Modular design for quality and cost

Bruno Agard, PhD
Ecole Polytechnique
Montréal, QC
bruno.agard@polymtl.ca

Samuel Bassetto, ing. jr. PhD
Ecole Polytechnique
Montréal, QC
samuel-jean.bassetto@polymtl.ca

Abstract—The purpose of this article is to help managers in early design of new product families. The proposal includes a single level module design formulation that considers quality and cost simultaneously. The method for testing the proposed algorithm is based on a case study of an electro-mechanical assembly device. The performance of the algorithm is compared to a design without modules. The main result is a model and an algorithm that optimizes quality (resp. cost) under the constraints of cost (resp. quality). It shows what modules to manufacture, in what quantities, and in which products to use them. The output also provides the predicted quality and cost, based on improvements made to the modules. This research enables the joint optimization of quality and cost by defining the modules to be manufactured.

Keywords: modularity, design for quality, design for cost, assembly, optimization

I. INTRODUCTION

Mass customization and pricing competition force companies to develop new strategies to cope with greater flexibility, while remaining competitive in terms of price and delivery time [1]. These strategies are undoubtedly key elements in gaining competitive advantage, or at least remaining competitive. However, customers require fully functional products, whatever the price. Their tolerance of product malfunctions is often very low. If a product is labeled “industrial”, whether it is a low cost one (T-Shirt, computer flash drive, pen) or a low volume one (airplane, substation circuit breakers, wind turbine), the manufacturer is expected to have fully understood its characteristics. The product is supposed to work properly and faultlessly, and any variability in its functions can be considered a risk to meeting the customer’s requirements.

Manufacturers put controls in place to master every level of their processes [2], and barriers are deployed throughout the manufacturing system to prevent faults from occurring [3]. While the integration of quality and quantity has been investigated in classical manufacturing lines [4], the interaction between quality and supply chain design in modular design has not, and constitutes an opportunity for investigation.

The concept of quality adopted in this paper conforms to the view of manufacturers and supply chain managers. It is the degree of conformance of products to predefined specifications and standards. This degree is measured by processes controls and inspections. Actions on quality have an impact on a global defect rate. Through this paper, the degree of conformance will be appreciated by the final failure rate of a product.

Preassembled parts affect this indicator. These subsets are called modules, and they are employed to solve diversity issues, like determining an optimal threshold manufacturing quantity. The creation of modules should leads to efficiencies in terms of reduced assembly time and overall cycle time, while maintaining high potential for diversity. When modules are produced from components, resulting modules may have different quality level than its components, depending on actions that have been performed during its manufacturing. Resulting quality of a module could be increased (for instance modules could be “sort” or “test”) or decreased (for instance in cases of handlings that produce scratches or default on modules). There is then a possible action on quality each time a module is created, being a positive one or a negative one.

This paper investigates module design considering quality, cost, and the product family-mix. It is structured in four parts. We presents in section 2 our model; in section 3, a case study; and in section 4, the results.

II. MODEL

A. Description of the product

Consider \( P \) to be a set of products to manufacture. Product \( P_k \) is made up of a set of components \( C_i \). \( r \) products are considered: \( k \in [1, r] \). The components are called \( C_i \), \( i \in [1, p] \). In its description, a product is represented as a vector of size \( p \) that expresses the components that are present (1) or absent (0) in it. For example, \( P_1=\{1, 0, 1, ..., C_{i=1}, ..., C_{p}=0\} \) means that product \( P_1 \) contains components \( C_1, C_3, ..., C_p \), but not \( C_2, ..., C_{p-1} \), and so on. Every product \( P_k \) has a failure rate \( \rho(P_k) \) and a cost \( \text{Cost}(P_k) \), and must be produced in a certain quantity \( Q(P_k) \).

For each product \( P_k \), if a failure rate (resp. cost) constraint is to be solved \( \rho(P_k) \neq 0 \); (resp. \( \text{Cost}(P_k) \neq 0 \) then \( \text{Cost}(P_k) \) (resp. \( \rho(P_k) \)) indicates 0. This modeling is used for the solving branching in equations (1 and 2).

