Analysis and Optimization of Assembly Lines Feeding Policies

Antonio C. Caputo*, Pacifico M. Pelagagge**, Paolo Salini**

*Department of Engineering, University of Roma Tre, via della Vasca Navale 79, 60100, Roma – Italy (acaputo@uniroma3.it)
** Department of Industrial Engineering, Information and Economics, University of L’Aquila, Zona industriale di Pile, 67100, L’Aquila – Italy (pacifico.pelahagge@univaq.it, paolo.salini@univaq.it)

Abstract

Purpose
In assembly systems, components have to be uninterruptedly fed to workstations. Selecting the materials delivery policy among the available alternatives (i.e. line-stocking, kitting or just in time supply) is a complex task as the above methods have both advantages and drawbacks. Moreover, literature lacks of comparative research on factors influencing the choice of kitting respect other line feeding methods, and a comprehensive methodology to assist in this system decision is still missing. As a result, policy selection is often based on subjective qualitative judgement, and firms may switch several times from one policy to the other without being sure which system is best for their environment. In order to contribute to a solution of this problem, a method is proposed allowing to define the optimal assignment of feeding policies to each component type, in order to minimize total systems cost.

Design/methodology/approach
In the paper an integer linear programming model is developed for the optimal choice of feeding policy, where a specific feeding method is assigned to each component in order to minimize a cost based objective function, factoring in workers cost, investment costs, work in process holding cost and floor space occupation cost. In the paper at first the formulation of the problem is carried out in the framework of mathematical programming, detailing the choice of decision variables, the structure of objective function representing overall system cost to be minimized, and constraints. Subsequently, a case study is described showing the application of the methodology to demonstrate its capabilities.

Originality/value
The proposed approach can represent a powerful and general purpose quantitative decision tool for production managers to assist in the selection of proper policies for components delivery to assembly systems at an early decision stage.

Keywords: Assembly lines feeding, Linear programming optimization, Kitting, Line storage.

1. Introduction
Design and management of assembly lines requires a decision about the way components and subassemblies are delivered to assembly stations. Three policies are typically available, namely line-stocking, kitting and kanban-based continuous supply (i.e. just in time supply). In a kitting policy (Brynzér 1995; Brynzér and Johansson 1995; Bozer and McGinnis 1992; Caputo and Pelagagge, 2011) no parts inventories are kept at the assembly stations and the specific assortment of components required to perform the assembly operations are grouped together and placed into a kit container. Kits are prepared in a stockroom and delivered to the assembly line according to the production schedule. In case of no kitting (i.e. continuous supply), every different part number is supplied in an individual container to the assembly line. Small sized containers may be moved in just in time fashion adopting a kanban-based policy. Otherwise, components containers holding bulk quantities are simply stored along the line and periodically replenished (line-stocking). Overall, in kitting systems material flow through the shop floor is simplified as only kits need to be moved to the assembly line instead of individual components containers, and WIP is minimized at the point of use. However, order picking associated with kit preparation is labour intensive thus an increased workforce is required. Just in time delivery or line stocking save the order picking labour required for kitting and guarantee continuous availability of stock at the assembly line, at the expense of a greater space utilization on the shop floor, and higher WIP along the assembly line. The just in time approach allows to somewhat reduce WIP
This optimization model is based on the descriptive approach capabilities in the context of a large scale occupation cost. In this work the latter model is further costs, work in process holding cost and floor space component type by factoring in workers cost, investment to define the optimal assignment of feeding policy to each linear programming decision making framework, allowing explicitly including all three feeding policies in a n integer adopting a more comprehensive cost model, and by subsequently extended (Caputo et al. 2010) by allowing a comparative research on factors influencing the choice of feeding respect other line feeding methods, and the fact that a comprehensive methodology to assist in this system decision is still missing.

