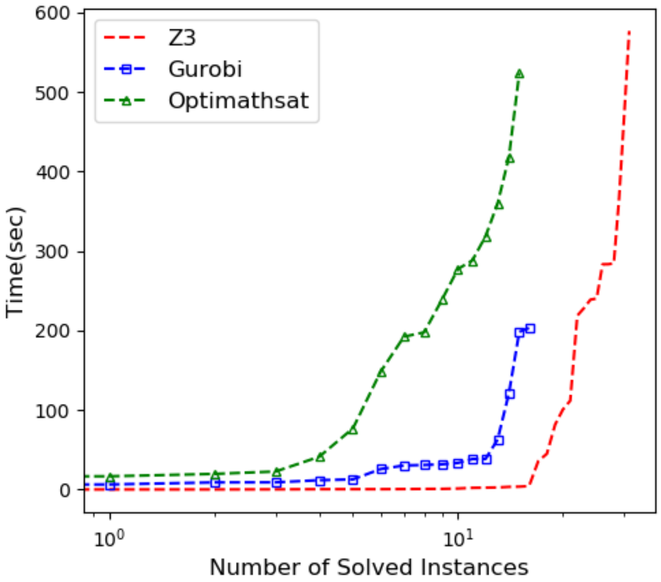
CHALMERS

ON SMT SOLVERS AND JOB SHOP PROBLEMS

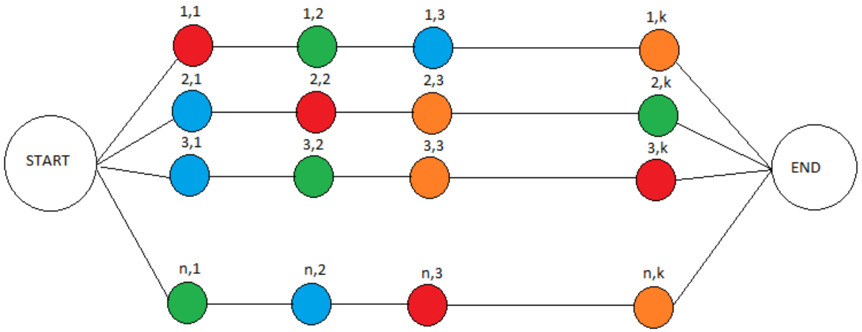
MODELS COMPARISON AND PERFORMANCE EVALUATION

Sabino Roselli

- Adaptation of classical model formulations for the Job-Shop Problem to the SMT language;
- I Comparison of SMT model formulations over benchmark instances;
- I Comparison of State-of-the-Art MILP and SMT solvers over benchmark instances.
- Novel formulation employing bit-vector theory



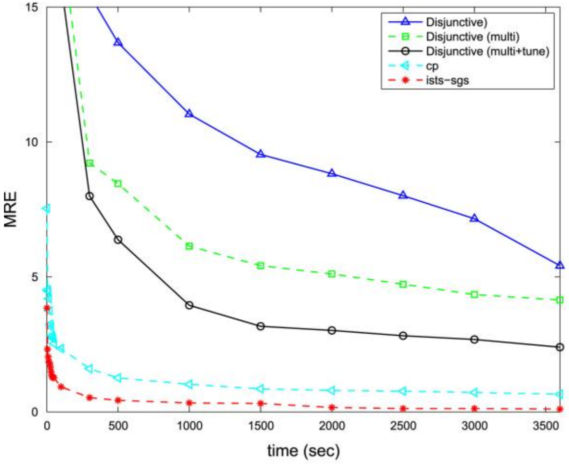
THE CLASSICAL JOB SHOP PROBLEM



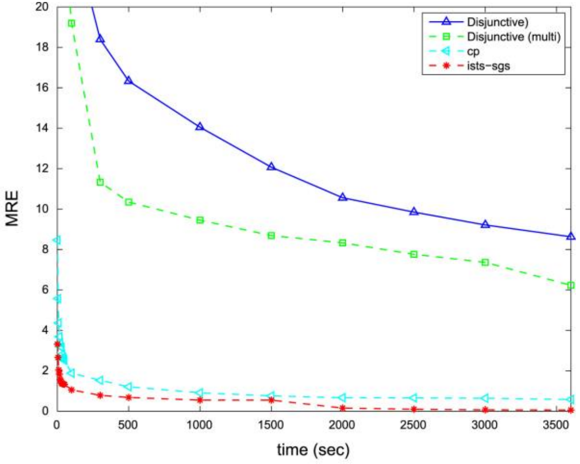
• $J = \{j_i\}_{i=1}^{n}$	Jobs
• $M = \{m_i\}_{i=1}^{k}$	Machines
• o_i^j	Operation
• d_i^j	Durartion
• $(o_1^j, \dots, o_i^j, \dots, o_k^j)$	Sequence

OPTIMIZATION

	DISJUNCTIVE	TIME-INDEX	RANK-BASED
Gurobi CPLEX SCIP	X X X	X X X	X X X
ILOG	X		



(a) 20 × 20 problems



(b) ta11 - ta20

W.-Y. Ku and J. C. Beck, ”Mixed integer programming models for job shop scheduling: A computational analysis ” Computers & Operations Research, vol. 73, pp. 165-173, 2016.

