Sequencing Minimum Product Set in Mixed Model U-line to Minimize Work Overload

Jie Gao and Jinlin Li
Management School
Xi’an Jiao Tong University

June 1, 2009
Outlines

1. Introduction
2. Problem description & mathematical formulation
3. Properties of the MMUL sequencing problem
4. A single-pass heuristic algorithm
5. Conclusions
Mixed model assembly line (MMAL) is a widely used production manner, which can produce two or more kinds of products simultaneously with negligible setup cost. MMAL can provide high production rate and flexibility to some extent. MMAL can provide high production rate and flexibility to some extent.

Fig. 1 Illustration of straight mixed model assembly line

Model sequence

A B C D In

station 1

station 2

station M

Out

Product A  Product B  Product C  Product D
Introduction

- Different products may need different parts and different task time.
- In a station, the assembly time of a product may not be equal to that of another one.
- In a MMAL, if products of intensive workload arrive at a station successively, the station may have a overload.
- Hence, if products of high workload and products of low workload arrive at the station intermittently, such overload may be avoided.

The model sequencing problem in MMAL concerns in which sequence should products be launched into the line.

Fig.1 Illustration of straight mixed model assembly line

Model sequence: A B C D

Station 1

Station 2

Station M

Out

Product A  Product B  Product C  Product D

2009/6/1
As described in Boysen et al. (2007), two basic objectives are usually discussed in literatures:

- **Leveling part usage**: Get constant material requirements, related to JIT production philosophy.
- **Minimizing work overload**: If several work intensive products arrive successively, the worker may not be able to finish all the jobs inside the station boundaries and then overloads happen.
  - Overloads can be compensated by utility workers or conveyor stoppage.
  - Operators can also rush to finish the product within the boundary at the risk of quality problem.

---

Introduction

Studies on model sequencing problem in MMAL


Introduction

- Most of the studies on the topic of MMAL sequencing problem assume the production ends after the last job in the sequence is processed.
  - The objective is one run performance

- In order to reduce the computational complexity of product sequencing, producing a minimum product set (MPS) cyclically and continuously is becoming common in practice (Bolat, 1997).
  - The object is long run performance
  - Minimum product set is the smallest product set having the same proportion as the production target
Introduction

- U-shaped assembly lines exist in JIT production systems.
- Compared to straight lines, U-lines can achieve
  - improved visibility and communication
  - convenience for rebalance, multi-skilled workers.
  - fewer workers and shorter line
- Workers are allowed to do jobs in both side of the line.

Fig. 2 Illustration of mixed model U-line
Introduction

- **Studies on U-shaped assembly line**
Introduction

- Mixed model U-line (MMUL) receives few attention in research.
  - most of the literature on U-line discuss the single model U-line.

- In each cycle, there are two products arrive at one crossover station from the front and back side, simultaneously.

- In MMUL, these two products may of two different kinds, the sequencing even have a larger impact on the performance of the MMUL.

- The MMUL sequencing problem has not considered explicitly in the past research.
Problem description

- Our study on mixed model U-line sequencing problem
  - based on MPS principle
  - to minimize work overload.

- In our study, the production line is a U-shaped conveyor
  - some stations can span both side of the line, which are called “crossover stations”. The front (back) section of a crossover station is called “front (back) segment”.
  - Other stations are just the same as stations in straight line, which are called “straight stations” or “complete segments” in contrast.
Problem description

Assumptions:

- **Paced line.** Cycle time is given.
- Different products are launched into the line according to a basic sequence **cyclically**.
- All segments have **closed** upstream and downstream boundaries.
- The **processing time** of each product in each segment is **given and deterministic**.
- Products are processed obeying a **non-preemptive FCFS policy**.
- The **travel time of workers are ignored**.
- Workers can’t help each other.

For easy description, assuming:

- There is only 1 worker in a station.
- The downstream moving speed of the conveyor is 1.
- Stations in MMUL are numbered according the method in Miltenburg et.al (1994).
Problem description

- The worker moves along the conveyor to process products
- If he can’t finish a product before it leaves the segment, he gives up the job on hand and move to get the next job.
- If he reaches the upstream segment boundary before the next job arrives, he waits there.