In recent times a growing interest has been witnessed in the literature about qualitative selection criteria of material handling approaches to delivery components to assembly lines (Wanstrom and Medbo 2008, Hanson 2012). However, quantitative analyses have been mainly limited to a direct comparison of kitting versus line stocking (Hua and Johnson 2010, Bozer and McGinnis 1992, Carlsson and Hensvold, 2008). Battini et al. (2009) instead compare trolley to work station, pallet to work station and kit to assembly line approaches. Caputo and Pelagagge (2008, 2011) suggested to adopt an ABC class-based approach to develop hybrid feeding policies where each item class is associated to a given policy, instead of feeding all components using the same policy. Their work is subsequently extended (Caputo et al. 2010) by allowing a genetic algorithm to choose the feeding policy at the level of single item. Finally, Limère et al. (2012) utilize a linear programming model to make an optimal assignment of parts to kitting or line stocking policies. Nevertheless, they neglect the just in time alternative and only consider workers cost related to picking, transportation, kit preparation, material replenishment. Caputo et al. (2012) further extend the approach of Limère et al. (2012) by adopting a more comprehensive cost model, and by explicitly including all three feeding policies in an integer linear programming decision making framework, allowing to define the optimal assignment of feeding policy to each component type by factoring in workers cost, investment costs, work in process holding cost and floor space occupation cost. In this work the latter model is further refined and a realistic case study is develop to verify the approach capabilities in the context of a large scale assembly line in the automotive sector.

2. Model description

This optimization model is based on the descriptive models developed by Caputo and Pelagagge (2011) to design and estimate performances of kitting, line storage and just in time components delivery systems. In this work the model is based under the hypothesis of one warehouse with a single I/O point, and a single-product assembly line consisting of M workstations arranged serially. However, the model can be easily extended to multi-product and multiple lines cases. The assembly line has a constant daily production volume and the considered time horizon for cost estimation is one day. We also assume that each component can be used on multiple workstations. Material handling personnel is distinct from line staff which is in charge of assembly operations only. A kit is here defined as a unit load holding all the components required to assemble one unit of the finished product. However, materials composing a single kit may be put into one or more separate containers according to weight and space limitations. Kits are prepared one at a time and delivered from the kitting area located at the warehouse I/O station to the first station of the line, then they travel along the line together with the product being assembled (traveling kit). Multiple kits can be accumulated at the start of the line.

In kanban and line stocking systems each workstation has its own containers available, and containers are not shared among multiple workstations. In kanban policy material is resupplied at each station with a lead time LT in separate containers dedicated to each component type. The required number of containers, therefore, depends on the daily consumption of parts and the replenishment lead time. In line stocking each station holds separate containers for each distinct component it uses, periodically resupplied at time intervals which depend from the adopted containers capacity. Constant-speed vehicles or walking operators are used to transport kit containers and components containers. However, different kind of vehicles, and containers sizes, can be used for different feeding policies. Empty containers are returned back to the central warehouse for replenishment. Exceptional items, such as those very cumbersome and heavy, requiring special handling care, are not included in the analysis.

In this model one and only one feeding policy is assigned to each type of component needed for assembling the end product, and the assignment is made in order to minimize a cost-based objective function. Binary decision variables are \( x_{ijp} \in [0,1] \), where \( i = 1 \) to N is the component type identifier, and \( p = [1, 2, 3] \) is the policy identifier with the following meaning, namely in a kitting policy \( p = 1 \), in a line storage policy \( p = 2 \), and in a just in time (kanban-based) policy \( p = 3 \). A decision variable has value \( x_{ijp} = 1 \) in case policy \( p \) is assigned to component \( i \), and \( x_{ijp} = 0 \) elsewhere. Cost items included in the model are personnel cost (this includes operators needed to fraction bulk cartons and pick components in the warehouse, to deliver materials to the workstations, as well as workforce engaged in kits preparation), investment cost (containers, storage racks and transport vehicles), WIP holding cost (proportional to the average level of inventory at the stations or along the line), and space occupation cost which is proportional to the floor space occupied by
accumulated stock at the workstations and specific floor space cost. Owing to the formulation adopted when building the optimization model, either resource sizing and cost functions are expressed in marginal (or incremental) terms, i.e. as the additional contribution to resource consumption and overall cost when the i-th part type is assigned to a given policy. In this manner, irrespective of the chosen part type-policy assignment, the overall cost is simply computed by summing over all part types their individual cost contribution. In particular, we define $C_{WP,i}$ the marginal workforce cost (€/day) incurred when policy $p$ is selected to deliver component type $i$. $C_{E,i}$ the marginal equivalent investment cost (€/day) incurred when policy $p$ is selected to deliver component type $i$. $C_{WIP,i}$ the marginal work in process holding cost (€/day) incurred when policy $p$ is selected to deliver component type $i$. $C_{S,i}$ the marginal space occupation cost (€/day) incurred when policy $p$ is selected to deliver component type $i$. Below the assumptions utilized to derive cost items are discussed respectively for picking ($p=1$), line storage ($p=2$) and kanban just in time ($p=3$).

