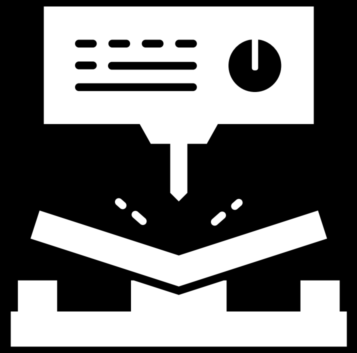
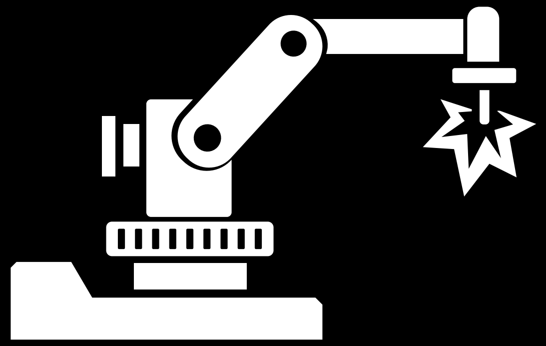
TRUMPF's Problem – Overview

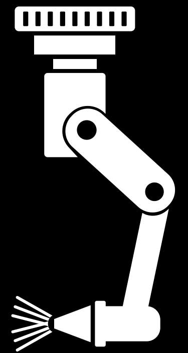
Base Example:



Bending – M_1, M_2, M_3



Welding - M_4, M_5



Painting – M_6

j Jobs containing o Operations on m Machines at Time t with Due Date d

→ J_1:(O_1, T_1, D_1), J_2:(O_2, T_2, D_2), J_3:(O_3, T_3, D_3)… and O_1:(M_1, M_2, M_3), O_2:(M_4, M_5), O_3:(M_6) → $X_{o,m,t}$

Different *additional* advancements for the Costfunction

Primary Goal:

Minimize the Total Number of Minutes over the Due Date

→Higher Costs with every Time Step further over the Due Date

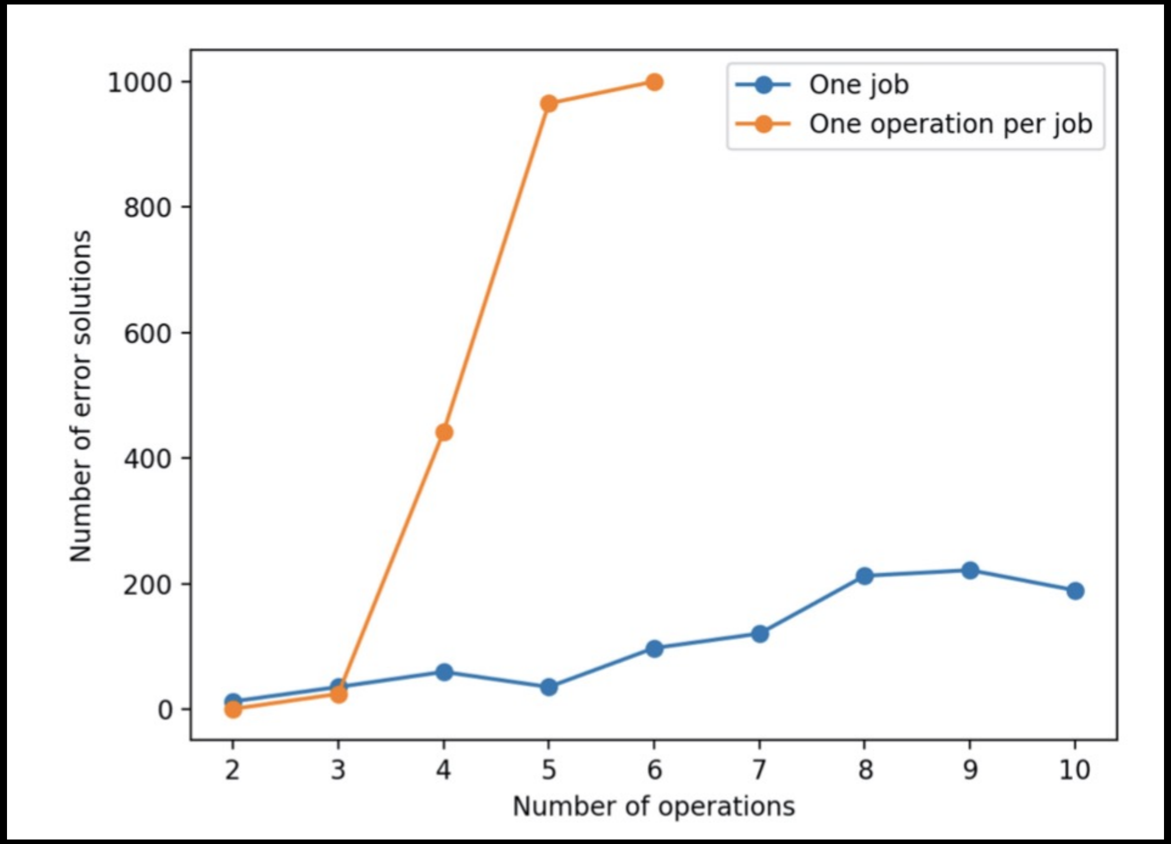
Additional Advancements:

