take into account the future changes that might be done on the system. Thus, when faced with flexibility requirements, namely the ability to face a spectrum of market scenarios, future system scalability becomes essen-

tial [Son et al., 2001], together with the ability to switch from one type of system architecture to another, e.g., from batch to dedicated production [Gamberi et al., 2008]. These challenges push manufacturers to take smarter decisions at greenfield stage, considering subsequent system alterations. The problem they are facing has the following characteristics: (1) it is multi-period, as the initial investment regards a system capable of producing for several years; this also implies that the problem will be subject to uncertainty (2) in general, the problem is multi-product; (3) decisions may still be taken over-time; (4) the system is modular.

In order to solve such a problem featuring many uncertain elements, stochastic approaches play a central role in system design [Tolio, 2008]. Among others, some of the main problems concerning system design of large-scale production systems include Assembly Line Balancing [Becker and Scholl, 2006], Buffer Allocation [Demir et al., 2014], Capacity estimation [Wazed et al., 2010]. Stochastic Programming [Birge and Louveau., 1997] is based on the assumption that the fluctuation of model parameters is governed by their probability distribution. Alternative methodologies to deal with uncertainty include robust approaches, such as the ones proposed in [Ben-Tal and Nemirovski, 2000] and reviewed in [Bertsimas et al., 2011]. Despite their potential, these approaches may lead to over designed solutions, with wasted capacity if the ranges of variation of parameters are not set according to their real behavior. Other works related to capacity reconfiguration management in RMSs dealing with uncertain

<sup>\*</sup> The research leading to these results has received funding from the European Union Horizon 2020 Program (FoF-11-2015) under grant agreement n 680759 (ReCaM: "Rapid reconfiguration of Flexible Production Systems through Capability-based Adaptation, Auto-configuration and Integrated tools for Production Planning"). The authors would like to thank the Robert Bosch GmbH for the support in this research.

demand scenarios have been proposed in [Renna, 2010], where the authors use simulation to assess the system performance under different scenario realizations. Capacity Management under uncertainty has been presented in [Asl and Ulsoy, 2003], where optimal policies are found using Markov Decision Processes.

However, in the literature, methods applicable to real industrial problems characterized by complexity, multi-objective KPIs and multi-period decision making are not widely available. In this paper, a new approach for the problem formalization and solution methodology of the greenfield design of a reconfigurable system capable of assembling multiple products with different features by considering the multi-period reconfiguration problem that arises from changing product volumes and product-mix is proposed. Therefore, the aim of this work is to provide a method for the formalization, modeling and a solution approach for the analysis of the multi-period and scenario-based design of a modular system. These activities are part of the ReCaM project framework [<http://recam-project.eu/>, 2015-2018], thus, the proposed method is implemented into a software tool and validated for the analysis of a real system design scenario.

The paper is organized as follows: section 2 illustrates the problem statement, section 3 describes the modeling approach, section 4 deals with the solution approach that has been used in this case, while section 5 discusses a real-case application coming from the ReCaM project. Final remarks are given in section 6.

## 2. PROBLEM STATEMENT

The problem introduced in section 1 considers a finite time horizon which is divided into a set of periods, as shown in Fig. 1. Every period includes a set of scenarios, represented by *circles*. Each scenario is identified by a product-mix and production volumes corresponding to each product type, which are assumed to be the result of a demand estimation process. The transition from one scenario to the next – represented by *arrows* between two consecutive periods – is uncertain and this information is represented by transition probabilities. This scenario tree representation helps to map alternative forecast data and the associated likelihood of their occurrence along the time horizon. Given the scenario tree, the goal is to find reconfigurable design solutions that minimize the expected cost of: purchasing resources, reconfiguring the system across multiple periods, and operating over the entire time horizon while guaranteeing the demand constraints in each scenarios.

The solution method proposed in this paper approaches the entire problem as two sub-problems. The first sub-problem refers to finding design solutions that are capable of satisfying the requirements defined within each scenario, which we will refer to as the “single-period” sub-problem. Each single-period problem can be considered as a greenfield design problem, which aims at answering the question: “Which is the feasible set of resources to be installed in the system, in order to satisfy the demand and the product-mix requirements of the specific scenario and logistics system performance measures?”. As a consequence, to each of the scenarios corresponds

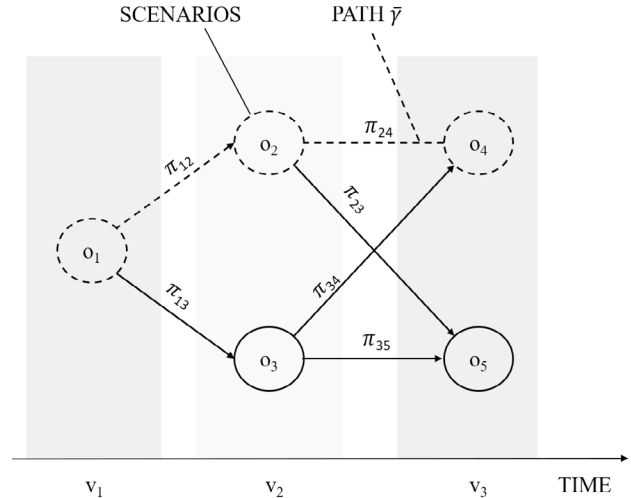


Fig. 1. Scenario Tree in the case with three time periods and five scenarios.