### 2.1 Computation of Marginal Workforce Cost

In a kitting policy the equivalent number of kit containers required to hold parts of type $i$ per unit end item is

$$n_{c,kit,i} = \frac{V_i \cdot n_i \cdot p \cdot p_i}{V_{kit}}$$

which depends on containers volume ($V_{kit}$) or their allowed weight ($p_{max}$), as well as the unit weight ($p_i$), the volume ($V_i$) and the consumption of parts per unit end item ($n_i$) of the i-th part type. Unless the same containers are reused twice or more each day, the total equivalent number of containers required to manage the i-th part for a daily production $D$ is $n_{cont,kit,i} \times D$.

Workers are required to locate and reach components at their storage locations in the warehouse to feed the kitting area, to pick individual parts and place them into kit containers, and to move containers from the kitting area to the start of the line. The average time to reach the storage location of i-th part and return to the kitting area is ($t_{fr}$) and can be estimated according to the adopted picking policy and warehouse configuration. We can also assume that when reaching a warehouse location the operator can pick a quantity of parts of type $i$ enough to complete $Q$ separate kits (with $Q$ integer and $\geq 1$) in order to avoid returning to that warehouse location for each new kit to be prepared, or that a single trip to a warehouse location allows to retrieve $Q$ different part types that are stored in nearby locations. Kitting operation can be quite time consuming, as it generally includes the following operations, namely counting/weighting of parts to ensure that the right number is included in the kit, preparation of components before insertion in the kit (i.e. cutting to measure, package removal, cleaning and quality control); kit preparation (insertion of parts in the right sequence and in the proper housing slot, including correct positioning control); compilation of missing component list for subsequent kit completion. Average time $t_{pick}$ required to pick and kit one unit of a part type can be measured as well or computed resorting to traditional time studied methods. As far as kit transportation is concerned, instead, the equivalent number of daily moves for the i-th part type (from the warehouse to the line with full containers and from the line to the warehouse with empty containers) $n_{cont,kit,i}$, being the number of container simultaneously transported by the material handling vehicle, is $2 \times n_{cont,kit,i} \times D/a$. We assume that each trip involves $k$ operators and that the one-way trip time is $t_m$, estimated on the basis of plant layout and material handling vehicle average velocity. From the above assumptions the number of daily workers required to prepare and transport kits can be computed assuming that each operator works a daily shift of $h$ hours and has an efficiency $\eta_{op}$. Then the number of workers times their wage rate $C_{wage}(€/day)$, allows to compute the marginal personnel cost for i-th part type allocated to a kitting policy.

$$C_{M,i} = C_{wage} \left( \frac{t_{fr} + t_{pick} + t_{fr} + t_{pick} \times k}{\eta_{op} \times h} \times D \times n_{cont,kit,i} \right)$$

In a line storage or just-in-time policy, containers dedicated to a part type are moved from the warehouse, or from a supermarket area in case of just-in-time policy, to the workstation utilizing that part. Conceptually the main difference is the size of containers and the resulting frequency of handling moves. In both cases $Z_i = \min \left( \frac{V_i \cdot n_i \cdot p_i}{V_{kit}}, \frac{V_i \cdot n_i \cdot p_i}{p_{max}} \right)$ is the equivalent number of parts of type $i$ which fit in a container. For simplicity here we assume that size $V_i$ of a container is specific of the feeding policy, i.e. that size is different in case of just-in-time or line storage policies, but is the same for all types of parts handled with a given policy. However, the user is free to specify a specific container size for any type of part if needed. Obviously $Z_i$ changes, for any given part type, according the container size and weight limits associated to each policy. It follows that the number of equivalent containers to be moved daily towards station $j$ in case the i-th part type is assigned to one of the two above policies (with zero buffer stock) becomes $(D \times n_i/Z_i)$, where $n_i$ is the number of items of the i-th component utilized at the j-th station per unit finished product ($n_i = 0$ if component $i$ is not utilized at station $j$). We assume that the average one-way trip duration is $t_{ws}$ and that a time $t_{fr}$ is required to fraction bulk containers in the warehouse to fill part containers to be handled to the line (in case of just-in-time this includes time to replenish the supermarket with the smaller sized container). The average time to reach the warehouse storage location of i-th part is $t_{fr}$. Therefore, the marginal personnel cost for i-th part type allocated to a line storage or just-in-time policy is

