Co-Evolution of Product Families and Assembly Systems

A. Bryan, J. Ko, S. J. Hu (2), Y. Koren (1)
Department of Mechanical Engineering
The University of Michigan, Ann Arbor, MI, USA

Abstract
To cope with the intense global competition that is characterized by high product variety and short life cycles, manufacturers need to share manufacturing systems across products and product generations. Co-evolution of product families and assembly systems is proposed as a novel methodology for the joint design and reconfiguration of product families and assembly systems over several product generations. The co-evolution methodology capitalizes on the opportunities for design and assembly system reuse that are offered by modular product architectures and reconfigurable assembly systems. As a result, co-evolution can lead to reduced product development costs and increased responsiveness to market changes.

Keywords:
Assembly System, Product Family, Co-evolution

1 INTRODUCTION
In today's market environment, manufacturers are faced with the challenge of cost-effectively supplying high variety within short product development times. Co-evolution of product families and assembly systems is introduced as a new product development methodology for the joint design and reconfiguration of assembly systems within and across product generations. The co-evolution methodology can enable manufacturers to remain competitive as it maximizes the reuse of product modules and reconfigurable systems [1] to ensure that manufacturing systems are effective for as many product generations as possible [2].

Various techniques for the reduction of product development time and costs have been proposed. Concurrent engineering is one such technique. Early concurrent engineering techniques focused on the integration of the detailed design phase with manufacturing [3]. More recent concurrent engineering techniques integrate the earlier conceptual product design phase with manufacturing system design in order to simultaneously determine product family and manufacturing system costs [4-6].

Figure 1 illustrates the main difference between co-evolution and traditional concurrent engineering strategies. Co-evolution is a method for the incorporation of product variants and assembly system changes within a family generation, as well as between product generations, through continuously reconfiguring modular products and manufacturing systems. With concurrent engineering techniques, usually each generation of products is associated with a unique assembly system.

![Figure 1: Comparison between concurrent engineering and co-evolution of product families and assembly systems.](image-url)
When products change from one generation to another, new assembly systems must be simultaneouslydesigned. Some researchers have previously recognized a need for product evolution or co-evolution [7-11]. However, systematic methods for such co-evolution have not been proposed.

This paper proposes methodologies for the co-evolution of products and assembly systems. The remainder of this paper is organized as follows. The details of the co-evolution methodology are explained in Section 2. This will be followed by an example in Section 3 and conclusions in Section 4.

2 METHODOLOGY

There are two main phases of the co-evolution methodology: (1) the joint design of the product family and assembly system in the first generation and (2) the later co-evolution of the product family and system (Figure 1).

The first generation design is assumed to be a 'clean sheet' design for both the product family and the assembly system. This initial design phase is important as the decisions made during this phase affect product family changes and assembly system reconfigurations in future generations. The co-evolution phase pursues economical product family changes and assembly system reconfigurations. As shown in Figure 1, the inputs to this phase are the existing product family, the required design changes, and the re-configuration constraints of the product family and assembly system. Using these inputs, new product modules are developed, the product family is modified and the assembly system is reconfigured. Since reconfiguring an assembly system can consume a lot of resources, a method for determining the effectiveness of the co-evolution plan is also proposed.

2.1 Phase I: design of the first generation product family and assembly system

The seller’s welfare utility model has traditionally been used for the selection of product variants for a product family [12-13]. The selected product family is then used as an input to assembly system design formulations such as mixed product assembly line balancing (MPALB). MPALB is used to determine task assignments and the optimal number of workstations [14]. Since the product family influences the design of the assembly line and the cost of the assembly system, it influences the product variants selected, the seller’s welfare models and MPALB models are combined to jointly determine the product family and assembly line design that leads to maximum profits. The maximum profit (\(\Pi\)) is achieved by maximizing revenues and minimizing costs as follows:

\[
\max \Pi = \sum_{j \in \text{PF}} \rho_j z_j - \phi M
\]

The decision variables are the product variants to be included in the product family (\(x_j\)), the product variants selected by customers (\(x_{jm}\)), the number of customers interviewed for the product family and the subscripts \(i\), \(j\), \(k\) and \(m\) are customers, product variants, modules and workstations respectively. The upper limit of \(i\) is \(I\); the number of customers interviewed; the upper limit of \(j\) is \(|PF|\); the cardinality of set \(PF\); the upper limit of \(k\) is \(K\); the maximum number of modules; \(M = \sum_{m=1}^{K} y_m\); \(\psi\) is the number of potential customers in the market. In Eq. 1, \(\rho_j\) is the selling price of one unit of product variant \(j\) and \(\phi\) is the combined fixed and variable costs for the assembly system.