a set of optimal solutions, namely configurations of the production system that can overcome the requests of the market at optimal cost. Therefore, multiple single-period problems can be solved independently from one another. However, a solution that has been proven optimal in the single-scenario problem may not be so in the multi-period problem. In fact, along the transitions between periods, additional decisions related to the reconfiguration of the system have to be taken, i.e., some resources may need to be purchased, while others might be just moved within the system, in order to adapt it to the requirements of the new scenario. These reconfiguration actions are dependent on the specific transitions between scenarios, which occur at the end of one period and the start of the next one, depending on which scenarios are expected to occur. Indeed, the selection of different designs for scenarios creates different reconfiguration requirements. Thus, the next stage of the sub-problem referred to as the “multi-period” problem is added. The goal is to find the optimal system configurations taking into account all the possible paths along scenarios that the system might have to face. Since the entire problem has a long-term horizon, it is assumed that decisions are not taken during time periods. The objective is to minimize the expected cost over the entire time horizon while satisfying the demand constraints in each scenarios.

### 2.1 The ReCaM Approach

The methodology is part of the ReCaM project. Specifically, it is part of a software platform that supports engineers in the green field design phase of assembly systems. The platform is composed of interconnected software building blocks that exchange information with both product and resource catalogues and the Manufacturing Execution Systems (MES). In the ReCaM approach, the system designs are based on modular and programmable resources called Mechatronic Objects (MOs), which provide the processing capabilities required by the products. Different types of MOs can be selected from a catalogue to execute tasks which vary with respect to the complexity of operations and the degree of automation. MOs are combined in different ways each providing the capabilities

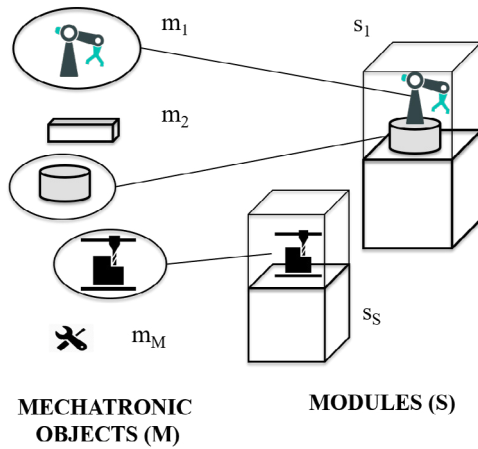


Fig. 2. Correspondence between MOs and Modules as part of a production system

required to perform a certain task. Thus, by combining MOs it is possible to obtain *modules* with a certain set of capabilities or combined capabilities. A *module* is a combination of MOs that can perform more than one task (see Fig. 2, e.g., a gripper has grasping capability, and together with an anthropomorphic arm it constitutes a module with both grasping and moving capabilities). Thus, the decision on the optimal selection of which modules compose the reconfigurable assembly system will be supported by the two sub-problems introduced in this section.

## 2.2 Single-Scenario Problem

This sub-problem solves a single scenario within a single period taken from the entire time horizon considered in the design problem. The product-mix and the target production volumes of each product type associated to the specific scenario are among the constraints. The product structure is provided in terms of subcomponents (Bill of Materials - BOM) and the assembly tasks that need to be performed, together with specific precedence constraints, are also provided (Bill of Operations - BOO). For each task, a certain set of *modules* are needed. This task-to-module assignment problem is commonly known as a *Multi-Product Line Balancing Problem* (MALP). The desired system needs to satisfy a target throughput. Three issues need to be considered: the setup time, the resource reliability, and the buffer capacity. Setup times required to switch between product variants are considered. Resources or MOs are assumed to be prone to failures. Therefore, failure and repair parameters, namely the Mean Time to Failure (MTTF) and Mean Time to Repair (MTTR) are considered in the model. Inter-operational buffers with finite capacity are considered in the model and *Buffer Allocation Problem* (BAP) is addressed. Thus, for the single period, the assembly system design problem consists in: (1) MALP, that is assigning tasks to stations subject to precedence constraints, machine reliability information, and MOs capability matching; (2) BAP, that is selecting the capacity of inter-operational buffer storage. Moreover, more than one feasible solution may be found due to conflicting objectives, such as lead time reduction and work-in-progress minimization.