$$C_{M,i} = C_{wage} \left( \frac{t_{fr} + t_{pick} + t_{fr} + t_{pick} \times k}{\eta_{op} \times h} \times \sum_{j} \left( D \times n_j / Z_j \right) \right)$$
Please note that while the equation is the same, the values of \( t_{wi}, t_{wi}, t_{wi}, k, Z_i \) and \( \theta \) depend from the chosen policy, given the different size of containers and material handling devices utilized in line storage or just-in-time contexts.

### 2.2 Computation of marginal investment cost

Equipment cost \( C_{E} \) is the sum of containers cost and capital investment of storage (i.e. containers) and transport equipment (i.e. forklifts, manual carts, tractor/trailers etc.). Operating cost of transport equipment (energy and maintenance) are neglected here, but can be included in the equivalent daily cost \( (C_{E}) \) of the vehicle if needed. In a kitting policy the investment includes kit containers and kits transportation vehicles. The marginal daily investment cost is

\[
C_{i2} = C_{V} \sum j(D_{n, ij}^{\text{kit}}) + C_{V} \left( \frac{2 D_{n, kit, i} L_{w}}{\omega V_{C}, h} \right)
\]

where \( C_{V} \) is the container daily equivalent unit cost, while \( L \) is the one way trip length from the kitting area to the line start and \( V_C \) the vehicle average velocity, so that \( L/V_C \) is equivalent to term \( t_{wi} \) in Equation (2). We assume that each set of kit containers is used only once a day and replenished the next day. In a line storage policy the investment includes line storage containers, containers transportation vehicles, and storage racks at the workstations. Containers cost is computed as the number of containers required to transport and hold parts at the workstations times the unit container cost. In line storage a single container per part type is usually left at each workstation utilizing that component, while \( NS_i \) is the number of different workstations utilizing the i-th part type. However we double the container number to account for the fact that while containers are left at the workstation an equal number is being replenished at the warehouse allowing to swap an empty container for a full one. Racks, having a daily cost per unit volume \( C_{SRU} \), are instead needed only to hold the containers transported and left at the workstation. The amount of rack space depends from the unit container volume \( V_C \). The equivalent number of vehicles needed to transport part containers at the workstation, considering that the vehicle can transport \( \omega \) containers simultaneously (with \( \omega \geq 1 \) and integer), is computed as the ratio of the required daily transportation hours (i.e. number of required equivalent trips \( \sum (D_{n, i}/\omega Z_i) \) times the two way trip duration \( (2L/V_C) \)) to the available daily work hours \( (h) \).

\[
C_{i2} = C_{E} 2N_{S} + C_{V} \left( \frac{2 L_{w}}{\omega V_{C}, h} \sum \left( \frac{D_{n, i}}{Z_i} \right) \right) + C_{SRU} N_{S}, V_{C}
\]

In case of a just-in-time policy the computation is similar, but multiple daily replenishment of smaller sized containers are required, so that the same container can be used more than once. The actual number of containers left at the j-th workstation to ensure the desired parts throughput is linked to the replenishment lead time through Little’s law \( (D_{n, i} L_{T}/Z_i) \). In this case too the overall number of containers is doubled because empty containers are swapped with full containers coming from the supermarket.

\[
C_{i3} = C_{E} \sum j \left( \frac{2 D_{n, i} L_{T}}{Z_i} \right) + C_{V} \left( \frac{2 L_{w}}{\omega V_{C}, h} \sum \left( \frac{D_{n, i}}{Z_i} \right) \right) + C_{SRU} V_{C} \sum \left( \frac{D_{n, i} L_{T}}{Z_i} \right)
\]