Type	Meaning	Advantages
JIT	Just in Time Production	• Lower cost of storage • In case of human interaction → time buffer for possible delays/errors
ASAP	Shortest Make span possible	• possibility to test boundaries of production capability and eventually increase capacity

Complexity of *Flexible* Job Shop Scheduling

Additional Complexity compared to Standard JSSP:

- Different number of machines per operation (3, 2, 1)
- Not every operation has to be included (e.g., only bending & painting)
    - *Note: Errors for n Operations: with more Operations in less Jobs <<< less Operations in more Jobs*
    - Graph by *K. Kurowski et al.*



- Time for every Job and Operation different: 20sec, 30sec,…8min → multiples of 10 sec → ≈2000 Time Steps in 340min (Deadlines)
- Complexity: $\mathrm{O} * \mathrm{M} * \mathrm{T} = 3 * 6 * 2000 = 36.000$

QAR QUANTUM APPLICATIONS & RESEARCH LABORATORY

Variable Pruning for Complexity Reduction

$$Complexity: \; O \cdot M \cdot T$$

$x_{O,M,T}$

O	M	T
Bending	Bending-Machine 1	1
Welding	Bending-Machine 2	2
Painting	Bending-Machine 3	…
	Welding-Machine 1	…
	Welding-Machine 2	…
	Painting Machine 1	…
		…
		2000

Pruning Strategies: Combinatorial > Logical > Time modulo

QAR QUANTUM APPLICATIONS & RESEARCH LABORATORY

Combinatorial Pruning

$$Complexity\colon\ M \cdot T$$

- delete all Values whose Index cannot exist. E.g.:
- $\mathrm{X}_{\mathrm{O,\,M,\,T}} := \mathrm{X}_{2,\,1,\,5}$ → M_1 (bending machine) is not in mapping Dictionary of O_2 (welding)

$X_{O,M,T}$	$X_{1,1,1}$	$X_{1,1,2}$	$X_{1,1,3}$	…	$X_{1,2,1}$	$X_{1,2,2}$	$X_{1,2,3}$	…	$X_{2,1,1}$	$X_{2,1,2}$	$X_{2,1,3}$	…
$X_{1,1,1}$	0	0	0	0	0	0	0	0	0	0	0	0
$X_{1,1,2}$		0	0	0	0	0	0	0	0	0	0	0
$X_{1,1,3}$			0	0	0	0	0	0	0	0	0	0
…				0	0	0	0	0	0	0	0	0
$X_{1,2,1}$					0	0	0	0	0	0	0	0
$X_{1,2,2}$						0	0	0	0	0	0	0
$X_{1,2,3}$							0	0	0	0	0	0
…								0	0	0	0	0
$X_{2,1,1}$									0	0	0	0
$X_{2,1,2}$										0	0	0
$X_{2,1,3}$											0	0
…												0