## 2.3 Multi-Period Multi-Scenario Problem

In the multi-period multi-scenario problem the possible future paths across time horizon, which feature several scenarios are considered. The information about the production requirements of each period becomes available over time. Indeed, at each time period, the scenarios' realizations (i.e., product mix and product quantities) are observed and decisions are taken accordingly. Each decision is taken after the observation and it depends on the period and on the observed scenario. However, the problem is solved at time "zero" with respect to the expected probability of having a certain scenarios in the future.

As shown in Fig. 1, in each period more than one scenario may occur, each characterized by its product-mix, and the corresponding volumes, etc. Each scenario is uniquely defined by an ID and it is known a-priori to which time period it belongs. The *transition probability* between scenarios are also inputs to this sub-problem.

The goal is to obtain the lowest total cost while respecting the target production level of the given product mix in each period. Therefore, the objective function consists in the minimization of the expected total system cost over the time horizon. The total cost includes:

- Investment cost (€) to buy resources (MOs and buffer capacity);
- Variable operational cost (€/h) to utilize a certain resource such as operators, energy, material, cost per failure, etc. In more details, the *variable operational cost* includes: the cost of having a working resource, the cost of having an idle resource, and the cost of having a resource in failure. It also considers an additional cost per failure event;
- Inventory cost (€/part) of having a certain number of work-in-process (WIP) parts held in buffers;
- Reconfiguration cost (€) to reconfigure a system layout to a new one for the next period to accommodate new production requirements. It involves: (1) The investment cost of buying a new instance of a certain resource; (2) The installation cost of a resource in the production line (e.g. connection to a HUB, installation on a workbench, installation of a new workbench); (3) The uninstallation cost of a resource from the production line; (4) The storage cost of resources in a resource-warehouse.

## 3. MODELING APPROACH

The modeling of the problem is based on the two sub-problems and the procedures introduced in Section 2. Thus, it starts by dividing the strategic planning time horizon into a discrete and finite number of periods  $V$ , each with deterministic and finite length. To each period is assigned a set of scenarios; for example, we may expect two scenarios representing the demand of a certain product in the following year, one with very high demand and another with medium-high demand. In this section, we further detail system reconfiguration assumptions, the model parameters and adopted notations.

Table 1. Notation

Sets	
$v \in V$ Periods	Set of discrete and finite time windows (i.e. periods) composing the entire long term planning horizon;
$o \in O$ Scenarios	Set of production forecast <i>scenarios</i> ;
$p \in P$ Products	Set of <i>product</i> variants to be produced over the entire planning horizon under consideration;
$i \in I$ Tasks	Set of <i>tasks</i> to be executed (among all product $p \in P$ );
$m \in M$ MOs	Set of <i>Mechatronic Objects</i> that can supply a certain set of capabilities;
$s \in S$ Stations	Set of <i>stations</i> that are instantiated, i.e., a processing station in which at least one MO is assigned;
$k \in K$	Position index of buffers along the production line
$b \in B$	Buffer types, which stands for technological choice of buffer system to be used between consecutive stations
Parameters	
$\Delta v$	Period length [h]
$\pi_{o_i, o_j}$	Transition probability from scenario $o_i$ to scenario $o_j$
Cost Factors	
<u>Investment Cost</u>	
$CMinv_m$	Unitary Investment Cost of module $m$ (€).
$CBinv_b$	Unitary Investment Cost of buffer $b$ (€).
Variable operational cost (energy, material, operators, idle times, etc.)	
$CMbusy_m$	Variable Cost of MO type $m$ while working (€/h).
$CMidle_m$	Variable Cost of MO type $m$ while starving and blocking (€/h).
$CMfail_m$	Variable Cost of MO type $m$ while in failure (€/h). It is assumed $CMfail_m = CMidle_m$ .
$CF_m$	Cost per failure event on MO type $m$ (€).
<u>Inventory Cost</u>	
$CBhold_{p,k}$	Inventory Cost of keeping one part of type $p$ in buffer $k$ (€/ (part * h))
$CMstock_m$	Inventory Cost of stocking a MO type $m$ . We assume $CMstock_m = CMidle_m$
<u>Installation Cost</u>	
$CMinstall_m$	Installation Cost of MO type $m$ at a certain station.
$CMuninstall_m$	Uninstallation Cost of MO type $m$ at a certain station.
$CBinstall_b$	Installation Cost of one unit of buffer-type $b$ at a certain buffer.

### 3.1 Assumptions and Notation

It is assumed that the set of *scenarios*, represented by *circles* in Fig. 1, are finite and describe possible forecasts of product-mix and quantities. However, the actual scenario outcomes are available at the beginning of each period and the decision can be taken according to this information. The initial solutions for period 1, need to guarantee the requirements defined for the first scenario, and this further assumes that there is no pre-existing system. Depending on the actual outcome of scenarios, in following periods different *reconfiguration actions* are considered. All reconfiguration actions happen only between two consecutive periods, and include: (1) the upgrade of a station in terms of installation/uninstallation of resources, (2) the purchase of a new resource, (3) the storing of a resource in a warehouse because it is not required for production. It is assumed that once an instance of a certain resource (MO, buffer) is uninstalled from the layout, it is moved to a warehouse to be stored. The reconfiguration costs are considered as the costs of reconfiguring the system between two scenarios in sequential periods, i.e., reconfiguring from system at time  $v$  to system at time  $v+1$ . Unused MOs are stored in a warehouse. The effective available production time is estimated considering reconfiguration times and setup times between products.