### 2.3 Computation of marginal work in process holding cost

In case of a kitting policy, considering that each of the \( M \) stations holds one traveling kit, that \( C_{i1,1} \) is the daily unit holding cost of \( i \)-th part type, which is expressed as the monetary value of the component times the daily carrying cost per unit value of WIP, and that at the start of the line each kit holds the entire multiplicity \( n_i \) of the \( i \)-th part type included in one unit of the end product, while it is empty at the end of the line, then the holding cost of the average amount of work in process, is

\[
C_{i1} = \frac{1}{2} N_{S_i} \left( C_{i1} n_i \right)
\]

where \( C_{i1} \) is the component daily equivalent unit cost. We assume that each kit container is used only once a day and replenished the next day. In a just-in-time policy the average amount of WIP is \( Z_i/2 \), given that each container holds a maximum of \( Z_i \) items and assuming that a single container is left at the workstation. This average WIP should be multiplied for the number of different workstations holding the part type. Therefore

\[
C_{i1}^{\text{WIP}} = \frac{1}{2} N_{S_i} \left( C_{i1} n_i \right)
\]

### 2.4 Computation of marginal space occupation cost

As far as space requirements on the shop floor are concerned, the space occupied by the single traveling kit can be neglected. In case multiple kits are transported and accumulated at the first station, the marginal space consumption cost is the unit daily floor space cost \( (C_{FS}) \) times the marginal occupied floor space,

\[
C_{i1}^{\text{FS}} = C_{FS} \left( \frac{n_i}{a_i x b_i} \right)
\]

where \( n_i \) is the number of containers which can be stacked in a column and \( a_i \) and \( b_i \) are the length of the kit containers base sides. In case of line storage we assume that up to \( n_i \) containers having a base area \( (a_i x b_i) \) can be stacked and that a stack can be composed of containers holding different part types. The allocation of a part type to a line storage policy implies the space occupation of at
least one additional container (assuming only one container per part type is held at each workstation utilizing that part), but this results in the increase of the actual footprint on the shop floor only when the overall number of containers exceeds a multiple of \( n_0 \). In this case, in fact, a new stack has to be erected or a new column of storage rack has to be added. To keep the model linear we adopted the following formulation for the marginal space occupation cost, even if this expression underestimates the actual space occupation as long as the number of containers is lower than an integer multiple of \( n_0 \). However, the larger the number of part types to be line stored is the lower the error.

\[
C_{i,j}^3 = C_{js} NS \left( \frac{(a_i b_j)}{n_{sl}} \right)
\]  

(11)

Finally, in a just-in-time policy, being \( D_{nj} \) LT/Zi the number of containers to be stored at the \( j \)-th workstation for a part type, and being \( n_{0j} \) the stack height, the marginal space occupation cost is assumed as

\[
C_{i,j}^3 = C_{js} \left[ \sum_j \left( \frac{(a_i b_j)(LTD_{nj})}{Z_i \ n_{0j}} \right) \right]
\]  

(12)

In this case too the previous warning of possible cost underestimation applies.

### 2.5 Optimization model

After defining the above expressions for marginal costs and resource consumption, the integer linear programming optimization model can be stated as

\[
\text{Minimize } \sum_{i=1}^{N} \sum_{p=1}^{J} (C_{i,p}^M + C_{i,p}^E + C_{i,p}^{WIP} + C_{i,p}^S) x_{i,p}
\]  

subject to

\[
\sum_{j=1}^{3} \left( x_{i,j} C_{i,j}^3 \right) + \sum_{j=1}^{3} \left( c_{i,j} x_{i,j} \right) \leq S_{j, av}
\]  

(14)

\[
\sum_{i=1}^{N} C_{i,j}^3 x_{i,j} \leq S_{ls}
\]  

(15)

\[
\sum_{p=1}^{J} x_{i,p} = 1
\]  

(16)

\[
x_{i,p} \in \{0,1\}
\]  

(17)

In this model Eq. (14) is the overall daily cost function and its components are computed resorting to Equations (2 to 12). Equation (14) states that for each workstation \( j \) (with \( j = 1, 2, \ldots, M \)), given the set \( J \) of component types used at a generic \( j \)-th workstation, the actual occupied space for parts storage (represented by the left hand term) is less or equal to the available floor space (\( S_{js, av} \)) at the station. This constraint is applied only with reference to the floor space consumed for stocking parts containers (in case of just in time or line storage policies) as the space occupied by the single travelling kit is considered to be negligible. Constraint (15) ensures that space occupied by kit containers at the first workstation is consistent with available space \( S_{ls} \) for multiple kits storing at the start of the line. This check does not apply to other station because kits enter the line one at a time and only one kit is present at any time at any workstation. Constraint (16) imposes that only one policy is assigned to any part type and that each part type is assigned to a delivery policy. Finally, Eq. (17) imposes that any decision variable can only have 0 or 1 values. When the model is not used for long range planning of new assembly systems, but rather to reorganize the material supply systems of existing facilities, then plant managers may also have the additional requirement of not hiring new operators. If this applies then the following constraint can be included in the model formulation.