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

Not(p Implies q) Or (r Implies p)

(p And r) Implies (i + 3 ≤ 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

minimize T_{max} subject to

$$s_{mj} \geq 0 \qquad \forall j \in J, m \in M \qquad (1)$$

$$T_{max} \geq s_{\sigma_k^j, j} + d_{\sigma_k^j, j} \qquad \forall j \in J \qquad (2)$$

$$s_{\sigma_i^j, j} \geq s_{\sigma_{i-1}^j, j} + d_{\sigma_{i-1}^j, j} \qquad \forall j \in J, i = 2, \ldots, k \qquad (3)$$

$$s_{mj} \geq s_{mj'} + d_{mj'} \vee s_{mj'} \geq s_{mj} + d_{mj}$$
$$j, j' \in J, j \leq j', m \in M \qquad (4)$$

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

2: impose the objective function to be lager 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.

TIME-INDEX

minimize T_{max} subject to

$$\bigvee_{t=0}^{H} s_{mjt} \qquad \forall m \in M, j \in J \tag{5}$$

$$s_{mjt} \to \bigwedge_{t'=0}^{t-1} \neg s_{mjt'} \wedge \bigwedge_{t'=t+1}^{H} \neg s_{mjt'} \qquad \forall t = 1, \ldots, H, m \in M, j \in J \tag{6}$$

$$s_{mjt} \to \bigwedge_{t'=t}^{t+p_{mj}} \neg s_{mj't'} \qquad \forall j, j' \in J, j \leq j', t = 1, \ldots, H, m \in M \tag{7}$$

$$s_{mjt} \to T_{max} \geq t + d_{mj} \qquad \forall m \in M, j \in J, t = 1, \ldots, H \tag{8}$$

$$x_{\sigma^j_{i-1}jt} \to \bigwedge_{t'=0}^{t+d^j_{i-1}} \neg x_{\sigma^j_{i-1}jt'} \qquad \forall i = 2, \ldots, k, t = 1, \ldots, H, j \in J \tag{9}$$

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_{max} subject to

$$s_{mt} \geq 0 \qquad \forall m \in M, t = 1 \dots k \tag{10}$$

$$x_{\sigma_k^j jt} \rightarrow T_{max} \geq s_{\sigma_k^j t} + d_{\sigma_k^j j} \qquad \forall j \in J, t = 1 \dots k \tag{11}$$

$$x_{mjt} \rightarrow s_{mt} + d_{mj} \leq s_{m,t+1} \qquad \forall t = 1 \dots k, m \in M, j \in J \tag{12}$$

$$(x_{\sigma_{i-1}^j jt_1} \wedge x_{\sigma_i^j jt_2}) \rightarrow s_{\sigma_{i-1}^j t_1} + d_{\sigma_{i-1}^j j} \leq s_{\sigma_i^j t_2} \qquad \forall t_1, t_2 = 1 \dots k, i = 2 \dots k, j \in J \tag{13}$$

$$\left(x_{mjt} \rightarrow \bigwedge_{t'=0}^{t-1} \neg x_{mjt'} \wedge \bigwedge_{t'=t+1}^{k} \neg x_{mjt'}\right) \wedge \bigvee_{k=0}^{k} x_{mjt} \qquad \forall t = 1 \dots k, m \in M, j \in J \tag{14}$$

$$\left(x_{mjt} \rightarrow \bigwedge_{j'=0}^{j-1} \neg x_{mj't} \wedge \bigwedge_{j'=j+1}^{n} \neg x_{mj't}\right) \wedge \bigvee_{t=0}^{k} x_{mj't} \qquad \forall t = 1 \dots k, m \in M, j \in J \tag{15}$$

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}

$\underline{s}_{mj}$ is a bit-vector variable of size H that has all bit sets from the time-step the operation is completed until the last position on the vector

$\bar{d}_{mj}$ 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

T = 16

Instance: 3 x 3

Duration mj: 3

S_{mj} 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (before assignment)

S_{mj} 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (after assignment)

$\bar{s}_{mj}$ 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 (after assignment)