In the following sub sections, the list of important notations, decision variables, constraints and objective functions are presented (the full exposition of the mathematical expressions related to each list is omitted due to space limitations). Table 1 summarizes the list of notation that has been used.

### 3.2 Decision Variables

We have used the following decision variables:

- Tasks-stations assignments in scenario  $o$ :

$$x_{i,p,s,o} = \begin{cases} 1 & \text{if task } i \text{ for producing product } p \\ & \text{is assigned to station } s \text{ in scenario } o; \\ 0 & \text{otherwise.} \end{cases}$$

- MOs-stations assignments in scenario  $o$ :

$$\xi_{m,s,o} = \begin{cases} 1 & \text{if MO type } m \text{ is assigned to station } s \\ & \text{in scenario } o \\ 0 & \text{otherwise.} \end{cases}$$

- $Nline_{m,o}$  is the number of instances of MO type  $m$  that has been assigned in the layout in scenario  $o$ :

$$Nline_{m,o} = \sum_{s \in S} \xi_{m,s,o}$$

- $\beta_{b,k,o}$  is the capacity of  $k$ -th buffer of buffer type  $b$  in scenario  $o$ .

- We indicate with  $n_{b,k,o}$  the number of buffer slots of type  $b$  installed in the  $k$ -th buffer in scenario  $o$ .

### 3.3 Non-linear Quantities of the Model

- The throughput of the assembly system in scenario  $o$ :  
 $TH^o = \mathcal{F}(x_{i,p,s,o}, \xi_{m,s,o}, n_k, MTBF_m, MTTR_m, \dots)$
- The utilization of each MO associated to a station in scenario  $o$ . We indicate by  $\bar{u}_{s,m,o}$  the average utilization time ( $h$ ).  
 $u_{s,m,o} = \mathcal{F}(x_{i,p,s,o}, \xi_{m,s,o}, n_k, MTBF_m, MTTR_m, \dots)$
- The probability of being in failure of each MO associated to a station in scenario  $o$ . We indicate by  $\bar{f}_{s,m,o}$  the average failure time ( $h$ ).  
 $f_{s,m,o} = \mathcal{F}(x_{i,p,s,o}, \xi_{m,s,o}, n_k, MTBF_m, MTTR_m, \dots)$
- In scenario  $o$ , the average inventory of product  $p$  in the buffer  $k$  is  $WIP_{p,k,o}$ , and the average inventory in the buffer  $k$  is  $WIP_{k,o}$ :

$$WIP_{p,k,o} = \mathcal{F}(x_{i,p,m,s,o}, n_{k,p}, \dots) \quad (1)$$

$$WIP_{k,o}^{tot} = \sum_{p \in P} WIP_{p,k,o} \quad (2)$$

- $\theta_o$  indicates the number of solutions available for scenario  $o$ . Then, it is valid:  $\sum_o \theta_o = |\Sigma|$ .
- $NF_{m,s}$  is the average number of failures on MO type  $m$  when installed on station  $s$ .

### 3.4 Constraints

The following constraints are valid for both single and multi period cases (here we omit their mathematical formulation).

- Demand satisfaction constraints: product demands have to be met for each of the respective time periods.
- Each station performs at least one task.
- Each task can be assigned to only one station.
- Each task has to be processed by a station with the proper features.
- Precedence constraints have to be satisfied: i.e. tasks have to be assigned in compliance with the products Bill of Operations (BOO).
- Predefined machine type constraints: some tasks may have to be performed on particular stations due to external and/or user-defined constraints.
- Tasks on the same station constraints: some tasks may have to be performed together on the same station due to technological, logistical, or user-defined constrained.
- Production completeness: for each product all tasks necessary for its assembly have to be assigned.
- Line Balancing constraint (cycle time constraints): the tasks executed on a station cannot exceed the input cycle time imposed by the specific technological process.
- Total buffer capacity may have a maximum
- Each buffer  $k$  may have a maximum capacity
- For each station one buffer is required.
- Each buffer is assigned to one buffer type, i.e. buffer technology to be used (transporter, statical storage, stacking system, ...)
- For each buffer, the storing of all products is feasible with the assigned buffer type.

### 3.5 Objective Function

The objective function consists in the minimization of the expected total system cost over the time horizon. The total cost includes investment (fixed) costs, operational costs, inventory costs, and reconfiguration costs.

