A branch & bound algorithm for the open-shop problem

Peter Brucker, Johann Hurink, Bernd Jurisch, Birgit Wostmann

1997 Discrete Applied Mathematics
Contents

- Introduction
- A branch & bound algorithm
  - The disjunctive graph model
  - Basic concepts of the branch & bound algorithm
  - A Branching scheme
  - Heads and tails
  - Immediate selection
  - Lower bounds
- Computational results
- Concluding remarks
Introduction

- Preview
  - Open-shop
    - $n$ jobs with $J_1, \ldots, J_n$
    - $m$ machines with $M_1, \ldots, M_m$
    - Each job $J_i$ consists of $m$ operations $O_{ij}$ ($j=1,\ldots, m$)
  - Decision point
    - Sequence jobs on each machine
    - Sequence operations(machines) of each job
Introduction

- Literature review
  - developed polynomial algorithm for $m=2$, $n=2$ and allowed preemptions case (1976)
  - Two-machine problem is solvable in polynomial time even under the consideration of one additional resource (in 1997, to appear)
  - under regular criteria, proven NP-hard (1991)
  - when $p_{ij}=1$, developed polynomial algorithm. (1993 [2], 1988[10])
A branch & bound algorithm

- Disjunctive graph model

(a) Disjunctive graph
(b) Complete selection
A branch & bound algorithm

- Basic concepts of the branch & bound algorithm
  - Every search tree node $r \rightarrow G(F_r) = (V, F_r)$
    - $F_r$: set of fixed disjunctive arcs in node $r$
  - Branching is done by dividing $Y(r)$ into disjoint subsets $Y(s_1),...,Y(s_q)$ for some $q$
    - ex) $1 \rightarrow \{2,3,4,5,...,10\}$
    - $12 \rightarrow \{3,4,5,...,10\}$
    - $123 \rightarrow \{4,5,...,10\}$
  - Set $\text{LB}(r) = \infty$ if the corresponding graph $G(F_r)$ contains a cycle
  - UB is updated when a new feasible solution is found which improves UB
A branch & bound algorithm

- A branching scheme
  - $P$: critical path in the graph $G(S)$
  - $S$: Complete selection
  - $L(S)$ be the length of $P$
  - block on $P$ in $G(S)$
    - all $u_i$ are either processed on the same machine or belong to the same job
    - extending the sequence from either side results in the violation of (a)
A branch & bound algorithm

- A branching scheme

- Theorem 1.

  Let $S$ be a complete selection corresponding to some solution of the open-shop problem and let $P$ be a critical path in $G(S)$. If there exists another complete selection $S'$ such that $L(S') < L(S)$, then there is a block $u_1, ..., u_l$ on $P$ and an operation $u_i$ in it such that either $u_i$ is before $u_1$ in $S'$ or $u_i$ is after $u_l$ in $S'$. 

\[ L(S) \quad \rightarrow \quad L(S') \]
A branch & bound algorithm

A branching scheme

- Concept
  - Assume that, we have complete selection “S”
  - Calculate block; Each block has 2 candidate operations
  - Checking the theorem
    - after change the sequence in a block

(1) If (1) is better then any other, fix the disjunctive arcs
    - after-candidate, also checking
    - Choose the first checking block with maximal cardinality

(2)
(3)
(4)
(5)
A branch & bound algorithm

Procedure Branch & Bound (r)

BEGIN
  Calculate a solution $y$ corresponding to a selection $S \in Y(r)$ using heuristics;
  IF $C_{\text{max}}(S) < UB$ THEN UB := $C_{\text{max}}(S)$;
  Calculate a critical path $P$;
  Calculate the blocks of $P$;
  Calculate the sets $E^B_j$ and $E^A_j$;
  FOR ALL operations $i \in E^\mu_j$ with $j = 1, \ldots, k$ and $\mu = A, B$ DO
    Fix disjunctions for the corresponding successor $s$;
    Calculate a lower LB($s$) for node $s$;
    IF LB($s$) $< UB$ THEN Branch & Bound ($s$)
  END
END
A branch & bound algorithm

- Heads and tails
  - Concept
    - head
    - tail

- Calculations
  - Head
    \[ r_i = \max_{J \subseteq Q, \; R \subseteq R_i} \left\{ \min_{J \subseteq J_i} r + \sum_{j \in J} p_j \right\}, \]
  - Tail
    \[ q_i = \max_{J \subseteq Q_i', \; R \subseteq R_i'} \left\{ \sum_{J \subseteq J_i} p_j + \min_{J \subseteq J_i} q_j \right\}. \]