Different products \( P_k \) may contain the same components \( C_i \). A module \( M_j \) is a set of components \( C_i \). Various options are available in product manufacturing: Option (A): uses all the necessary individual components for each product. This is the basic assembly process. Option (B): uses a mix of components and modules for each product. The module creation process is described in the following.

In Figure 1, following option A, product \( P_1 \) would be made of raw assembly of components \( C_1, C_2 \) and \( C_3 \) and \( P_2 \) would be...
made of C₂, C₃, C₄. In options B, a module M₁ is made of C₂ and C₃, and we see different way to use this module in P₁ and P₂. Then product P₁ would be made of C₁ and M₁ and C₂ would be made of C₄ and M₁. The design of M₁ enables actions on its costs and its quality.

Figure 1 shows different options for manufacturing products P₁ and P₂.

**Figure 1: Problem description**

A quality issue is understood as a failure that occurs in a product, a component, or a module. A failure can occur during manufacture or during an assembly operation. It can also appear later, at which point it is referred to as a reliability issue. Its main characteristic is to propagate along the supply chain, and such problems are rarely detected by classical functional tests.

In manufacturing the products in P, workers select the components (or modules) needed from different lots. Each lot comes from a specific contractor who guarantees the reliability of the entire lot. The failure rate for the set of modules Mⱼ is ρ(Mⱼ). For each lot, ρ(Mⱼ) ∈ [0, 1], ρ(Mⱼ) = 0 means that all the products are reliable, ρ(Mⱼ)=1 means that 100% of the modules Mⱼ are faulty. Failure rates depend on:

- The ability of suppliers to produce reliable components
- The ability of suppliers to identify unreliable components in their processes
- The ability of shipping and incoming departments to identify quality defects

The same applies to the cost of each component and module. In option (A) (Figure 1), the failure rate and cost of P₁ will depend on components C₁, C₂, C₃, and in option (B₁), the failure rate and cost of P₁ will depend on C₁ and M₁, and so on.

Depending on the objective for each type of product (in terms of cost and failure), we look to answer the following questions: Is it better to buy a module M₁ instead of two components C₂ and C₃? and Is it better for P₁ and/or P₂ to use this module M₁?

### B. Mathematical modeling

The notations are the following:

- Cᵢ is a component;
- C is the set of components Cᵢ, Cᵢ ∈ [1, p];
- Mᵢ is a binary vector of size p, called module j. The vector represents the components it contains; for example, module M₁=(1, 0, —, 0) contains only component C₁. A component can also be considered like a module (with only one component);
- M is the set of modules Mⱼ, Mⱼ ∈ [1, q];
- Pₖ is a binary vector of size p, called product k. The vector represents the component it requires; for example, product P₁=(1, 0, 1, ... , C₁=1, ..., Cₚ=0) means that product P₁ contains components C₁, C₃, ..., Cₚ, ..., but not C₂, ..., Cₚ, and so on.
- P is the set of products Pₖ, Pₖ ∈ [1, r];
- Cost(Cᵢ), Cost(Mⱼ), and Cost(Pₖ) are the cost of Cᵢ, Mⱼ, and Pₖ respectively;
- ρ(Cᵢ), ρ(Mⱼ), and ρ(Pₖ) are the failure rates of Cᵢ, Mⱼ, and Pₖ respectively;
- Q(Pₖ) is the quantity of products Pₖ to manufacture;
- x is a binary vector of size q, such that xⱼ=1 if Mⱼ ∈ M'. It is the decision variable.

The goal is to determine the subset of modules M' ⊆ M of minimum cost, such that all products in P can be built, each product Pᵢ respecting its own constraints. If a product Pᵢ has no failure rate constraint, then ρ(Pᵢ) = 0, in which case product Pᵢ has a maximum cost constraint to respect. If a product Pᵢ has no cost constraint, then Cost(Pᵢ) = 0, in which case product Pᵢ has a maximum failure rate constraint to respect.