*Single-Scenario.* In the following, we will use indicator function  $\mathbb{I}_{v,o} = 1$  if scenario  $o$  belongs to period  $v$  and  $\mathbb{I}_{v,o} = 0$  otherwise. The single-scenario objective function is as follows:

$$Z_o = \sum_{s \in S} \sum_{m \in M} \mathbb{I}_{v,o} \Delta v \xi_{m,s,o} [CM_{busy_m} \cdot \bar{u}_{s,m,o} \quad (3)$$

$$+ CM_{fail_m} \cdot \bar{f}_{s,m,o} + CM_{idle_m} \cdot (1 - \bar{u}_{s,m,o} - \bar{f}_{s,m,o})] \quad (4)$$

$$+ \left[ \sum_{p \in P} \sum_{k \in K} CB_{hold_{p,k}} \cdot WIP_{p,k,o} \right] \quad (5)$$

$$+ \sum_{m \in M} (CMinv_m + CMinstall_m) \cdot (Nline_{m,o})$$

$$+ \sum_{k \in K} \sum_{b \in B} (CB_{inv_b} + CB_{install_b}) \cdot n_{b,k,o} \quad (6)$$

$$+ \sum_{m \in M} \sum_{s \in S} CF_m \cdot NF_{m,s} \quad (7)$$

where (4) is the variable cost of having MO type  $m$  working/idle at station  $s$  in the line, (5) is the inventory cost for all products in buffer  $k$  of a certain type  $b$ , (6) is the investment cost of buying and install all MO types and buffer capacities to create the layout, and (7) is the cost of all failure events that happen among stations.

*Reconfiguration between scenarios.* When faced with a multi-period problem, each scenario can still be solved independently as a single-period problem. Thus, a certain number of feasible solutions are available for each scenario. These solutions are available independently from the way they have been obtained. In the following, we denote by  $\sigma_{r,o} \in \Sigma$  the  $r^{th}$  solution found for the single-period, single-scenario problem for scenario  $o \in O$ . We indicate with  $\Psi$  the set of all the permutations of feasible solutions. Clearly,  $|\Psi| = \prod_{o \in O} \theta_o$ . A particular permutation  $\psi_i \in \Psi$  could be, for example in a 5-scenario case such as the one in Fig. 1,  $\psi_i = \{\sigma_{1,o_1}, \sigma_{1,o_2}, \sigma_{1,o_3}, \sigma_{1,o_4}, \sigma_{1,o_5}\}$ , that is the set of all the first solutions for each scenario. In the following part we will assume we are referring to one particular permutation  $\psi_i$  of the solutions.

Starting from an initial scenario, the reconfiguration costs can be obtained following a certain path. Following we will refer to a path  $\gamma \in \Gamma(\psi_i)$  as a set of solutions that represents a feasible sequence over time periods, where  $\Gamma(\psi_i)$  is the set of feasible paths related to a particular,  $i$ -th permutation  $\psi_i$ . For example, take the particular path  $\bar{\gamma} = \{\sigma_{1,o_1}, \sigma_{1,o_2}, \sigma_{1,o_4}\}$ . We assume all the feasible paths are known a priori.

Suppose to go from scenario  $o_1$  to  $o_2$  from period  $v$  to  $v + 1$ . A certain number of reconfiguration actions are

taken to pass from solution  $\sigma_{r,o_1}$  to  $\sigma_{r,o_2}$  (here index  $r$  is not significant). The following actions may be taken:

- Buy a new instance of module  $m$  and install it in-line at a certain station  $s$ ;
- Move an instance of module  $m$  from one station of the system  $s$  to another station  $s$ ;
- Store an instance of module  $m$  from one station of the system  $s$  to a warehouse/stock;
- Insert an in-house (existing) instance of module  $m$  from the warehouse to a certain station  $s$ .

For each path  $\gamma$ , that is every possible path that visit one scenario for each  $v \in V$ , the objective function should be composed by the cost of each independent scenario-solution with a cost-correction due to the transition from one scenario  $o_i$  in period  $v$  and the following  $o_j$  in period  $v + 1$ . The complete form of the objective function is explicit in Appendix A.

The total cost  $Z_{tot,\psi_i}$  depends on the particular permutation of solutions and is calculated as follows:

$$Z_{tot,\psi_i} = \sum_{\gamma \in \Gamma(\psi_i)} Z_\gamma \cdot \pi_\gamma \quad (8)$$

where  $Z_\gamma$  is the sum of the cost of single-scenario solutions among the path  $\gamma$ , and  $\pi_\gamma$  is the probability of following a particular path  $\gamma$  and is calculated as  $\pi_\gamma = \prod_{\{i\} \in \Gamma(\psi_i)} \pi_{i,i+1}$ .

The multi-period, multi-scenario problem becomes:

$$\min_{\psi_i \in \Psi} Z_{tot,\psi_i} \quad (9)$$

That is, finding the permutation of solutions  $\psi_i$  that minimizes the total system cost.