\[
\frac{1}{C_{op}} \sum_{i=1}^{N} \left[ x_{i,1} C_{i,1}^M + x_{i,2} C_{i,2}^M + x_{i,3} C_{i,3}^M \right] \leq N_{op, av}
\]  

(18)

In the above equation the left hand term represents the overall number of operators needed to feed all the components with the allocated policies, while the right hand term represents the available number of operators. In this simplified model no constraint forces the number of workers, vehicles, containers and stacking columns on the shop floor to have an integer value (which, instead, is an actual physical constraint) otherwise the problem becomes non-linear owing to the discontinuities arising in the cost functions and constraints. Therefore, the obtained solution should be manually transformed to a feasible solution by rounding to the next integer the obtained value of the above parameters, even if this may involve an increase in the cost function value and a deviation from the optimal solution.

### 3. Case study

In order to show the capabilities of the method a case study from the automotive sector is considered here. The case study refers to a final assembly line for industrial vehicles. The line is composed of 97 workstations. Each vehicle is assembled using 1852 different part types. The multiplicity of a given part type can change in the 1 to 34 range. Part types have a weight range from 10 [g] to 41.2 [kg], and a volume variable between 24 [cm³] and 0.33 [m³]. Unit cost of parts ranged between 0.1 and 205.9 €. The single-model assembly line has a throughput of 18 end units per day. Storage space at workstation is limited to a length along the line of 8 m, with a depth equal to the depth of storage container.

Additional system data and assumed parameters are resumed in Table 1. The model was implemented and solved resorting to Matlab computing environment. When running the model the following optimal assignment of delivery policies to component types is obtained, namely 75.8% of components are to be delivered in just in time manner adopting a kanban based policy; 18.3% of components are to be stocked at the workstations; while the remaining 5.9% of components should be kitted. This optimal strategy is much more cost effective than any pure policy, using a single delivery mode for all components, as
shown in Table 2. Table 2 also compares the theoretical optimal cost solution obtained by the solver and the subsequent feasible solution where all variables which must have an integer value (i.e. number of workers, number of vehicles, number of containers stacking columns on the shop floor etc.) have been rounded up to the next integer. As shown in the Table, an optimal supply policy assignment in this case study allows a cost reduction of about 60\%-70\% respect pure policies, thus confirming the effectiveness of the proposed method. Figure 1, instead, shows the subdivision of cost item according to the various supply policies. The figure shows that labor is the single most relevant cost and especially affects JIT and kitting policies aspects. WIP holding cost instead is more relevant in line storage and JIT policies.

In order to test model performance a sensitivity analysis has been made. Particularly the influence of working cost (figure 2), holding cost (figure 3), space cost (figure 4) have been studied. As shown in Fig. 2 we notice that when the operator cost is increased the percent relevance of policies with high labor consumption reduces and line storage policy is extended to a greater number of items.

Conversely, when the holding cost rises, as shown in Fig. 3, kanban policy is advantaged at the expense of line storage, which is affected by higher amount of stored material.

The same happens when the storage space cost increases, as shown in Fig. 4. Given that line storage consumes the largest amount of storage space, the higher is the surface area cost the lower is the adoption of a line storage policy.

4. Conclusions
In this paper an integer linear programming model has been developed to assign to each component to be delivered to an assembly line the optimal feeding policy, choosen between kitting, line stocking and just in time supply alternatives. Results show that the selection of items feeding modes is not a trivial matter and that given the different characteristics and value of components, to adopt the same feeding policy for all items can be a poor solution. Intuition based approaches, qualitative assessments, or rules of thumbs can lead to unsatisfactory results as well, while the optimal choice resulting from item-by-item analysis allows significant savings provided that the organizational burden to arrange different concurrent feeding systems on the shop floor can be tolerated. It is expected that the proposed approach can represent a powerful and general purpose decision tool for production managers to assist in the selection of proper policies for components delivery to assembly systems at an early decision stage.

References