Two kinds of constraints exist in P: for r₁ products, there is a maximum cost constraint, and for r₂ products, there is a maximum failure rate constraint (r₁+r₂=r). Each product in P may have a different constraint.

We call (Figure 2):

- Aeq a binary matrix of size q.p formed by all products in P.
- Afailure a vector of size q that contains failure rates for all modules in M.
- Ascost a vector of size q that contains costs for all modules in M.
The mathematical formulation is the following:

$$\min_x C(x)$$

such that

for all $k$ in $[1, r]$,

if $\text{Cost}(P_k)=0$, $A_{\text{failure}} \cdot x \leq \rho(P_k)$

if $\rho(P_k)=0$, $A_{\text{cost}} \cdot x \leq \text{Cost}(P_k)$

$$A_{\text{eq}}^T \cdot x = P_k$$

where

$$C(x) = \sum \delta_{jk} \text{Cost}(M_j) \cdot Q(P_k) + \sum x_j \cdot G$$

$$\delta_{jk}=1$$, if product $P_k$ contains module $M_j$

If $\text{Cost}(P_k)=0$ a quality constraints is to be solved for product $P_k$ (equation 1), if $\rho(P_k)=0$ a cost constraints is to be solved for product $P_k$ (equation 2). $C(x)$ is the total cost of the whole product family, and represents the sum of the costs of all the necessary modules (based on the quantity of products, and so the number of each type of module), plus the total number of modules multiplied by a management cost $G$ per module. The management cost has been shown to have a major impact on the number of modules in the final product solution [5]. For computation purposes, $G$ is assigned a fixed value for all the experiments, so that the quality and cost of modules can be compared for analysis. This problem includes the Set Partitioning Problem (Equation 3), which is then NP-hard in the strong sense [6].

C. Problem solving

This optimization problem cannot be solved by standard optimization software for large instances. As explained previously, the problem is an NP-hard 0–1 optimization problem. In order to determine an approximate solution, we adopted a simulated annealing procedure. Figure 3 depicts the general scheme of the algorithm.

After the input data have been read (a description of the products to manufacture, a list of possible modules, and the set of parameters : $G$ and $d$ that will be explained in the following), the first step, called "weight and filter", follows. For each module, the number of use cases is evaluated (a simple comparison of $M_j$ and $P_k$, where $P_k(i)$ should always be higher than or equal to $M_j(i)$). The use case value of each module represents its weight. A weight of 2, means that the module could potentially be used in 2 different products. All modules with a weight equal to 0 are deleted from the search space. This is the filter operation.

An initial solution is selected and evaluated. The initial solution ($x$) is constructed, such that it contains nothing but modules with only one component. This means that only components are considered at the start of the process. Modules from $x$ are added/removed to improve the solution, as follows: With an initial value $x$, $C(x)$ (Equation 4) is evaluated, which is the initial temperature. For every constraint, Equations (1) to (3), that is not respected, a penalty is added. $Best(x)$ I $C(x)$, $x^*$ $x$, Level $\leftarrow$ 0 and $Iteration$ $\leftarrow$ 0.

A neighbourhood of $x$ is constructed, and two alternatives are considered:

- If $x$ does not permit the manufacture of all products, respecting all constraints, a module is added to $x$, and we obtain $x'$.
- If $x$ permits the manufacture of all products, a random process decides whether to add or remove a module from $x$, and we obtain $x'$.

The module to be added or removed is randomly selected, the random process being weighted with the use case number of each module. Modules with a large (small) weight are more likely to be selected to be added (removed).

1. $C(x')$ is evaluated in a similar way to that in step 2. $Iteration$ $\leftarrow$ $Iteration$ $+$ 1.
2. If \( C(x') \leq C(x) \) (with respective penalties), then the neighbor is accepted: \( x \leftarrow x' \) and Level \( \leftarrow 0 \), otherwise go to step 7; if \( C(x') \leq Best(x) \), then the best solution is recorded: \( Best(x) \leftarrow C(x') \) and \( x^* \leftarrow x' \).