#### 4. SOLUTION APPROACH

The solution method is heuristic. Each scenario is solved independently as single-period problem, and a certain number of near-optimal solutions are generated (depending on the specific requirements, this number can be controlled and it is user-dependent). By sampling one solution from each single scenario, a permutation of solutions at the scenario level is generated. The result is a set of solutions  $\Sigma$ , and the corresponding set of permutations  $\Psi$ . From this point, the heuristic follows an exhaustive enumeration over all the  $\psi_i \in \Psi$ . More in detail, the procedure follows these steps:

- List all the permutations  $\psi_i \in \Psi$ .
- Calculate the set of feasible paths for each permutation  $\Gamma(\psi_i)$ .
- For each permutation, calculate the solution of single-scenario problems for all the scenarios along the feasible paths as in equation (3) - (7).
- For all the feasible paths, evaluate the expected total cost as in equation (8).
- Solve the minimization problem as in equation (9).
- Enumerate the cost of solutions found and select the minimum ones.

The choice of such an approach is compliant with the ReCaM project requirements, and it fits to the size and

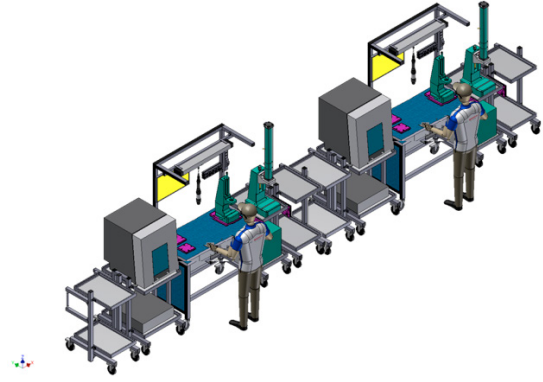


Fig. 3. The 2-stations production system used in the case-study (in the third period, a third station has been added to the system)

complexity of the targeted problems. Indeed, the tool is part of a modular platform that requires the computation of several KPIs for each of the solutions provided.

#### 5. REAL CASE STUDY

The proposed methodology has been applied to an industrial use case in the ReCaM project. The use case involves an assembly line for the production of valves (the 3D-model of the line is in Fig. 3). The designed assembly system needs to be capable of producing 6 product types. For the design process 19 modules have been identified as the result of a matching procedure, which has been verified together with the Mechatronic Objects supplier according to their capability in satisfying processing requirements and achieving KPIs.

After this initial procedure, the Flexible System Engineering Platform (FSEP) software, which embeds the approach proposed in this paper is used to perform the analysis. The time horizon considered in the design is a 3 year period featuring 5 different scenarios as in Fig. 1. The single-scenario sub-problems have been solved and 11 solutions have been drawn for each of the 5 scenarios, for a total of  $|\Psi| = 161051$  permutations. Each of these solutions represent the configuration of an assembly line composed by no more than 3 semi-automatic stations. The complete enumeration algorithm took around 10 minutes to be solved on a DELL XPS13 laptop with 8 GB memory and INTEL 2.3-GHz, i7 processor.

The optimal Pareto solutions provide the initial system configuration, and subsequent reconfiguration actions corresponding to scenario paths. For each period, the investment, reconfiguration and operational costs are calculated by using the performance evaluation and optimization tools. Solutions can be clustered according to user-defined KPIs. The robustness of selected optimal solutions are finally verified using a discrete event simulation tool, and these system designs can be visualized through a 3D visualization tool.

Following is a list of solutions with specific characteristics that are chosen to demonstrate the implications of reconfigurable solutions.

- Solution A is the best solution in terms of total expected cost among all the 161051 alternatives.

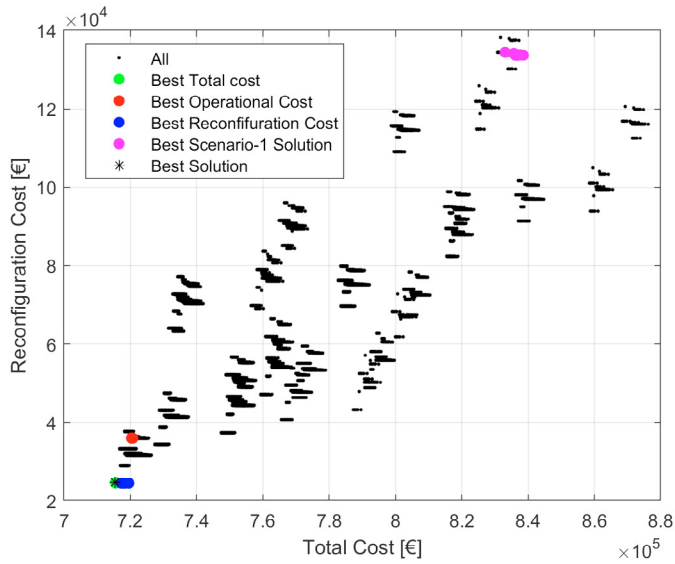


Fig. 4. Results when applying the FSEP to a case-study.

