New Algorithms to Predict the Finish Production Time of Orders (FPTO) in a Furniture Production System

Samgadi Palgunadi Y
Jurusan Informatika
Universitas Sebelas Maret
Jln Ir Sutami 36 A Surakarta
palgunadi@uns.ac.id

ABSTRACT

Best practice in a furniture production system, due to the fact customers want to know more exactly when their orders will be delivered, the furniture production managements have to have a system to predict when the orders can be fulfilled. This paper will describe the algorithms to calculate the finish production time of orders (FPTO) in a furniture production system as well as their time complexities. The production line is set to be single processor- multi-steps (SPMS). Some asumptions made are infinite time services, machine availability, one production step at a time, uniform production step, priority order and priority item asumptions in production line. If the number of production steps is m and the number of orders is n, it can be shown that there exists a linear polinomial, in O(m), to calculate FPTO for single-order, and a polinomial algorithm, in O(mn), to calculate FPTO for multi-orders.

Keywords:
furniture production system, algorithm, FPTO, SPMS, polinomial algorithm

1. BACKGROUND OF STUDY

The broad goal of manufacturing operation management, a resource con-strained scheduling problem, is to achieve a coordinated efficient behaviour of manufac-turing in servicing production demands. Operation scheduling is viewed as a major issue which is a complex task requiring coordination. Shop-floor scheduling, in general, is a combinatorially complex, NP-hard problem [5]. Evaluation of operational procedures includes production scheduling where some parameters must be considered such as arrivals of orders, parts, or raw materials, processing, assembly, or inspection times, machine times to failure, machine repair times, loading/ unloading times, and setup times [1].

Order fulfillment is, in the most general sense, the complete process from point of sales inquiry to delivery of a product to the customer [4]. In its broadest definition, the possible steps in the process are order configuration, order booking, order acknowledgment/confirmation, invoicing/billing, order sourcing/planning, order changes (if needed), order processing - process step where the distribution center or warehouse is responsible to fill order [6].

Several system simulation models have been developed to assist to designing, evaluating, and managing hardwood lumber and furniture manufacturing systems [7]. In job-shop scheduling problem, production scheduling has to be performed often to ensure the optimal/near-optimal production schedule with the objective to minimize makespan. Davorin K. and Miroljub K. in [3], proposed genetic algorithms to solve job-shop scheduling problem. In general, job-shop scheduling problem can be solved polynomially, in $O(nm \log^2(nm))$ where $n$ denotes the number of jobs and $m$ denotes the number of machines [2].

Best practice in a furniture production system, due to the fact customers want to know more exactly when their orders will be delivered, the furniture production managements have to have a system to predict when the orders can be fulfilled. The problem is similar to job-shop scheduling problem or regular queuing problem. However, in some furniture orders, there are some items that shall be produced, and each item contains several units. For example a single order may contain 3 kitchen sets, 2 office sets and 4 bed sets. A kitchen set contains some units: cupboard, cabinet, table, and chairs which each unit has different style and size with that of the office sets. This paper will describe two new algorithms to predict finish production time of orders (FPTO).

2. ELEMENT SCHEMES

Some object elements have to be considered in FPTO, such as orders, items, units, men and machine capasities, production system and time. These elements has a relationship depicted as Fig.1.
production time (F), number of order of each items where the
items can be kitchen set (KS), garden set (GS), office set (OS),
and bedroom set (BS). To sake of simplicity, the number of
item in an order can be depicted as quadple \((a, b, \gamma, \delta)\) to
represent the number KS, GS, OS, and BS respectively. Every
item has some number of units and has been set to have
maximum 4 units, for example item KS has 4 unit: cupboard,
cabinet, table and chairs.

The above shema leads to construct the item structure as
Item (idOrder, idItem). For shake of simplicity we can denote
an item \((i,j)\) to say the item \(j\) of the order \(i\), so as the unit
\((i,j,k)\) means that the \(k\)-unit of item \((i,j)\).

