Study on column generation for the lot-sizing and scheduling problem with sequence-dependent setup time

Dandan Zhang¹, and Canrong Zhang²,*

¹Logistics Engineering and Simulation Laboratory, Graduate School at Shenzhen, Tsinghua University, Shenzhen, 518055 China
²Department of Industrial Engineering, Tsinghua University, Beijing, 100084 China

Abstract. The capacitated lot-sizing and scheduling problem with sequence-dependent setup time and carryover setup state is a challenge problem in the semiconductor assembly and test manufacturing. For the problem, a new mixed integer programming model is proposed, followed by exploring its relative efficiency in obtaining optimal solutions and linearly relaxed optimal solutions. On account of the sequence-dependent setup time and the carryover of setup states, a per-machine Danzig Wolfe decomposition is proposed. We then build a statistical estimation model to describe correlation between the optimal solutions and two lower bounds including the linear relaxation solutions, and the pricing sub-problem solutions of Danzig Wolfe decomposition, which gives insight on the optimal values about information regarding whether or not the setup variables in the optimal solution take the value of 1, and the information is further used in the branch and select procedure. Numerical experiments are conducted to test the performance of the algorithm.

1 Introduction

The capacitated lot-sizing and scheduling problem with sequence-dependent setup times (CLSD) is a common problem and frequently encountered in semiconductor assembly and test manufacturing (ATM) factory. In practice it is often required several hours for schedulers to generate a solution based on their experiences, which leads to the significance of research on the models and solution methods to seek effective and efficient solutions.

The CLSD problem was first proposed by Haase et al.[1]. Despite CLSD being studied intensively, carryover of setup states is not allowed in most research due to complexity. However, Prokka et al.[2] compared the models with and without setup carry-over, and verified that the one with setup carry-over can save setup cost. Besides that, as indicated in Karimi et al.[3], the CLSP with backlogging is more difficult to solve by introducing a shortage cost in the objective function. For more information, please refer to [4-6].

So far, Xiao et al.[7] is one of the few papers that constructed an MIP model to handle CLSD considering carryover setup state, machine eligibility and backlogging with multiple
In this paper, we propose a new MIP model and the per-machine Danzig Wolfe decomposition, which are equivalent to the formulation in Xiao et al.\[7\]. Related results show these two models have relative efficiency in obtaining optimal solutions and linearly relaxed optimal solutions. Besides that, we build a statistical estimation model to give insight on the optimal values, especially about information regarding whether or not the setup variables in the optimal solution take the value of 1, and the information is further used in the branch and select procedure. The remainder of this paper is organized as follows: several mathematical formulations are constructed in section 2; the analytics branch and select method is presented in section 3; numerical experiments are conducted in section 4. And the final section concludes the paper.

2 Mathematical Formulations

In this problem, planners have to determine the production quantities, product assignment and sequencing on each machine for each time period while minimizing the inventory holding cost, backlog cost and setup cost. We consider the MIP model in Xiao et al.\[7\] as baseline model(BM). The assumptions are: (1) Machine setup cost and setup time are associated with the sequence of processing, and have nothing to do with the processing time; (2) State carry-over is allowed; (3) The initial inventory and backlog quantity of all items equal to zero.

To present problem formulations, we define the following notation:

Sets and parameters:
- \( M \) set of machines, indexed by \( m \)
- \( I \) set of items, indexed by \( i \) and \( j \)
- \( T \) set of periods, indexed by \( t \)
- \( h_i \) holding cost per unit of item \( i \) in period \( t \)
- \( b_i \) backlog cost per unit of item \( i \) in period \( t \)
- \( p_{mi} \) process time per unit of item \( i \) on machine \( m \)
- \( \tau_{ij} \) sequence-dependent setup time between items \( i \) and \( j \)
- \( d_{it} \) demand of item \( i \) in period \( t \)
- \( c_{mt} \) capacity of machine \( m \) in period \( t \)
- \( e_{mi} \) 1, if machine \( m \) is eligible to process item \( i \); 0, otherwise
- \( sc \) setup cost per time period

Decision variables:
- \( Y_{mijt} \) 1, if item \( i \) immediately precedes \( j \) on machine \( m \) during period \( t \); 0, otherwise
- \( Z_{mit} \) 1, if the setup state of item \( i \) on machine \( m \) is carried over from period \( t \) to \( t+1 \); 0, otherwise

2.1 The Simplified Facility Location Reformulation(SFL)

Formulating SFL requires introducing a set of variables, \( Q_{mip} \) (the number of units of item \( i \) produced at machine \( m \) in period \( t \) to satisfy demand in period \( p \)).

