ON SMT SOLVERS AND JOB SHOP PROBLEMS
MODELS COMPARISON AND PERFORMANCE EVALUATION

Sabino Roselli
- Adaptation of classical model formulations for the Job-Shop Problem to the SMT language;

- Comparison of SMT model formulations over benchmark instances;

- Comparison of State-of-the-Art MILP and SMT solvers over benchmark instances.

- Novel formulation employing bit-vector theory
THE CLASSICAL JOB SHOP PROBLEM

- $J = \{j_i\}_{i=1}^n$
- $M = \{m_i\}_{i=1}^k$
- $o_i^j$
- $d_i^j$
- $(o_1^j, ..., o_i^j, ..., o_k^j)$

Jobs
Machines
Operation
Duration
Sequence
SAT AND SMT

• Express the Problem through Boolean variables in Conjunctive Normal Form

\[(p \lor \neg r \lor p) \land (\neg q \lor \neg r \lor p)\]

• Implement Decision Procedures over Theories to extend the original problem

\[\text{Not}(p \text{ Implies } q) \text{ Or } (r \text{ Implies } p)\]
\[(p \text{ And } r) \text{ Implies } (i + 3 \leq 0)\]
STANDARD MODELS

Disjunctive: operations defined by their starting time

$s_{mj}$: Integer variable and models the start time of job $j$ on machine $m$

Time-Index: Execution time is split into steps

$s_{mjt}$: Boolean variable that is true if job $j$ starts on machine $m$ at time $t$

Rank-Based: focus on the machine side

$x_{mjt}$: Boolean variable that is true if job $j$ starts on machine $m$ at the $t$-th position

$s_{mt}$: Integer variable representing the starting time of position $t$ of machine $m$
DISJUNCTIVE

\begin{align*}
\text{minimize } & \quad T_{\text{max}} \text{ subject to} \\
& \quad s_{mj} \geq 0 \quad \forall j \in J, m \in M \quad (1) \\
& \quad T_{\text{max}} \geq s_{\delta_k,j} + d_{\delta_k,j} \quad \forall j \in J \quad (2) \\
& \quad s_{\delta_i,j} \geq s_{\delta_{i-1},j} + d_{\delta_{i-1},j} \quad \forall j \in J, i = 2, \ldots, k \quad (3) \\
& \quad s_{mj} \geq s_{mj'} + d_{mj'} \vee s_{mj'} \geq s_{mj} + d_{mj} \\
& \quad j, j' \in J, j \leq j', m \in M \quad (4)
\end{align*}

1: restricts variables domain to be larger than or equal to zero;

2: impose the objective function to be larger than or equal to the start time of the last operation of each job plus its duration;

3: precedence constraints among the operations belonging to the same job;

4: two operations sharing the same resource cannot take place at the same time.
5-6: only one operation is executed on a machine per each time step;

7: each machine executes only one operation at a time;

8: sets the objective function as larger than operations completion times;

9: precedence constraint among operations of the same job.
RANK-BASED

minimize $T_{\text{max}}$ subject to

10: the start variable domain is restricted to the natural numbers;

11: objective function;

12: precedence constraint among the operations executed on a machine;

13: operations precedence within the same job;

14: each job can be assigned only once to a certain machine;

15: a position can be assigned only to one job.
NOVEL MODEL

Time Index (bit-vector)

$s_{mj}$ is a bit-vector variable of size $H$ that can only have one bit set. The position of such bit defines the step at which job $j$ starts on machine $m$.

$\bar{s}_{mj}$ is a bit-vector variable of size $H$ that has as many bits set as time units the job $j$ takes to be executed on machine $m$. The most significant bit in the trail corresponds to the bit set on the variable $s_{mj}$.

$d_{mj}$ is a bit-vector constant of size $H$ that has the least significant bits set based on the duration of job $j$ on machine $m$. 
MODEL VARIABLES

\[ S_{mj} 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \text{ (before assignment)} \]

\[ T = 16 \]

\[ S_{mj} 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 \text{ (after assignment)} \]

Instance: 3 x 3

\[ \bar{s}_{mj} 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 \text{ (after assignment)} \]

Duration mj: 3

\[ \bar{S}_{mj} 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 \text{ (after assignment)} \]

\[ \bar{d}_{mj} 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 \text{ (constant vector)} \]
maximize $T_{\text{max}}$ subject to

16-17: only one bit is set

$$s_{mj} \neq 0$$  \quad \forall j \in J, m \in M \quad (16)$$

$$s_{mj} \land (s_{mj} - 1) = 0$$  \quad \forall j \in J, m \in M \quad (17)$$

$$\bar{s}_{mj} = \bigvee_{i=0}^{d_{mj}} (s_{mj} >> i)$$  \quad \forall j \in J, m \in M \quad (18)$$

18-19: defining $\bar{s}_{mj}$ and $s_{mj}$

$$s_{mj} = \neg(\bar{s}_{mj} \lor (-\bar{s}_{mj}))$$  \quad \forall j \in J, m \in M \quad (19)$$

20: latest start for each operation

$$s_{jm} \land \bar{d}_{jm} = 0$$  \quad \forall j \in J, m \in M \quad (20)$$

21: precedence constraint

$$s_{m,o_i^j} \land s_{m,o_i^j} = 0$$  \quad \forall j \in J, m \in M, i = 2 \ldots k \quad (21)$$

22: most significant bit must be 0

$$s_{mj} \land (1 << H - 1) = 0$$  \quad \forall j \in J, m \in M \quad (22)$$

23: objective function set up

$$T_{\text{max}} = \bigvee s_{mo_i^j}$$  \quad \forall j \in J, m \in M \quad (23)$$

24: non overlapping constraint

$$\bigwedge_{j,j' \in J, j \neq j'} (\bar{s}_{mj} \land \bar{s}_{mj'})$$  \quad \forall m \in M \quad (24)$$
EXPERIMENTAL RESULTS

- Generated Instances up to size 9 x 9
- Benchmark instances up to size 15 x 15
## THEORIES

### GENERAL ALGORITHMS
- DPLL-based Simplex
- .....  

### ALGORITHMS FOR IDL
- Bellman-Ford Algorithm
- Floyd-Warshall

<table>
<thead>
<tr>
<th></th>
<th>BIT VECTOR</th>
<th>IDL</th>
<th>LIA</th>
<th>SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TI</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RB</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TI(BV)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SOLVERS COMPARISON
CONCLUSIONS

SMT solvers proved an efficient method to solve JSP (both Standard and Flexible)

Among the most common models, the Disjunctive is the one that showed the best performance

Are these the only best model formulations?