Production rules of any furniture production system, in
general, will vary according to management. In this simulation
the system will be set according to the following asumptions:

a. Men and machines services set to be available for 24 hours
day 7 days a week (infinite time services asumption).
b. No machine will be down at any service time, i.e. there is
no alocation time to repair the machine (machine
availability asumption).
c. Each unit can only be processed only by one machine at a
time and each machine can only process one unit at a time
(one production step at a time asumption).
d. The processing steps to product a unit contains \(m\) production steps sequentially (uniform production step
assumption).

Let \(P\) be production set, i.e. \(P = \{P_1, P_2, ..., P_m\}\). If a step to
process a unit shall be skipped then the production time of
the unit in that step is set to 0. Production time for different
unit at different step may be set unequal.

At any production step \(P(k)\) \(k\) be the arrival time \(a_{\text{Time}}(k)\),
processing time \((\alpha, \beta, \gamma, \delta)\) occurs to enter to the next step respectively. Of course,
we do not need to consider delay time for the last production
step.

The FTPO is then solved by calculating the three functions
to obtain the predicted finishing order time, \(F\), of every order in
\(O\).

Observe that

a) If maximum number of units in each item is set to be
equal, say 4, then total number of units in a particular
order is \(4(\alpha+\beta+\gamma+\delta)\).
b) Processing time in \(P(l)\) of unit \(u(i,j,k)\) is \(p(t_{j,k,l})\), where
\(p = a, \beta, \gamma, \text{ or } \delta\).

Calculation predicted finishing order time, \(F(i)\), of order
\(O(i)\) can be done as follows.

a) Initial condition

Initial condition happens when the first unit of the first
order enters to the production line, where all men-machine
condition in each step are still idle. Observe that in initial
condition, following equations are obtained.

\[
\begin{align*}
\text{s(1,1,1,1) = } & \text{aTime(1) = 0} \\
\text{f(1,1,1,1) = a * t(1,1,1)} \\
\forall k, \text{m} = 1.. m - 1, \text{ d(1,1,1, k) = 0} \\
\forall m, \text{f(1,1,1,m) = } & \sum_{k=1}^{m} t(1,1,k) \\
\forall m > 1, \text{a(1,1,1,m) = f(1,1,1,m-1)} \\
\text{F(1,1,1,1) = f(1,0,0,4,m)}
\end{align*}
\]

b) At any production step \(P(k)\)

It can be shown that in production step \(k\), the following
conditions will occur

\[
\begin{align*}
\forall l, i = 1.. m - 1, \text{ d(i,j,k,l) = } & \text{f(i,j,k,l) + d(i,j,k,l)} \\
\forall i, j, k, l + 1 = & \text{f(i,j,k,l) + d(i,j,k,l)} \text{ if not}
\end{align*}
\]

3. MATHEMATICAL MODEL OF OFPT

Given a set of \(n\)-orders \(O\), with \(|O| = n\) and the production
line steps \(P\), with \(|P| = m\). In every production step \(P(l) \in P\)
there will be set a processing time \(t_{j,k,l}\) associate to produce a
unit \(u(i,j,k)\) of particular order \(O(i) \in O\).

The mathematical model of FPTO can then be develop as
follows. Let \(u(i,j,k)\) be a unit of item \((j,k)\) in order \(i\), and the unit
will be processed at step \(l = l, .., m\).
4. SINGLE ORDER CASE - $O(i)$

Any single order case - $O(i)$ can then be solved by developing the following algorithm based on the formulas described before. To sake of simplicity, some time functions must be developed. Function start, function finish and function delay will calculate the arrival time, finishing time in each production step and the delay time to enter the next step respectively.