The formulation is given as follows:

\[
\begin{align*}
\min & \sum_m \sum_i (\sum_{p=t+1}^{p_{mip}} h_i (p-t) Q_{mip} + \sum_{p=t}^{p_{mip}} b_i (t-p) Q_{mip} + sc \sum_p \tau_{ij} Y_{mijt}) \\
\sum_m \sum_i Q_{mip} &= d_{it}, \forall i \in I, t \in T \\
Q_{mip} &\leq d_{it} \times (\sum_t Y_{mijt} + Z_{mit \rightarrow t}), \forall m \in M, i \in I, t, p \in T \\
\sum_m p_{mi} Q_{mip} + \sum_j \tau_{ij} Y_{mijt} &\leq c_{mi}, \forall m \in M, t \in T
\end{align*}
\]


\[
\sum_i Z_{mit} \leq 1, \quad \forall m \in M, \; t \in T
\]  \hspace{1cm} (5)

\[
\sum_i Y_{mijt} + Z_{mt, t-1} = \sum_i Y_{mijt} + Z_{mit}, \quad \forall m \in M, \; i \in I, \; t \in T
\]  \hspace{1cm} (6)

\[
\sum_i Y_{mijt} \leq e_{mi}, \quad \forall m \in M, \; i \in I, \; t \in T
\]  \hspace{1cm} (7)

\[
\sum_i Y_{mijt} \leq e_{mi}, \quad \forall m \in M, \; i \in I, \; t \in T
\]  \hspace{1cm} (8)

\[
Z_{mit} \leq e_{mi}, \quad \forall m \in M, \; i \in I, \; t \in T
\]  \hspace{1cm} (9)

In the SFL model, expression (1) is the objective function to minimize the sum of inventory holding cost, backlog cost, setup cost. Equality (2) ensures the whole demand volume is produced until the end of the planning horizon. Inequality (3) corresponds to setup forcing constraints, while inequality (4) implies that the machine capacity cannot be violated. Inequality (5) makes sure that \(Z_{mit}\) carries at most one setup state from period \(t\) to period \(t + 1\). Equality (6) is the flow conservation constraint of the setup state. Inequalities (7) - (9) guarantee that the item can only be produced on the eligible machines. For conciseness, the field range constraints for each variable are omitted in the model.

### 2.2 An informative Danzig-Wolfe decomposition and column generation

On account of the sequence-dependent setup time and setup state carry-over in this paper, decomposing the primal problem into several single-item lot-sizing sub-problems is not easy due to that the value of setup state variable is decided by two adjacent items instead of only one item and the setup structure is dependent on the previous periods or the setup sequences within periods. Hence we propose a per-machine Danzig Wolfe decomposition method based on BM, which leads to a number of single-machine CLSD problems, along with a master problem with inventory holding cost and backlog cost. Specially, we let \(p \in \Omega^m\), where \(\Omega^m\) is defined as a set of setup plans for machine \(m\). That is, \(\delta_p \in \{0, 1\}, \forall p \in \Omega^m\) indicates whether to accept the setup plans for machine \(m\) or not. For a production plan \(p\) at machine \(m\), we define the following decision variables:

- \(I_{it}\)  \quad inventory of item \(i\) at the end of period \(t\)
- \(B_{it}\)  \quad backlog quantity of item \(i\) at the end of period \(t\)
- \(Q_{ipt}\)  \quad process quantity of item \(i\) during period \(t\) in the plan \(p\)
- \(Y_{ijtp}\)  \quad 1, if item \(i\) immediately precedes \(j\) in the plan \(p\) during period \(t\); 0, otherwise
- \(Z_{ijtp}\)  \quad 1, if the setup state of item \(i\) in the plan \(p\) is carried over from period \(t\) to \(t+1\); 0, otherwise