Fig. 3 Illustration of a U-shaped conveyor with 2 crossover stations and basic sequence is BACD.
Problem description

Notation

Indices
- \( n \): index of product types \( n = 1, 2, \ldots, N \)
- \( m \): index of station \( k = 1, 2, \ldots, M \)
- \( k \): index of positions in the basic sequence \( k = 1, 2, \ldots, D \)
- \( T \): index of segment types. \( T = \text{s/f/b} \) represents complete/front/back segment, respectively

Parameters
- \( c \): cycle time
- \( N \): total number of product types
- \( M \): total number of stations
- \( d_n \): number of products from type \( n \) in MPS
- \( D \): total number of products in MPS, \( D = \sum_{n=1}^{N} d_n \)
- \( S \): set of indices of crossover stations
- \( l^T_m \): length of segment \( T \) of station \( m \)
- \( t^T_{m,n} \): processing time of product \( m \) in segment \( T \) of station \( m \)

Decision variables
- \( x_{n,k} \): equal to 1 if a product from type \( n \) is assigned to position \( k \) in the basic sequence, else equal to 0
Problem description

Notation

Calculated decision variables

\( sp_{m,k}^T \) distance from segment upstream limit when the worker in station \( m \) start processing the \( k \)th product at segment \( T \).

\( ep_{m,k}^T \) distance from segment upstream limit when the worker in station \( m \) stop processing the \( k \)th product at segment \( T \).

\( u_{m,k}^T \) amount of unfinished work when \( k \)th product is processed at segment \( T \) of station \( m \).

\( w_{m,k}^T \) amount of waiting time before \( k \)th product arrives at segment \( T \) of station \( m \).

\( \Pi \) the basic sequence according to which products are launched into the line

\( \Pi (k) \) index of the product type in the \( k \)th position of \( \Pi \)

In straight station \( m \), as described in literature:

\[
sp_{m,k}^s = \max \{ 0, ep_{m,k-1}^s - c \} \quad (1)
\]

\[
ep_{m,k}^s = \min \{ l_m^s, sp_{m,k}^s + t_{m,\Pi(k)}^s \} \quad (2)
\]

\[
u_{m,k}^s = \max \{ 0, sp_{m,k}^s + t_{m,\Pi(k)}^s - l_m^s \} = sp_{m,k}^s + t_{m,\Pi(k)}^s - ep_{m,k}^s \quad (3)
\]

\[
w_{m,k}^s = \max \{ 0, c - ep_{m,k-1}^s \} = sp_{m,k}^s + c - ep_{m,k-1}^s \quad (4)
\]
Problem description

In crossover station \( m \), the distance between the upstream boundary of front and back segment is

\[
L_m = \sum_{i=m}^{M} l_i^T + l_m^f
\]

Let

\[
p_m = \left\lfloor \frac{L_m}{c} \right\rfloor \quad a_m = L_m - cp_m
\]

(\( \left\lfloor x \right\rfloor \) is the maximum integer not larger than \( x \)). Then the jobs’ arrival sequence in station \( m \) is:

\(...(k+d_n)\)th product arrive the front segment
kth product arrive the back segment
(k+1+d_n)th product arrive the front segment...

This explain why worker in crossover station walk clockwise in figure 3
Our objective is to minimize the total overload during the production of a Minimum Product Set.

The overload depends on the operator position of each station at the beginning of the production of the MPS.
Problem description

- When the production is continuous and cyclic, the initial operator position at each station for each production of MPS may change.

- Question 1
  - When the production is continuous and cyclic, will the MMUL reach “steady state”,
  - i.e., the status of processing each PMS in each station will replicate every $D$ cycles
Problem description

In crossover station $m$

$$sp_{m,k}^f = \max\{0, ep_{m,k-p_m-1}^b - (c - a_m)\}$$

(5)

$$ep_{m,k}^f = \min\{l_m^f, sp_{m,k}^f + t_{m,\Pi(k)}^f\}$$

(6)

$$sp_{m,k}^b = \max\{0, ep_{m,k+p_m}^s - a_m\}$$

(7)

$$ep_{m,k}^b = \min\{l_m^b, sp_{m,k}^b + t_{m,\Pi(k)}^b\}$$

(8)

$$u_{m,k}^f = \max\{0, sp_{m,k}^f + t_{m,\Pi(k)}^f - l_m^f\} = sp_{m,k}^f + t_{m,\Pi(k)}^f - ep_{m,k}^f$$