- Solution B is the worst solution: it has the highest expected total cost among the 161051 alternatives.
- Solution C is the solution composed of configurations that provide the minimum cost in each scenario if considered independently.
- Solution D is the solution having the lowest reconfiguration cost over the time horizon.
- Solution E covers the requirements of all the scenarios including the highest product-demand of (scenario 5) without any need for reconfiguration.

Solutions' plot is shown in Fig. 4. From the graph it can be seen that solutions regarding different system KPIs may not correspond. Moreover, it can be seen that there could be a substantial difference among the solution optimized over all the paths and the optimal solution considering only the first period (single-scenario). Table 2 lists the expected total costs of these solutions.

Table 2. Solutions comparison in terms of total expected cost

Configuration	Cost [€]	Difference from optimal solution [%]
A	715,430	-reference-
B	876,256	22.5%
C	719,558	0.6%
D	734,233	2.6%
E	966,447	35.1%

This preliminary application to the use-case resulted in a total cost reduction of 26% compared to the conservative solution (i.e. the solution found by covering from the most adverse scenario over the time horizon). We agree that these results are the proof of applicability of the proposed method to a vast spectrum of applications.

## 6. CONCLUSIONS AND FUTURE RESEARCH

A methodology for supporting the green-field design of reconfigurable assembly systems by considering a multi-period problem, characterized by uncertain product-mix and demand scenarios is proposed. Using this method, the

system configuration decisions taken at the initial design phase can consider possible future system modifications that might be needed within the planning horizon. This approach allows to identify optimal designs that are capable of quickly and efficiently adapting towards product variant and production quantity changes. Therefore, this can lead to increase in the operational efficiency of system and an improved guarantee level to achieve the predefined set of target KPIs. The application on a real industrial case demonstrates the performance advantages that can be derived by using the method and the implication of reconfigurable systems. Indeed, as shown by the results, the optimal solution is not always the one that minimizes the total cost over a certain time horizon, but the system configuration that is more adaptable to the anticipated future changes.

Future works will extend the model formulation in order to include spatial constraints, product routing constraints, layout constraints, as well as specific user-defined constraints. The search for the global optimal will be performed using appropriate solution methods. Although complete enumeration can be a fast approach for a small number of scenarios, in complex cases with high number of scenarios, efficient and faster solution techniques are needed. To this scope, techniques such as genetic algorithms, branch-and-bound algorithms, and neural networks are among the best candidates for the next developments.

## REFERENCES

- Asl, F.M. and Ulsoy, A.G. (2003). Stochastic optimal capacity management in reconfigurable manufacturing systems. *CIRP Annals*, 52(1), 371 – 374.
- Becker, C. and Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3), 694 – 715. Balancing Assembly and Transfer lines.
- Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data.
- Bertsimas, D., Brown, D., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review*, 53(3), 464–501. doi:10.1137/080734510.
- Bi, Z.M., Lang, S.Y.T., Shen, W., and Wang, L. (2008). Reconfigurable manufacturing systems: the state of the art. *International Journal of Production Research*, 46(4), 967–992.
- Birge, J.R. and Louveaux, F. (1997). Introduction to stochastic programming.
- Demir, L., Tunali, S., and Eliyi, D. (2014). The state of the art on buffer allocation problem: A comprehensive survey. *Journal of Intelligent Manufacturing*, 25(3), 371–392.
- Gamberi, M., Gamberini, R., Manzini, R., and Regattieri, A. (2008). An analytical model to evaluating the implementation of a batch-production-oriented line. *International Journal of Production Economics*, 111(2), 729–740.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Brussel, H.V. (1999). Reconfigurable manufacturing systems. *CIRP Annals*, 48(2), 527 – 540.
- Mehrabi, M.G., Ulsoy, A.G., and Koren, Y. (2000). Reconfigurable manufacturing systems: Key to future manu-

- facturing. *Journal of Intelligent Manufacturing*, 11(4), 403–419.
- Renna, P. (2010). Capacity reconfiguration management in reconfigurable manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 46(1), 395–404.
- Son, S., Olsen, T.L., and YipHoi, D. (2001). An approach to scalability and line balancing for reconfigurable manufacturing systems. *Integrated Manufacturing Systems*, 12(7), 500–511.
- Tolio, T. (2008). *Design of flexible production systems*. Springer.
- <http://recam-project.eu/> (2015-2018). Recam project.
- Wazed, M.A., Ahmed, S., and Nukman, Y. (2010). A review of manufacturing resources planning models under different uncertainties: State-of-the-art and future directions. *South African Journal of Industrial Engineering*, 21(1), 17–34.