The constraints for the problem are as follows:

\[
a_{ij} z_j - a_i \geq y \left( y_j - 1 \right) \quad \forall i = 1, \ldots, I; \; \forall j \in PF
\]

\[
a_{ij} z_j - a_i z_j \geq y \left( y_j - 1 \right) \quad \forall i = 1, \ldots, I; \; \forall r, j \in PF; \; r \neq j
\]

\[
\sum_{j \in PF} z_j \leq 1 \quad \forall i = 1, \ldots, I
\]

\[
s_j \leq z_j \quad \forall i = 1, \ldots, I; \; j \in PF
\]

\[
\sum_{j \in PF} z_j \leq J
\]

\[
\sum_{k=1}^{K} x_{km} = 1 \quad \forall k = 1, \ldots, K
\]

\[
\sum_{k=1}^{K} \sum_{j \in PF} v_{j,k} x_{km} \leq \lambda y_m \quad \forall m = 1, \ldots, K
\]

\[
x_{km} = \sum_{n=1}^{y_m} x_{mn} \quad \forall m = 1, \ldots, K; \; \forall k = 1, \ldots, K; \; \forall q \in P(k)
\]

\[
y_m \leq y_n \quad \forall n, m = 1, \ldots, K; \; n \leq m
\]

\[
z_j \in \{0, 1\}; \; x_{km} \in \{0, 1\}; \; s_j \in \{0, 1\}; \; y_m \in [0, 1]
\]

In Eqs. 2 and 3, \(a_{ij}\) is the utility that customer \(i\) receives from product variant \(j\); \(a_i\) is the utility that customer \(i\) receives from an existing product, and \(y\) is a very large number. Eqs. 2 and 3 ensure that the customer utility for the selected product variant exceeds that of other product variants in the product family and of the existing product. Eq. 4 indicates that each customer selects at most one product variant and Eq. 5 prevents the customer from selecting a product variant not offered in the product family. Eq. 6 ensures that the total number of product variants selected for the product family does not exceed \(J\), the limit set by marketing. Eq. 7 ensures that each assembly task is assigned to a workstation. Eq. 8 prevents the overloading of workstations and tasks from being assigned to unavailable workstations. In this constraint, \(z_{j,k}\) is the time required to assemble module \(k\) of product variant \(j\) and \(\lambda\) is the planned life of the assembly system. Eq. 9 prevents precedence constraint violations. In this constraint \(P(k)\) is the set of predecessors of module \(k\). Eq. 10 ensures that lower numbered stations are filled before higher numbered stations. Eq. 11 contains the feasibility constraints on the decision variables.

The profit maximization formulation outlined in Eqs. 1-11 is a non-linear integer program. These problems are known to be NP hard and therefore difficult to solve by exact optimization methods. Therefore, a genetic algorithm is used to solve this problem.

2.2 Phase II: joint reconfiguration of the product family and assembly system

As customer preferences change, new product functions need to be introduced and the assembly system updated. The reconfiguration decisions for the product family and assembly system are based on the new customer requirements and the constraints of the existing assembly system. They include determination of which of the modules with new functionality should be added, how they should be combined with existing product variants to form new product variants, whether any of the existing product variants should be dropped from the product family, and
how much additional resources would be required to supply the new product family. Adopting a new product family (\(\Pi^\prime\)) would only make sense if it allows the manufacturer to improve profits over the profits obtained from an existing product family (\(\Pi\)). Therefore, the objective function of the second phase of co-evolution is to maximize profit increase (\(\Delta \Pi\)) as follows:

\[
\begin{align*}
\text{maximize} & \quad \Delta \Pi = \Pi^\prime - \Pi \\
\text{subject to} & \quad \sum_{j=1}^{M} b_{j} Y_{j} = \sum_{m=1}^{M} \sum_{k=1}^{K} j_{mk} x_{mk} + \sum_{m=1}^{M} \sum_{k=1}^{K} t_{km} x_{km} \leq \sum_{m=1}^{M} \sum_{k=1}^{K} c_{m} x_{km} \\
& \quad \sum_{j=1}^{M} \sum_{k=1}^{K} j_{mk} x_{mk} \leq \sum_{m=1}^{M} \sum_{k=1}^{K} t_{km} x_{km} \leq \sum_{m=1}^{M} \sum_{k=1}^{K} c_{m} x_{km}, \\
& \quad x_{mk}, y_{j}, z_{jk}, t_{km} \in \{0,1\}.
\end{align*}
\]