J(Q: job, R: machine)
A branch & bound algorithm

- Lower bounds
  - using Head/Tail
    - tail
      $$r_i + p_i + \max \left\{ \max_{j \in B_k \setminus \{i\}} (p_j + q_j); \sum_{j \in B_i \setminus \{i\}} p_j + \min_{j \in B_t \setminus \{i\}} q_j \right\}$$

  - head
      $$\left\{ \max_{j \in B_i \setminus \{i\}} (r_j + p_j); \min_{j \in B_i \setminus \{i\}} r_j + \sum_{j \in B_i \setminus \{i\}} p_j \right\} + p_i + q_i$$
A branch & bound algorithm

- Calculation of heuristic solutions
  - for complete selection
- Considering Point
  - is it optimal?
    - under makespan measure, the only considering point is “Critical path”
    - characteristic of the openshop (Unclear)

- Heuristic with matching algorithm
  - matching with Job, Machine, Operation (Job-Machine matching), unconflict,
  - apply priority rule

- sum-matching/minimization:
  \[
  \min \left\{ \sum_{(U,M) \in A} p_{ij} \mid A \text{ is a matching of maximal cardinality} \right\}
  \]

- sum-matching/maximization:
  \[
  \max \left\{ \sum_{(U,M) \in A} p_{ij} \mid A \text{ is a matching of maximal cardinality} \right\}
  \]

- bottleneck-matching/minimization:
  \[
  \min \left\{ \max_{(U,M) \in A} p_{ij} \mid A \text{ is a matching of maximal cardinality} \right\}
  \]

- bottleneck-matching/maximization:
  \[
  \max \left\{ \min_{(U,M) \in A} p_{ij} \mid A \text{ is a matching of maximal cardinality} \right\}
  \]

- modified bottleneck-matching/minimization:
  \[
  \min \left\{ \max_{(U,M) \in A} \{r_{ij} + p_{ij}\} \mid A \text{ is a matching of maximal cardinality} \right\}
  \]

- modified bottleneck-matching/maximization:
  \[
  \max \left\{ \min_{(U,M) \in A} \{r_{ij} + p_{ij}\} \mid A \text{ is a matching of maximal cardinality} \right\}
  \]
Computational results

- **Instances**
  - **Benchmark problems**
    - B&B1 : sum-matching/minimization
    - B&B2 : sum-matching/maximation
    - B&B3 : Modified bottleneck-matching/minimization
  - **Table 1 : Optimal solution(time)**
  - **Table 2. : Number of search tree nodes/CPU time**

<table>
<thead>
<tr>
<th>(n,m)</th>
<th>Nodes</th>
<th>B &amp; B₁</th>
<th>B &amp; B₂</th>
<th>B &amp; B₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4,4)</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>(5,5)</td>
<td>542</td>
<td>471</td>
<td>626</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15.2</td>
<td>11.4</td>
<td>15.3</td>
<td></td>
</tr>
<tr>
<td>(7,7)</td>
<td>17353</td>
<td>10988</td>
<td>7340</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1374.0</td>
<td>824.6</td>
<td>565.9</td>
<td></td>
</tr>
<tr>
<td>(8,8)</td>
<td>8284</td>
<td>1380</td>
<td>353</td>
<td></td>
</tr>
<tr>
<td></td>
<td>854.0</td>
<td>149.4</td>
<td>29.2</td>
<td></td>
</tr>
<tr>
<td>(9,9)</td>
<td>172832</td>
<td>12654</td>
<td>139893</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32016.1</td>
<td>1870.3</td>
<td>23301.0</td>
<td></td>
</tr>
<tr>
<td>(10,10)</td>
<td>101995</td>
<td>315869</td>
<td>274347</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25259.3</td>
<td>75249.1</td>
<td>64665.1</td>
<td></td>
</tr>
</tbody>
</table>
Computational results

- Results
  - Table 3: compare with Tailied, Brasel
  - Table 4: Hardness of instances
    - LB: Trivial Lower bound (Maximum Sum of processing times)
    - MIN: (Minimum sum of processing times)
    - DIFF: MIN/LB
    - WORKLOAD: total processing time / (m * LB)
  - Table 5: different solution methods
    - vs timilimet
      - UBneu: solution found by tabu search
      - CPUneu: CPU time for the tabu search
Concluding remarks

- Problem
  - Openshop problem (m x n)
- Branch & Bound algorithm
  - Solution with Matching problem
- Comparing
  - with literatures
- ADV/DISADV