3. A random number \( \alpha \) is compared to \( p(\text{Level}) \), if \( \alpha \leq p(\text{Level}) \), \( x \leftarrow x' \), otherwise \( x' \) is rejected.

4. If Level \( \geq \text{Max Level} \), \( p(\text{Level}) \) is updated; if Iteration \( \geq \text{Max Iteration} \), the optimization process is stopped.

5. All modules in \( x^* \) that do not appear in any product \( P_k \) are removed, \( C(x^*) \) is updated. \( x^*, C(x^*) \) is the list of non feasible products, and the evaluations of all \( P_k \) are given.

6. A random number \( \alpha \) is compared to \( p(\text{Level}) \), if \( \alpha \leq p(\text{Level}) \), \( x \leftarrow x' \), otherwise \( x' \) is rejected.

7. If Level \( \geq \text{Max Level} \), \( p(\text{Level}) \) is updated; if Iteration \( \geq \text{Max Iteration} \), the optimization process is stopped.

8. All modules in \( x^* \) that do not appear in any product \( P_k \) are removed, \( C(x^*) \) is updated. \( x^*, C(x^*) \) is the list of non feasible products, and the evaluations of all \( P_k \) are given.

III. CASE STUDY

A. Description of the product

This case study is structured around the modular design of headlamp devices.

**Figure 4: Illustration of the head-lamp device.**

The device presented in Figure 4 produces a maximum of 700 Lumens to keep the cost of the components low (under $70). Its weight is less than 250g. It has 10 functional parts, among them a lamp, a battery pack, a microcontroller, and a switch. It is made up of 56 components. Many options can be accommodated on this device. The case study is made of 8 options, 15 functions, 7 constraints, 11 products, 1 supplier per function and 1 quality grade per function. It is presented in Table 1.

Every component performs a specific function, and is defined with a cost and a quality rating (evaluated based on its failure rate). It is possible to preassemble these modules.

The failure rate and the cost of a module both depend on the components it contains.

<table>
<thead>
<tr>
<th>Function</th>
<th>Name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>LED (mounted on a star PCB, with 2 connections per pole)</td>
<td>Warm</td>
</tr>
<tr>
<td>F2</td>
<td>PCB, 3.7V, 5-level driver (on, off, low, middle, high)</td>
<td>White</td>
</tr>
<tr>
<td>F3</td>
<td>Batteries with voltage, driver, and charger</td>
<td>2 positions; strong; waterproof (to 100m and gas resistant)</td>
</tr>
<tr>
<td>F4</td>
<td>Battery case</td>
<td>Case for helmets with short cables; waterproof</td>
</tr>
<tr>
<td>F5</td>
<td>Switcher</td>
<td>Case to be carried manually with long cables; waterproof</td>
</tr>
<tr>
<td>F6</td>
<td>Battery case</td>
<td>For Camp</td>
</tr>
<tr>
<td>F7</td>
<td>Battery case</td>
<td>For Petzl Eclat-roc</td>
</tr>
<tr>
<td>F8</td>
<td>Battery output charger</td>
<td>For Petzl Elos</td>
</tr>
<tr>
<td>F9</td>
<td>Battery case</td>
<td>For Militarian</td>
</tr>
<tr>
<td>F10</td>
<td>Straps for helmet</td>
<td>Navy Blue</td>
</tr>
<tr>
<td>F11</td>
<td>Battery waterproof reinforcement</td>
<td>Militarian</td>
</tr>
</tbody>
</table>

**Table 1: Details of the 15 functions selected for the test.**

The following has been adapted for computation purposes:

The failure rate of a module or is the sum of the failure rate of the components it contains minus \( d \). It is a positive value.

\[
\rho(M_i) = \max \left( \sum_{i \in M_i} \rho(C_i) - d ; 0 \right)
\]

(5)

The failure rate of a product is the sum of the failure rate of the modules it contains minus \( d \). It is a positive value.

\[
\rho(P_k) = \max \left( \sum_{i \in P_k} \rho(M_i) - d ; 0 \right)
\]

(5bis)

In both equations (5) and (5bis) \( d \) is a positive value; it corresponds to a quality operation that increases the overall quality of the set of modules that is concerned. For example \( d \) could represent the result of a sorting operation that is performed after the module has been assembled. The failure rate of the module lot is then decreased.