The master problem (MP) is given as follows:

\[
\min \sum_p \sum_i \left( c_i \sum_j Y_{ijtp} \delta_p + \sum_i \left( h_i I_{it} + b_i B_{it} \right) \right)
\]  \hspace{1cm} (10)

\[
I_{i, t-1} - B_{i, t-1} + \sum_p \sum_i Q_{ipt} \delta_p - I_0 + B_0 = d_{it}, \quad \forall i \in I, \; t \in T
\]  \hspace{1cm} (11)

\[
I_0 - B_0 + \sum_p \sum_i S_{ipt} \delta_p = \sum_i d_{it}, \quad \forall i \in I
\]  \hspace{1cm} (12)

\[
\sum_p Y_{ijtp} = Y_{mijt}, \quad \forall m \in M, \; i, j \in I, \; t \in T
\]  \hspace{1cm} (13)

\[
\sum_p Z_{ijtp} = Z_{mit}, \quad \forall m \in M, \; i \in I, \; t \in T
\]  \hspace{1cm} (14)

\[
\sum_p \delta_p = 1, \quad \forall m \in M
\]  \hspace{1cm} (15)

If we relax the integrality requirements of \(Y_{mijt}\) and \(Z_{mit}\), constraints (13) - (14) become redundant and can be removed from MP. The resulting model is referred to as the linear
relaxation master problem (LMP). During the column generation algorithm, the scheme starts solving a LMP with only a subset of its columns. We refer it as the linear relaxation restricted master problem (LRMP). At each iteration of column generation, it generates new profitable and limited columns to be added to the formulation, if any, by solving \( m = |M| \) pricing problems. Otherwise, the current solution of the master problem is proven to be optimal and the algorithm stops.

Let \( \pi_i, \theta_i \) and \( \mu_m \) be the dual variables of constraints (11)-(12) and (15), respectively. Given an optimal solution of the LRMP, the reduced cost \( RC_p \) of each production plan \( p \in \Omega^m \) is given by:

\[
RC_p = \sum_i \sum_t (sc \sum_j \tau_{ji} Y_{jitp} - (\theta_i + \pi_i) Q_{itp}) - \mu_m \tag{16}
\]

If the reduced cost \( RC_p \) of production plan \( p \) is negative, the corresponding column will be added to the LRMP. The pricing subproblem is formulated as follows:

\[
\min RC_p \tag{17}
\]

\[
\sum p_{mt} Q_{itp} + \sum \sum \tau_{ji} Y_{mjit} \leq c_{mt}, \forall t \in T \tag{18}
\]

\[
Q_{itp} \leq \sum d_{it} \times (\sum_j Y_{mjit} + Z_{mit-1}), \forall i \in I, t \in T \tag{19}
\]

subject to constraints (5)-(9) for each machine \( m \). Inequality (18) implies that the machine capacity cannot be violated. Inequality (19) imposes an upper bound on \( Q_{itp} \).

3 The analytics branch and select method (ABS)

Due to the NP-hardness of CLSP, we develop an analytics branch and select method with two steps. Firstly, we build a generalized linear model (GLM) to derive the likelihood information whether or not the setup variables \( Y_{mit} = \sum_j Y_{mjit} + Z_{mit-1}, \forall m \in M, i \in I, t \in T \) in the optimal solution take the value of 1 for small-size problems, secondly, using the analytics information, we generate several restricted branching and random subproblems that is easier to solve by fixing a subset of setup variables for regular-size problems.

The specific steps of generating GLM is presented below:

Step 1: Divide the relaxed 0-1 variable values yielded by the LP relaxation of SFL into 5 bins ([0.0, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8), and [0.8, 1.0]), and denote the bins of the relaxed values by \( Y_{bmit}, \forall m \in M, i \in I, t \in T \).