(9)

$$u_{m,k}^b = \max\{0, sp_{m,k}^b + t_{m,\Pi(k)}^b - l_m^b\} = sp_{m,k}^b + t_{m,\Pi(k)}^b - ep_{m,k}^b$$

(10)

$$w_{m,k}^f = \max\{0, c - a_m - ep_{m,k-p_m-1}^b\} = sp_{m,k}^s + c - a_m - ep_{m,k-p_m-1}^b$$

(11)

$$w_{m,k}^b = \max\{0, a_m - ep_{m,k+p_m}^f\} = sp_{m,k}^s + a_m - ep_{m,k+p_m}^f$$

(12)
Problem description

When the production is continuous and cyclic, the MMUL will reach “steady state”, i.e., the status of processing each product in each station will replicate every $D$ cycles

Lemma 1: $\exists K \in N, \forall k > K, \forall m, T, s p^T_{m,k} = s p^T_{m,k+D}$,

i.e. $sp^s_{m,k} = sp^s_{m,k+D}, \forall m \notin S$

$sp^f_{m,k} = sp^f_{m,k+D}, \forall m \in S$

Proof: According to Eqs. (1),(2),(5)-(8), $\forall m,k, \{s p^T_{m,(k+iD)}\}_{i=1,2,...}$ is a monotonically increasing or decreasing sequence of number with lower bound 0 and upper bound $(l^s_m - c)$ or $(l^b_m + a_m - c)$, it will converge.

Corollary 1: In steady state

$\min \sum_T \sum_{m=1}^M \sum_{k=1}^D u^T_{m,k} \Leftrightarrow \min \sum_T \sum_{m=1}^M \sum_{k=1}^D w^T_{m,k} \Leftrightarrow \min \sum_T \sum_{m=1}^M \sum_{k=1}^D (s p^T_{m,k} - e p^T_{m,k})$
The origin mathematical formulation:

\[ \min \sum_{T} \sum_{m=1}^{M} \sum_{k=1}^{D} u_{m,k}^{T} \quad \text{Minimizing total overload} \]

s.t.  

* Eqs. (1) – (12) \( \forall m, k \)  
  - The expression of status in steady state
  
  \[ \sum_{k=1}^{D} x_{n,k} = d_{n} \quad \forall n \]  
  - All products need to be processed
  
  \[ \sum_{n=1}^{N} x_{n,k} = 1 \quad \forall k \]  
  - Products are launched one by one
  
  \[ x_{n,k} \in \{0, 1\} \quad \forall n, k \]

Note in Eqs. (1)-(12), \( \forall m, k, T, sp_{m,k}^{T} = sp_{m,\text{wrap}(k,D)}^{T} \), where \( \text{wrap}(k,D) = k + iD \), \( i \) is an integer so that \( 1 \leq k + iD \leq D \).

Eqs. (1)-(12) are nonlinear constraints, making the model hard to solve.
Mathematical formulation

Following from corollary 1, relax Eqs. (1)-(12) to inequalities, then the relaxation model is:

$$\begin{align*}
\min & \sum_{T} \sum_{m=1}^{M} \sum_{k=1}^{D} (sp_{m,k}^T - ep_{m,k}^T) & \text{Equivalent to minimizing total overload} \\
\text{s.t.} & sp_{m,k}^T \geq 0 & \forall m, k, T \\
& ep_{m,k}^T \leq l_m^T & \forall m, k, T \\
& sp_{m,k}^s \geq ep_{m,k}^{s,\text{wrap} (k-1,D)} - c & \forall m \notin S, k \\
& ep_{m,k}^s \leq sp_{m,k}^s + \sum_{n=1}^{N} t_{m,n}^s x_{n,k} & \forall m \notin S, k \\
& sp_{m,k}^f \geq ep_{m,k}^{f,\text{wrap} (k-p_m-1,D)} - (c - a_m) & \forall m \in S, k \\
& ep_{m,k}^f \leq sp_{m,k}^f + \sum_{n=1}^{N} t_{m,n}^f x_{n,k} & \forall m \in S, k \\
& sp_{m,k}^b \geq ep_{m,k}^{b,\text{wrap} (k+p_m,D)} - a_m & \forall m \in S, k \\
& ep_{m,k}^b \leq sp_{m,k}^b + \sum_{n=1}^{N} t_{m,n}^b x_{n,k} & \forall m \in S, k \\
& \sum_{k=1}^{D} x_{n,k} = d_n & \forall n \\
& \sum_{n=1}^{N} x_{n,k} = 1 & \forall k \\
x_{n,k} \in \{0, 1\} & \forall n, k
\end{align*}$$

relationships between start and end positions in steady state

All products need to be processed

Products are launched one by one
Proposition 1: the optimal solution of relaxation model is optimal to origin model.

The basic thought: A solution not feasible to origin model can’t be optimal to relaxation model. If a solution optimal to relaxation model is feasible to origin model, it must be optimal to origin model.

Relaxation model is a linear mixed integer programming (MIP) problem with \((2M+2|S|+N)D\) variables, \(ND\) integer variables and \([(4M+4|S|+N+1)D+N]\) constraints.

- MIP methods, such as branch-and-bound can be used
- Usually \(D\), the number of products in MPS, is not quite large. Commercial optimization software can be used.
  - When \(ND \leq 50\), the optimal solution can be found within 2 minutes.
- For large problem, efficient heuristic is needed.
Consider the problem in the viewpoint of time consumption

\[ \forall m \notin S, (s_p^{m,k+1} - s_p^{m,k}) + c = t_{m,\Pi(k)}^s - u_{m,k} + w_{m,k}^s \]

\[ \forall m \in S, (s_p^{m,k-p_m} - s_p^{m,k}) + a_m = t_{m,\Pi(k)}^f - u_{m,k} + w_{m,k-p_m}^b \]

\[ (s_p^{m,k+1} - s_p^{m,k-p_m}) + c - a_m = t_{m,\Pi(k-p_m)}^b - u_{m,k-p_m} + w_{m,k+1}^f \]

available time = required processing time– overload+ idle time

The relationship is intuitive and can be verified by Eqs.(1)-(12)

This thought is the base of all subsequent results.

In steady state, the available time to process consecutive D products in any station is cD.

No loss of generality, assuming \( \forall m, n, T, t_{m,n}^T \leq l_m^T \)
Properties of the MMUL sequencing problem

- Define $U_m(\Pi)$ as the overload in station $m$ in steady state if the basic sequence is $\Pi$

- **Lemma 2:** In straight station $m$, if any one of the following conditions is satisfied, an overload occurs.

  \[ \sum_{n=1}^{N} (t_{m,n}^s - c) d_n + \sum_{n=1}^{N} (2c - l_m^s - t_{m,n}^s)^+ d_n > 0 \]

  Proof: the result follows from lemma 1 and the thought
  
  available time = required processing time– overload+ idle time

- **Corollary 2:** In steady state, the lower bound of total overload in straight station $m$ is given by: $\forall \ \Pi$

  \[ U_m(\Pi) \geq [\sum_{n=1}^{N} (t_{m,n}^s - c) d_n + \sum_{n=1}^{N} (2c - l_m^s - t_{m,n}^s)^+ d_n]^+ \]
Properties of the MMUL sequencing problem

Lemma 3: In crossover station $m$, if any one of the following conditions is satisfied, an overload occurs

$$\sum_{n=1}^{N} (t_{m,n}^{f} + t_{m,n}^{b} - c)d_{n} + \sum_{n=1}^{N} (c - l_{m}^{b} - t_{m,n}^{f})^{+}d_{n} + \sum_{n=1}^{N} (c - l_{m}^{f} - t_{m,n}^{b})^{+}d_{n} > 0$$