$\underline{S}_{mj}$ 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 (after assignment)

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

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

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

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

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

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

$$\underline{s}_{m,o^j_{i-1}} \wedge s_{m,o^j_i} = 0 \qquad \forall j \in J, m \in M, i = 2 \dots k \quad (21)$$

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

$$T_{max} = \bigvee \underline{s}_{mo^j_k} \qquad \forall j \in J, m \in M \quad (23)$$

$$\bigwedge_{\substack{j,j' \in J \\ j \neq j'}} (\bar{s}_{mj} \wedge \bar{s}_{mj'}) \qquad \forall m \in M \quad (24)$$

16-17: only one bit is set

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

20: latest start for each operation

21: precedence constraint

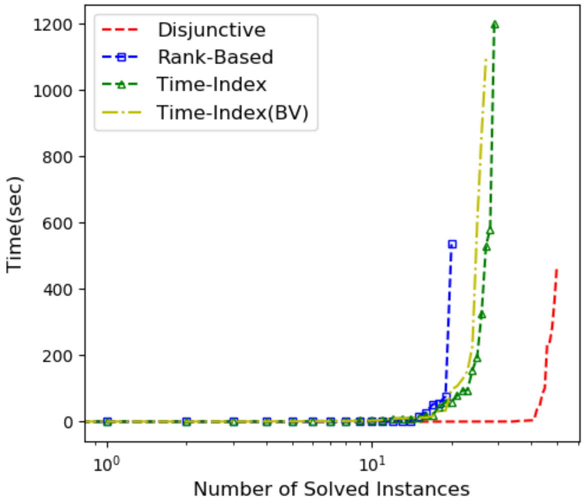
22: most significant bit must be 0

23: objective function set up

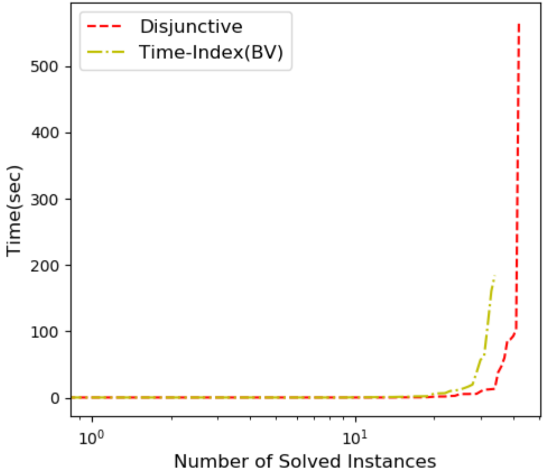
24: non overlapping constraint

EXPERIMENTAL RESULTS

Z3



OptiMathSAT



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

THEORIES

	BIT VECTOR	IDL	LIA	SAT
DS		X		
TI			X	
RB			X	X
TI(BV)	X			

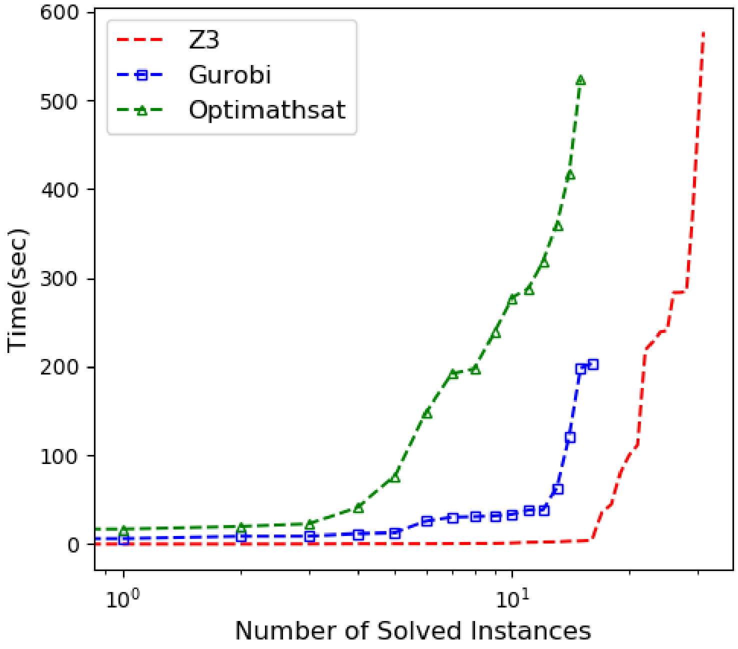
GENERAL ALGORITHMS

- DPLL-based Simplex
- …..

ALGORITHMS FOR IDL

- Bellman-Ford Algorithm
- Floyd-Warshall

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?

CHALMERS