Step 2: Let the set of all iterations of column generation be \( L \), and let the solution values of pricing subproblems for setup variables be \( Y_{lmit}, \forall m \in M, i \in I, t \in T, \forall l \in L \).

Step 3: For each instance, calculate the average of \( Y_{opt} \) for each scenario of \( (Y^b, Y^l) \), \( \forall l \in L \), and let the average be \( Y^{ave} \). Hence we get a number of observations \( (Y^{ave}, Y^b, Y^l), \forall l \in L \).

Step 4: The generalized linear models are built at the \( l \)-th iteration of column generation, using observations of \( (Y^{ave}, Y^b, Y^l), \forall l \in L \), from all test instances. The modelling outputs are equations given as follows:

\[
\rho^l (Y) = a^l + b^l \times Y^b + c^l \times Y^l, \forall l \in L \tag{20}
\]

where \( a^l, b^l \) and \( c^l \) are coefficients of the model built at the \( l \)-th iteration of column generation. Calculate \( \rho^{ave} (Y) = (\rho^1 (Y) + ... + \rho^L (Y))/L \).

The following steps illustrates the flow of the analytics branch and select method:

Step 1: Set threshold value \( \zeta = 0.5 \), \( Y^{best} = 0 \), \( FS = \emptyset \), \( BS = \emptyset \), \( SS = \emptyset \), and rank all the setup variables in descending order of \( \rho^{ave} (Y) \).
Step 2: Select a variable $y$ whose index is not in $FS$ to generate two sub branches by fix it to 0 or 1 and insert the index into subset $BS$.

Step 3: Let $M = \{mit \mid \rho^{\text{rel}}(Y_{mit}) > \zeta \text{ and } BestY_{mit} = 1, \forall mit \in \Phi/(FS \cup BS)\}$. For each branching subproblem, we select a number of indices from $M$ by roulette wheel selection with a probability $\rho^{\text{rel}}(Y)$ and insert them into $SS$.

Step 4: After $FS$, $BS$ and $SS$ are chosen, subproblems are created. The setup variables $Y$ in $FS$ and $SS$ will be fixed to the current best incumbent solution. To jump out of local optimum, we also create random sub problems with the same number of variables being fixed, whose values are randomly selected as 0 or 1.

Step 5: Implement the relax and fix method to the subproblems in a limited time. If the objective value is less than the upper bound, update the upper bound and $Y^{\text{best}}$.

Step 6: If the subproblem with the best objective solution is one of the branching subproblems, we insert the index in $BS$ into $FS$. However, if the subproblem with the best objective solution is one of the random subproblems, we set $FS = \emptyset$.

Step 7: Terminate the algorithm either when the total solving time exceeds the time limit or all setup variables $Y$ have been fixed, otherwise go to Step 2.

4 Numerical experiments

In this section, three sets of experiments are conducted based on Xiao et al.[7], which are coded by JAVA and run on a computer with an 8GB RAM and a 2.2 GHz CPU.

4.1 Model validation

Table 1 shows the comparison of optimal solutions for BM and SFL. While BM has the smallest number of variables and constraints, their optimal values are the same, and the runtimes of the SFL formulation are much shorter than those of BM. It seems that SFL outperforms BM in terms of efficiency.

<table>
<thead>
<tr>
<th>Model</th>
<th>Variable number</th>
<th>Constraint number</th>
<th>Objective Value</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>186</td>
<td>194</td>
<td>163</td>
<td>92ms</td>
</tr>
<tr>
<td>SFL</td>
<td>276</td>
<td>311</td>
<td>163</td>
<td>76ms</td>
</tr>
</tbody>
</table>

4.2 Comparison on linearly relaxed optimal solutions

Table 2 shows the comparison results of linearly relaxed optimal solutions. It can be seen that the LP relaxation of SFL and per-machine Dantzig Wolfe decompositions are 6% and 12% higher than that of BM on average, however they consume more computation time, especially for Dantzig Wolfe decomposition.