**Failure rate**

It is assumed by this choice that chosen modules act as key elements of the product. By the way the failure of one of them generates a failure of the product and impacts its quality. This assumption is acceptable for core parts of a product. Even if inside such core components redundancy is organized, globally the component will be perceived as an entity with improved quality characteristics. The action on quality is modeled by the quantity \( d \). It is also assumed that \( d \) will not change over time. This assumption is a strong limitation, as every enterprise has continuous improvement programs. Nevertheless we decide to keep this variable as constant to manage a tractable model.

**Cost**

The cost of a module is computed related to the components included. We adopted the following formulation

\[
\text{Cost}(M_i) = (1 - a) \left( \sum_{i \in M_i} \text{Cost}(C_i) \right)
\]

(6)

\( a \) is a positive value \( [0 ; 1] \). It indicates that a module is less expensive than the sum of the components it contains.

So, the cost and failure rate of a finished product \( P_k \) are directly linked to the modules and the components selected for its manufacture.
B. Technical constraints of this scenario and numeric values

Not all combinations of components result in a technically or commercially feasible product. The following constraints are observed:

The head lamp device contains only one type of LED then each product Pk contains F1 or F2, not both. Also only a 4-levels or a 5-levels PCB device is allowed (F3 or F4). The switch is reinforced or not reinforced (F5 or F6). The battery cases made for helmets or to be carried manually (F7 or F8). If the batteries are carried manually, then there are no helmet straps (if F8, then not F9, F10, F11). Helmet straps depend on the helmet (F9 or F10 or F11). One of two colors is available in each product (F14 or F15). A feasible product must contain either F1 or F2, but not both. Also, if a final product contains F8, it will not contain F9, F10 or F11.

This set of functions and the related constraints return 2,930 possible different modules. Our study proposes to select the set of modules that will permit the manufacture of the following set of final products.

<table>
<thead>
<tr>
<th>Function</th>
<th>Cost (in $)</th>
<th>Failure rate (10^-6)</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>23</td>
<td>0 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2</td>
<td>30</td>
<td>0 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>60</td>
<td>1 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>27</td>
<td>0 0 0 0 0 0 1 1 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>3</td>
<td>1 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>4</td>
<td>2 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F7</td>
<td>35</td>
<td>1 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F8</td>
<td>30</td>
<td>0 0 0 0 0 0 1 1 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F9</td>
<td>4</td>
<td>3 0 0 0 0 0 1 1 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F10</td>
<td>1</td>
<td>1 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F11</td>
<td>4</td>
<td>5 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F12</td>
<td>2.5</td>
<td>3 1 1 1 1 1 0 0 1 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F13</td>
<td>20</td>
<td>5 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F14</td>
<td>2</td>
<td>1 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F15</td>
<td>3</td>
<td>1 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Functions and products

Table 2 contains different models of head lamps for manufacture. For example, P1 is a lamp for cavers. It must be reliable (an expected failure rate of 15.10^-6), and we would like to provide it at the lowest possible cost. This lamp contains functions F2, F3, F5, F7, F10, and F12. Based on a simple assembly of raw components, the resulting product, P1, will have failure rate of 15.10^-6, and a final cost of $131.50.