Proof: the basic thought is the same as the proof for lemma 2

Corollary 3: In steady state, the lower bound of total overload in crossover station $m$ is given by: $\forall \Pi$

$$U_{m}(\Pi) \geq [\sum_{n=1}^{N} (t_{m,n}^{f} + t_{m,n}^{b} - c)d_{n} + \sum_{n=1}^{N} (c - l_{m}^{f} - t_{m,n}^{b})^{+}d_{n}$$

$$+ \sum_{n=1}^{N} (c - l_{m}^{b} - t_{m,n}^{f})^{+}d_{n}]^{+} \quad (13)$$

Corollary 4: In steady state, the lower bound of total overload in MMUL is given by: $\forall \Pi$

$$U(\Pi) \geq \sum_{m=1}^{M} U_{m}(\Pi)$$
Properties of the MMUL sequencing problem

Define

\[ U_m(sp, \pi) \]: the total overload to process the job sequence \( \pi \) in station \( m \) if the start position is \( sp \).

\[ W_m(sp, \pi) \]: the total idle time after processing each job in job sequence \( \pi \) in station \( m \) if the start position is \( sp \).

\[ SP_m(sp, \pi) \]: the start position when the worker in station \( m \) start the next job after finish job sequence \( \pi \) from start position \( sp \).

\[ J_m(\pi) \]: the job sequence in station \( m \) if the basic sequence is \( \pi \).

- In straight station, \( J_m(\pi) = \pi \).

- In crossover station, \( J_m(\pi) \) is related to \( p_m \). For example, \( \pi = BACD \), \( p_m = 2 \), then \( J_m(\pi) = B^fC^bA^fD^bC^fB^bD^fA^b \).
Properties of the MMUL sequencing problem

Lemma 4:

\[ U_m(0, J(\Pi)) \leq U_m(\Pi) \leq U_m(0, J(\Pi)) + SP_m(0, J(\Pi)) \]

Proof: Following from Eqs. (1)-(12), \( U_m(sp, \pi) \) is a non-decreasing function of \( sp \), while \( W_m(sp, \pi) \) is a non-increasing function of \( sp \). Assuming the start position to process \( \Pi \) in steady state is \( sp^* \). Obviously, \( sp^* \geq 0 \), hence

\[ U_m(0, J(\Pi)) \leq U_m(sp^*, J(\Pi)) = U_m(\Pi) \]

meanwhile,

\[ U_m(\Pi) = U_m(sp^*, J(\Pi)) = W_m(sp^*, J(\Pi)) + \sum_T \sum_{k=1}^D t_{m,n} d_n - cD \]

\[ \leq W_m(0, J(\Pi)) + \sum_T \sum_{k=1}^D t_{m,n} d_n - cD \]

\[ = U_m(0, J(\Pi)) + SP_m(0, J(\Pi)) - 0 \]
Properties of the MMUL sequencing problem

Given $\Pi$, $U_m(\Pi)$ can be obtained by:

procedure: calculating total overload in steady state

Input: $\Pi, t_{m,n}^T, l_m^T, c, p_m, a_m$

Output: $U_m(\Pi), ew$

begin

Initialize: \( sp_{m,1} = 0, ew = 0, md = \text{BigM}, U_m(\Pi) = 0 \)

Generate $J_m(\Pi), t_{m,j}, l_{m,j}$

for $j=1$ to $D$ do

Update $ep_{m,j}, u_{m,j}, w_{m,j}, sp_{m,j+1}$ using Eqs. (1)-(12)

$U_m(\Pi) = U_m(\Pi) + u_{m,j}$

If $md > 0$

If $w_{m,j} > 0$

$ew = ew + \min\{w_{m,j}, md\}$

$md = md - \min\{w_{m,k}, md\}$

Else

$md = \min\{md, l_{m,j} - ep_{m,j}\}$

end

$U_m(\Pi) = U_m(\Pi) + sp_{m,j} - ew$

end

$t_{m,j}$: processing time of the $j$th job in station $m$;

$l_{m,j}$: length of segment where the $j$th job is performed

$ep_{m,j}$, $(ep_{m,j}, w_{m,j}, u_{m,j})$ the start position (end position, idle time, overload) before (after, after, in) processing the $j$th job in station $m$;

$ew$ is the maximum evitable idle time
Properties of the MMUL sequencing problem

- Lemma 4 and the procedure "calculating total overload in steady state" imply some idle time in $W_m(0, J(7))$ can be "compensated" and turn into available time for processing.