<table>
<thead>
<tr>
<th>Objective Value</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>SFL</td>
</tr>
<tr>
<td>11791</td>
<td>12503</td>
</tr>
<tr>
<td>26ms</td>
<td>77ms</td>
</tr>
</tbody>
</table>

4.3 Comparison on linearly relaxed optimal solutions

As Machine utilisation(M), setup time(S) and the cost ratio between inventory holding and backlog(C) have significant effects on the performance of the solutions, two levels (H and L) are set for each of them, producing eight factor combinations. In order to test the
algorithm reliability, 16 instances with different problem sizes and parameter settings are generated. The results of small-size problems as shown in Table 3 indicate that in terms of average gap, the best solutions can be obtained by CPLEX followed by the ABS and RF. All the heuristic algorithms are effective and can obtain the optimal solutions or the near-optimal solutions with gaps less than 2%. From the results of large test problems in Table 4, we can see that the BS algorithm is optimized by 6.04% compared with the RF algorithm.

Table 3. The results of small-size problems.

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>m–s–c</th>
<th>ObjVal</th>
<th>Gap(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CPLEX</td>
<td>RF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1474</td>
<td>1479</td>
</tr>
<tr>
<td>LLL</td>
<td></td>
<td>1274</td>
<td>1282</td>
</tr>
<tr>
<td>LLH</td>
<td></td>
<td>2109</td>
<td>2142</td>
</tr>
<tr>
<td>LHL</td>
<td></td>
<td>1861</td>
<td>1894</td>
</tr>
<tr>
<td>LHH</td>
<td></td>
<td>3144</td>
<td>3164</td>
</tr>
<tr>
<td>HLL</td>
<td></td>
<td>2936</td>
<td>2959</td>
</tr>
<tr>
<td>HLH</td>
<td></td>
<td>3578</td>
<td>3702</td>
</tr>
<tr>
<td>HHL</td>
<td></td>
<td>3945</td>
<td>4095</td>
</tr>
<tr>
<td>HHH</td>
<td></td>
<td>2540</td>
<td>2589</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td>2540</td>
<td>2589</td>
</tr>
</tbody>
</table>

Table 4. The results of large-size problems.

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>m–s–c</th>
<th>ObjVal</th>
<th>Gap1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RF</td>
<td>ABS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1217</td>
<td>1195</td>
</tr>
<tr>
<td>LLL</td>
<td></td>
<td>921</td>
<td>902</td>
</tr>
<tr>
<td>LLH</td>
<td></td>
<td>2295</td>
<td>2075</td>
</tr>
<tr>
<td>LHL</td>
<td></td>
<td>2023</td>
<td>2075</td>
</tr>
<tr>
<td>LHH</td>
<td></td>
<td>2778</td>
<td>2726</td>
</tr>
<tr>
<td>HLL</td>
<td></td>
<td>2777</td>
<td>2694</td>
</tr>
<tr>
<td>HLH</td>
<td></td>
<td>5252</td>
<td>4840</td>
</tr>
<tr>
<td>HHL</td>
<td></td>
<td>4453</td>
<td>4119</td>
</tr>
<tr>
<td>HHH</td>
<td></td>
<td>2714</td>
<td>2550</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td>2714</td>
<td>2550</td>
</tr>
</tbody>
</table>

5 Conclusions

This paper focuses on the CLSD problem in semiconductor ATM factories, and an enhanced model SFL is proposed, which can provide a satisfied solution in shorter time, while SFL and per-machine Dantzig Wolfe decomposition can provide better lower bound. Meanwhile, this paper builds a generalized linear model to facilitate the branch and select procedure. Extensive numerical experiment results show that the average difference between the solutions obtained by the algorithm and the optimal solutions is less than 1%. And the algorithm is better than the relax and fix algorithm for all sized test problems.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 71472108, and the Shenzhen Municipal Science and Technology Innovation Committee under Grant No. JCYJ20170412171044606.

References

1. K. Haase, Oper. res, 18, 51-59 (1996)
5. G. Belvaux and L. A. Wolsey, Manage. scien, 47, 993-1007 (2001)