In this formulation, the previous product family's profit is a parameter. The new decision variables introduced for the co-evolution problem are \(x_{mk}\), \(y_{j}\), and \(z_{jk}\). \(x_{mk}\) indicates whether option \(\ell\) of module \(k\) is in product variant \(j\). This leads to the design of new product variants or the redesign of previous product variants. \(x_{mk}\) is the station assignment variable for new modules and \(c_{m}\) is the number of parallel resources required at each workstation. Since the assembly system already exists, \(x_{mk}\) is considered as a parameter and \(M\) is the upper bound of \(m\). \(k^\prime\) represents new modules and has an upper limit \(K^\prime\), and \(\ell\) is a subscript for options of new modules or options and has upper limit \(L^\ell\). An intermediate variable, \(z_{jk}\), depends on the revenue that could be previously obtained from the product variant \(\ell\) and the revenue that can be obtained from the new modules \(\{z_{jk}\}\).

All the constraint equations, except Eq. 8, remain the same with \(x_{lm}\) being replaced by \(x_{km}\) where appropriate.

Eq. 8 is modified to Eq. 13 to account for changes in the workload assigned to each workstation as the product family evolves. This is required because although the module-workstation assignment remains the same, the volume of previous modules can change. Eq. 13 ensures that the time required to assemble new and old modules at a workstation does not exceed the total available time.

\[
\sum_{d=1}^{D} \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{J} (\theta_{l} x_{mk} + \phi_{m}) \leq \Delta_{m} x_{km} \quad \forall m = 1, \ldots, M
\]

where \(\theta_{l}\) is the time required to assemble one unit option \(\ell\) of module \(k\) and \(\phi_{m}\) is the current unit workload of all the modules previously assigned to workstation \(m\) for product variant \(j\). The feasibility constraints for the new decision variables are as follows:

\[
\theta_{l} x_{mk} \in \{0,1\}; \quad x_{km} \in \{0,1\}; \quad c_{m} \geq 0
\]

As in Section 2.1, this formulation is a nonlinear integer program and is solved using a genetic algorithm.

### 2.3 Effectiveness of the co-evolution plan

A reconfiguration process should be efficient in terms of (1) reuse, (2) time and (3) transition cost. The level of reuse is measured by the reuse ratio which is defined as the ratio of the number of system elements reused to the total number of system elements [2].

\[
\tau(r(\delta, \delta+1)) = \frac{\sum_{e=1}^{E} r_{e}(\delta, \delta+1) \cdot n_{e}}{\sum_{e=1}^{E} n_{e}}
\]

\(\tau(r(\delta, \delta+1))\) and \(\tau(r(\delta, \delta+1))\) are the transition reuse ratio and the transition reuse ratio for system element \(e\) in making \(\delta\) and \(\delta+1\) changes to the product family, \(E\) is the total number of system elements, \(\tau_{e}\) is the weight factor for system element \(e\). The transition time for updating the assembly system \(u(\delta, \delta+1))\) is defined as the maximum time required for changing each of the system elements between two versions of the product family \(u_{e}(\delta, \delta+1))\).

\[
u_{e}(\delta, \delta+1) = \max_{e \in E} [u_{e}(\delta, \delta+1) : \forall e]
\]

This time should be as short as possible in order to minimize production losses during co-evolution. The efficiency of these transformation activities can be measured in terms of financial units as follows:

\[
u_{e}(\delta, \delta+1) = \sum_{e=1}^{E} t_{e}(\delta, \delta+1)
\]

3 ILLUSTRATION

The example outlined in this section demonstrates the implementation of the co-evolution methodology.

### 3.1 Generation 1 product family and assembly system

The first generation (Gen 1) product family is derived from 11 modules (M1-M11). Modules M1-M6 occur in every product variant and do not have options. They are referred to as base modules. The other modules have options. The first option is not to have the module and the remaining options are variations of the module. The base product variant is defined as the product variant without any optional modules.

The precedence diagram for the assembly is shown in Figure 2. This precedence diagram is a modification of the one given in [14]. Each task in the precedence diagram is assumed to represent the assembly of an individual module. This assumption allows the serial number for an assembly task to be consistent with the serial number of the module assembled at the respective assembly task. The assembly times for the base product variant are also shown in Figure 2. Since the first node on the precedence diagram represents the introduction of the first module to the line, its associated task time is set to zero. The assembly times for the second option of M9, M10 and M11 are 2min, 9min and 2min, respectively. The assembly times for the third option of M10 and M11 are 3min and 5min respectively.

![Figure 2: Precedence diagram and task times (min) for the Gen 1 base modules.](image-url)
The reuse value of the system is 0.98. Upgrade to the system as early as possible. An example was used to illustrate the implementation and benefits of the co-evolution approach.

**5 REFERENCES**