#### Appendix A. RECONFIGURATION ACTIONS: TERMS OF THE OBJECTIVE FUNCTION

The variable  $Nstock_{m,o}$  indicates how many instances of module  $m$  are in the warehouse in a certain scenario  $o$ . Assume to go from scenario  $o1$  to  $o2$ , as follows:

$$Nstock_{m,o2} = (Nline_{m,o1} - Nline_{m,o2} + Nstock_{m,o1})$$

If  $Nstock_{m,o2}$  is negative, it means that  $|Nstock_{m,o2}|$  buying actions are needed. In order to consider other reconfiguration actions, we can define  $\delta_{m,s,o1,o2}$ :

$$\delta_{m,s,o1,o2} = \xi_{m,s,o2} - \xi_{m,s,o1}$$

Therefore:

- When  $\delta_{m,s,o1,o2} = 0$  nothing change and there is no reconfiguration action for module  $m$ ;
- When  $\delta_{m,s,o1,o2} = -1$  a module  $m$  was in station  $s$  in scenario  $o1$  and it has been uninstalled in scenario  $o2$ ;
- When  $\delta_{m,s,o1,o2} = 1$  a module  $m$  is installed in  $s$  in scenario  $o2$ .

Following are all the terms composing the objective function in section 3.5. Particularly, suppose without loss of generality to start the path  $o1 \rightarrow o2 \rightarrow o3$ :

- Cost of independent solutions in scenarios  $o1, o2, o3$ :

$$Z_\gamma = Z_{o1} + Z_{o2} + Z_{o3} \quad (A.1)$$

- Correction of cost in scenario  $o2$  and  $o3$  (cost of investment and installation of machines and buffers):

$$\begin{aligned} & - \sum_{m \in M} (CMinv_m + CMinstall_m) \cdot Nline_{m,o2} - \\ & - \sum_{k \in K} \sum_{b \in B} (CBinv_b + CBinstall_b) \cdot n_{b,k,o2} \end{aligned} \quad (A.2)$$

$$\begin{aligned} & - \sum_{m \in M} (CMinv_m + CMinstall_m) \cdot Nline_{m,o3} - \\ & - \sum_{k \in K} \sum_{b \in B} (CBinv_b + CBinstall_b) \cdot n_{b,k,o3} \end{aligned} \quad (A.3)$$

- Cost of having a stock in scenarios  $o1, o2, o3$  (It is assumed that  $Nstock_{m,o1} = 0$ ):

$$+ \sum_{m \in M} CMstock_m \cdot \max\{Nstock_{m,o2}; 0\} \quad (A.4)$$

$$+ \sum_{m \in M} CMstock_m \cdot \max\{Nstock_{m,o3}; 0\} \quad (A.5)$$

where:

$$Nstock_{m,o2} = (Nline_{m,o1} - Nline_{m,o2} + Nstock_{m,o1})$$

$$Nstock_{m,o3} = (Nline_{m,o2} - Nline_{m,o3} + Nstock_{m,o2})$$

- Cost for buying new MO instances in scenarios  $o2$  and  $o3$  with respect to the preceding scenario:

$$+ \sum_{m \in M} CMinv_m \cdot \max\{-Nstock_{m,o2}; 0\} \quad (A.6)$$

$$+ \sum_{m \in M} CMinv_m \cdot \max\{-Nstock_{m,o3}; 0\} \quad (A.7)$$

- Reconfiguration cost for transition  $o1 \rightarrow o2$  and  $o2 \rightarrow o3$  (uninstallation and installation):

$$\begin{aligned} & + \sum_{m \in M} \sum_{s \in S^{(\gamma)}} [CMinstall_m \cdot \max\{\delta_{m,s,o1,o2}, 0\} + \\ & CMuninstall_m \cdot \max\{-\delta_{m,s,o1,o2}, 0\}] \end{aligned} \quad (A.8)$$

$$\begin{aligned} & + \sum_{m \in M} \sum_{s \in S^{(\gamma)}} [CMinstall_m \cdot \max\{\delta_{m,s,o2,o3}, 0\} + \\ & CMuninstall_m \cdot \max\{-\delta_{m,s,o2,o3}, 0\}] \end{aligned} \quad (A.9)$$

where  $S^{(\gamma)} = S^{(o1)} \cup S^{(o2)} \cup S^{(o3)}$  is the union of set of stations in  $o1, o2$  and  $o3$ .

- Cost for buying new buffer capacity in scenarios  $o2$  and  $o3$  with respect to the preceding scenario:

$$\begin{aligned} & + \sum_{k \in K^{(\gamma)}} \sum_{b \in B} (CBinv_b + CBinstall_b) \\ & \cdot \max\{n_{b,k,o2} - n_{b,k,o1}; 0\} \end{aligned} \quad (A.10)$$

$$\begin{aligned} & + \sum_{k \in K^{(\gamma)}} \sum_{b \in B} (CBinv_b + CBinstall_b) \\ & \cdot \max\{n_{b,k,o2} - n_{b,k,o1}; 0\} \end{aligned} \quad (A.11)$$

where  $K^{(\gamma)} = K^{(o1)} \cup K^{(o2)} \cup K^{(o3)}$  is the union of set of buffers in  $o1, o2$  and  $o3$ .