Function start $(i,j,k,l)$ {
    If $i = j = k = l = 1$ then return 0
    else if $l = 1$ then
        return $f(i,j,k-1,1)+aTime$
    else return $f(i,j,k,l-1) + d(i,j,k,l-1)$
}

Function finish $(i,j,k,l)$ {
    if $i= j = k = 1$ then return $α * \sum t(j,k,l)$
    else {
        $x = getNumber(j)$
        return $a(i,j,k,l) + x * t(j,k,l)$
    }
}

Function delay $(i,j,k,l)$ {
    if $i = j = k = 1$ then return 0
    else if $(j = k = 1$ and $i > 1$) then
        if $(order(i).aTime > f(i-1,1,1))$ then return 0
        else return $f(i-1,1,1) - order(i).aTime$
    else return 0
}

The complete singleOrder algorithm, to solve single-case-order- $O(i)$, can then be described as follows

**Algorithm singleOrder FPTO**

Input : a single Order $i$ $(i,aTime, α, β, γ, δ)$
Output : finishing time of order, $F(i)$
// * initial step
$i = 1, j = 1, k = 1,$
for $l = 1$ to $m$ do {
    $a(i,j,k,l) = start(i,j,k,l)$
}

The complexity of the Single-Order algorithm can be stated as follows.

**Theorem 1.** Let $m = |P|$ be the number of production steps. Single order FPTO then can be solved polinomially in $O(m)$

5. MULTI-ORDER CASE

The multi-order case can be solved directly by utilizing the single-order case algorithm. Since there exists priority order and priority item assumptions in the production line, the algorithm to solve multi-case FPTO is developed based on calculating the finish time of each order independently and sequentially. However, some modifications must be made in processing the first unit of the next order, especially in calculating the arrival time and delay time of the first unit.

Due to the fact that

$\forall i > 1, s(i,1,1,1) = aTime(i) + f(1,4,4,1)$

let $p = f(i-1,1,2) - s(i,1,1,1),$  

$\forall i > 1$ and $1 < l < m.$

if $p > 0$ then $d(i,1,1,1) = p$ else 0

the function updateStart and function delay2 can be used to replace the start-function and the delay-funtion to accomodate the multi-order case. These function will then be described as follows

Function updateStart$(q)$ {
    return order(q).aTime+ $f(q-1,4,4,1)$
}

Function delay2 $(i,j,k,l)$ {
    if $i = j = k = 1$ then return 0
    else if $(j = k = 1$ and $i > 1$) then
        if $(order(i).aTime > f(i-1,1,1))$ then return 0
        else return $f(i-1,1,1) - order(i).aTime$
    else return $f(i-1,1,1) - order(i).aTime$
}

The complete multi-order algorithm is then described as follows

**Algorithm multiOrder FPTO**

Input : a set of order $O$, $|O| = n$, sorted ascending order on $aTime$
Output: array of finish time of the order $F(i)\ i=1..n$

for $q=1$ to $n$ do {
If $q = 1$ then perform singleOrder FPTO for $i = 1$
else {
    $order(q).aTime = updateStart(q)$
    for $j := 1$ to $4$ do {
        for $k := 1$ to $4$ do {
            for $l := 1$ to $m$ do {
                $s(i,j,k,l) = start(i,j,k,l)$
                $f(i,j,k,l) = finish(i,j,k,l)$
                if $l < m$ then {
                    if $j = k = 1$ then
                        $d(i,j,k,l) = delay2(i,j,k,l)$
                    else $d(i,j,k,l) = delay(i,j,k,l)$
                }
            }
        }
    }
    $F(i) = f(i,4,4,m)$
}}

The complexity of the multi-order FPTO algorithm can then be stated in the following theorem.

**Theorem 2.** Let $m = |P|$ be the number of production steps. Let $n = |O|$ be the number of orders. Multi-order FPTO then can be solved polinomially in $O(mn)$.

### 6. CONCLUSION

It can be concluded that the FPTO can be solved in polinomial time algorithm, $O(mn)$, where $m$ be the number of steps in production line, and $n$ be the number of orders. Possible future research is to develop FPTO algorithms in a multi-processor multi-step (MPMS) production line system. The slightly modification on the algorithms described in this paper might be used to solve the FPTO in MPMS.

### REFERENCES