In order to test the algorithm, we decided to consider a function (F11) that is not necessary in any product. Table 3 presents the quality (in number of failures per 10^6 products) and cost (in $) expected for each product to be manufactured, as well as the quantities of products to manufacture (in thousands). For instance, for product P1, the constraint is to obtain an overall failure rate lower than 16.10^-6 at the lowest possible cost. The quantity produced is to be 50. Comparing these numbers with those in the last two lines of Table 2: for product P11, if each component is assembled individually, the final cost will be $131.50; however, the overall quality will not meet the failure rate requirement (17 > 16). In some cases, raw material assembly is an acceptable solution for the market, but the product could still be improved from a cost (or quality) perspective. For instance, for product P11, the required level for the failure rate is 20.10^-6; and, with raw material assembly, it is possible to achieve 17.10^-6.

Table 3. Constraints, objectives, and quantity per product

In our case study, we noted the quantities produced by a lamp manufacturer: 790. 10^3 products were ordered and split into 11 product types. The results from this case study are presented in the following section.

IV. RESULTS

We consider here the above-defined problem. Modules are manufactured and assembled under cheaper conditions than raw materials, and a sorting operation makes it possible to discard a few of the problematic modules. For computational purposes, a=0.05 and d=1 for Equations 5 and 6. As explained previously, G (the management cost of a module) has a major impact on the number of modules. In the following, G=300. The penalty cost for each non-feasible product is $1,000. For the simulated annealing procedure, the parameters are the following: Max_Iteration = 500 and number of levels = 3, with a p(Level) of 0.4, 0.2, and 0.1 respectively; also Max_Level = 100 iterations.

Figure 5 shows that, starting with 7 non-feasible products, modules are added to the current solution until all the products are feasible. This point is reached after 85 iterations. The algorithm seeks to improve the objective function by removing modules until some products become non feasible, adding to modules for feasibility and removing them for improvement. The end of the process had selected 28 modules. The final solution is 5 times cheaper that the solution that requires raw material assembly. The 28 modules selected make it possible to produce all the required products (respecting both constraints), and it is the best solution found up to now in terms of cost.

Figure 5. Number of modules (left axis) and non feasible products (right axis).
C(x). Some modules are removed from the best solution and 18 are proved to be sufficient to solve the problem.

Table 4. Module composition matrix.

Table 4 shows the solution made up of 18 modules (M1 to M18), obtained from the assembly of several functions. For instance, Module 5 has a cost of $61.50 and a failure rate of 1.10^{-6}. It is a package made up of F2 and F7. The use case of every module is presented in Table 5. For instance, Product 4 contains M2, M3, and M10.

Table 5. Product x Module Matrix

Results are provided in Table 6.

For instance, product P1, ordered in a quantity of 50,000, will have a final cost of $126,425 and a failure rate of 16.10^{-6}. Note that the raw assembly solution (noted in the remainder, C0) can be made up of two parts (quality and cost): for example, 17.10^{-6} and $131.50. This method retrieves a better solution for each parameter (quality and cost). Globally, the algorithm outperforms C0. This example proposes modular design as a solution to cope jointly with quality and cost constraints. Instead of performing a raw component assembly, the modules have to be defined. During preassembly operations, a quality assessment can be carried out, so that the modular design, combined with quality and cost control, becomes part of the continuous improvement cycle of the manufacturing system. The example proves that it is possible to address a particular market by simultaneously considering modularity, quality, and cost control.

V. CONCLUSION

This paper is about modular design. It takes into account the actions taken on cost and quality every time a module is used, which enables the production of a particular product family, each product respecting is own constrains. These results are really encouraging, as they constitute the initial solution for an industrial team wishing to reduce the discrepancy between marketing needs and manufacturing system abilities. This gap is filled by joint action on the modules, that is, action in terms of quality and costs. With their partners’ quality management and cost management skills, this team can try to achieve better performance in terms of market coverage. This research opens up opportunities for further study. The first concerns the influence of quantity on the stability of the module. The second concerns the introduction of a list of suppliers, along with their relative performances. Taking this information into account could lead to an optimum manufacturing solution, or to a robust one, which might be more costly but more resilient to disruptions. Finally, a 1-level module design has been proposed here, and a multilevel modeling should be considered as well.

VI. REFERENCES