- This method to calculate $ew$ in the procedure "calculating total overload in steady state" is the base of succeeding procedure to calculate evitable idle time and update lower bound.
A single-pass heuristic algorithm

- The single-pass heuristic algorithm is a constructive method.
- It tries to fill the next position of a basic sequence with the product, which results the minimum overload for sequenced jobs plus a lower bound for the remainder of the sequence.
A single-pass heuristic algorithm

Assuming $j$ products have been sequenced, let

$R=(r_1,\ldots,r_N)$: the vector of remaining product types;

$r_n$: number of products from type $n$ not sequenced;

$i$: index for jobs not sequenced

$NS_m$: set of jobs not sequenced.

$LB_m(R)$: the lower bound for overload to process jobs generated by $R$ in station $m$ in steady state

$US_m$: overload in processing subsequences of sequenced jobs

$WS_m$: idle time in processing subsequences of sequenced jobs

$ew_m$: maximum evitable idle time

$t_{m,i}$: processing time of job $i$ in station $m$

$l_{m,i}$: length of the segment in station $m$ where job $i$ is performed
A single-pass heuristic algorithm

☐ Explanation to subsequences of sequenced jobs:

- An example: In crossover station, \( p_m = 2 \)
- when \( j < \min\{\text{wrap}(p_m + 1, D), D - \text{warp}(p_m, D)\} \), the job sequence generated by sequenced products is not consecutive.
  - \( j = 1, \prod = B^{**}, J_m(\prod) = B^{f***}B^b** \), no subsequence
- when \( j \geq \max\{\text{wrap}(p_m + 1, D), D - \text{warp}(p_m, D)\} \), adjacent jobs form two subsequences
  - \( j = 2, \prod = BA^{**}, J_m(\prod) = B^fA^{f**}B^bA^b \), subsequence: \( A^bB^f \)
  - \( j = 3, \prod = BAC^*, J_m(\prod) = B^fC^bA^fC^fB^bA^b \), subsequences: \( A^bB^fC^bA^f, C^fB^b \)

☐ The criterion to select a product to fill the next position is \( LB_m(R) + US_m \)

- \( US_m \) can be calculated using Eqs. (1)-(12)
A single-pass heuristic algorithm

A lower bound for the overload in processing remaining products in straight station \(m\) is given by (14):

\[
LB_m (R) = \left[ \sum_{n=1}^{N} (t_{m,n}^s - c)r_n + (ep_{m,j}^s - c)^+ - ew_m + \sum_{n=1}^{N} (2c - l_{m,n}^s - t_{m,n}^s)^+ r_n \right]^+
\]

\[
= \left[ \sum_{n=1}^{N} (t_{m,n}^s - c)d_n - US_m + WS_m - ew_m + \sum_{i \in NS_m} (2c - l_{m,i}^s - t_{mi})^+ \right]^+
\]

Compare to the lower bound in Bolat (1997). The evitable idle time \(ew\) must be considered in our problem.

\[
LB_m (R)_{\text{Bolat}} = \left[ \sum_{n=1}^{N} (t_{m,n}^s - c)r_n + (ep_{m,j}^s - c)^+ \right]^+
\]

If \(j\) products have been sequenced, a lower bound for the overload in processing remaining products in crossover station \(m\) is given by:

\[
LB_m (R) = \left[ \sum_{n=1}^{N} (t_{m,n}^f + t_{m,n}^b - c)d_n - US_m + WS_m - ew_m + \sum_{i \in NS_m} (c - (l_{m,i}^s + l_{m,i}^b) + l_{mi} - t_{mi})^+ \right]^+
\] (15)
A single-pass heuristic algorithm

In a straight station, the evaluation function $US_m + LB_m(R)$ can obtained by:

**procedure:** $TU$ update procedure in straight station

**Input:** $R$, $\forall (k)$

**Output:** $TU$

begin
  for $m \notin S$ do
    update $ep_{m,k}$, $u_{m,k}$, $w_{m,k}$, $sp_{m,k+1}$ using Eqs. (1)-(4)
    $US_m = US_m + u_{m,k}$, $WS_m = WS_m + w_{m,k}$
    If $md > 0$
      If $w_{m,k} > 0$
        $ew_m = ew_m + \min\{w_{m,k}, md\}$
        $md = md - \min\{w_{m,k}, md\}$
      Else $md = \min\{md, l_m^s - ep_{m,k}\}$
    update $LB_m(R)$ using $R$, $US_m$, $WS_m$, $NS_m$, $ew_m$ using Eq. (14)
    update $TU_m = US_m + LB_m(R)$
  end
end
A single-pass heuristic algorithm

- The evaluation function $US_m + LB_m(R)$ in crossover station can be obtained in the same thought of $TU$ update procedure in straight station.

- Focus on the evitable idle time occurring in subsequences and assuming all the idle time in isolated jobs can be compensated.

- Differences to the $TU$ update procedure in straight station
  - If $k < \min\{\text{wrap}(p_{m+1}, D), D-\text{warp}(p_m, D)\}$, no subsequence exists. $TU$ can be calculated by the right side of inequality (13).
  - If $k \geq \max\{\text{wrap}(p_{m+1}, D), D-\text{warp}(p_m, D)\}$, there are two subsequences.
    - Placing a product will add 2 jobs to the job sequence and the length of each subsequence will increase 2, which should be considered when updating $US_m$ and $ew_m$. The amount of total evitable idle time is the sum of evitable idle time in each subsequences.

- Else, there is one subsequence.
  - Only need to pay attention to the new-added 2 jobs in the subsequence updating $US_m$ and $ew_m$. 

2009/6/1
A single-pass heuristic algorithm

Procedure of the single-pass heuristic:

procedure: single-pass heuristic

Input: $t_{m,n}^T, l_m^T, c, d_n$

Output: $\Pi$

begin

Initialize: $r_n=d_n; \Pi=\emptyset; sp_{m,k}=n; US_m=0; WS_m=0; ew_m=0; md=\text{BigM}$

for $k=1$ to $D$ do

for $n=1$ to $N$ do

// select a product type resulting in minimum evaluation function

if $r_n>0$

$\Pi(k)=n; r_n=r_n-1$

for $m=1$ to $M$ do

Update $TU_m$ using $TU$ update procedure

end

if $\text{sum}(TU_m)<\text{best}$

best=$US_m + LB_m(R); bindex=n$

end

$\Pi(k)=bindex; r_{bindex}=r_{bindex}-1$

end

end
A single-pass heuristic algorithm

As an illustration, the heuristic is applied to a modified instance in Kara et.al (2007). Cycle time = 29

<table>
<thead>
<tr>
<th>Stations</th>
<th>Product types</th>
<th>$l_m$</th>
<th>$p_m$</th>
<th>$a_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-f</td>
<td>A 13 B 6 C 21 D 21 E 22</td>
<td>49</td>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>1-b</td>
<td>A 5 B 7 C 9 D 13 E 0</td>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-f</td>
<td>A 18 B 18 C 23 D 34 E 10</td>
<td>49</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>2-b</td>
<td>A 10 B 17 C 10 D 2 E 13</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A 20 B 17 C 21 D 32 E 32</td>
<td>39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-f</td>
<td>A 11 B 10 C 2 D 26 E 17</td>
<td>37</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>4-b</td>
<td>A 7 B 11 C 12 D 15 E 2</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A 55 B 27 C 37 D 29 E 33</td>
<td>66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d_n$</td>
<td>A 2 B 1 C 1 D 3 E 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The solution given by heuristic: AABDECDDED

Overload in steady state: 83

83 is the lower bound of total overload, hence it’s a optimal solution.
This paper has dealt with sequencing a minimum product set in mixed model U-line.

The objective is to minimize work overload (unfinished work) in steady state.

An mathematical model is built and can be converted to a linear mixed integer programming problem.

Some properties of MMUL sequencing problem are identified.

A single-pass heuristic is proposed to solve large MMUL sequencing problems.