>

$X_{O,M,T}$	$X_{1,1,1}$	$X_{1,1,2}$	$X_{1,1,3}$	…	$X_{1,2,1}$	$X_{1,2,2}$	$X_{1,2,3}$	…
$X_{1,1,1}$	0	0	0	0	0	0	0	0
$X_{1,1,2}$		0	0	0	0	0	0	0
$X_{1,1,3}$			0	0	0	0	0	0
…				0	0	0	0	0
$X_{1,2,1}$					0	0	0	0
$X_{1,2,2}$						0	0	0
$X_{1,2,3}$							0	0
…								0

QUANTUM APPLICATIONS & RESEARCH LABORATORY

Logical Pruning for Heads and Tails

- For any operation needs to be time for predecessor and successor operations, so we can remove illogical start times, e.g., Painting at $t=0$, if we also need to bend & weld

- We removed all variables $0 \leq x_{ij} < S$, where S is the sum of execution times of all operations prior the considered one. Then, we also removed all variables $T - S \leq x_{ij} < T$, where S is the sum of execution times of all operations after the considered one. *(Kurowski et al)*

$$Complexity\text{:}\\ M[T - M\langle p\rangle + 1]$$

$where\ \langle p\rangle\ represents\ the\ average\ execution\ Time\ of\ the\ operations$

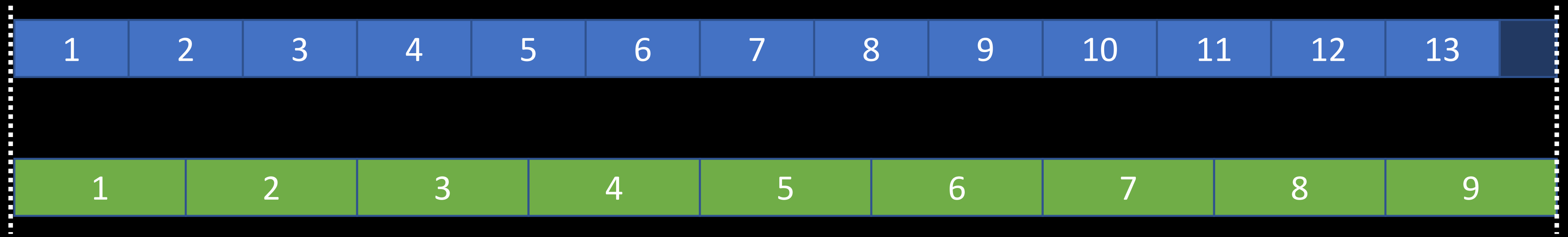
-- (Venturelli et al)

QAR QUANTUM APPLICATIONS & RESEARCH LABORATORY

TRUMPF

Time Modulo Pruning

- Each operation can vary in time on each machine for every job
- Bending = mod 20sec | Welding = mod 30sec | Painting = mod 60sec | = mod 10sec
- Example A: Bending 120sec | Welding 330sec | Painting 420sec | = mod 30sec
- Example B: Bending 260sec | Welding 270sec | Painting 0sec | = mod 10sec



13 Time Steps at 20 sec bending

9 Time Steps at 30 sec welding

- Approach: Rounding **up** to common denominator with small margin of error (10sec), which brings us down from ≈ 2000 Timesteps to ≈ 670 (with a margin of error of 40sec down to 335 at mod 60sec)
- Added Value from real-world business perspective: small additional time buffers

Phase Separation

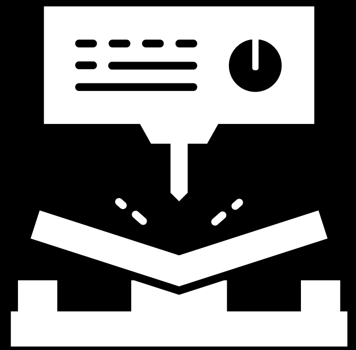
TRUMPF

Sum of all logical Tuples

$x_{O,M,T}$	$x_{1,1,1}$	$x_{1,1,2}$	$x_{1,1,3}$	…	$x_{1,2,1}$	$x_{1,2,2}$	$x_{1,2,3}$	…
$x_{1,1,1}$	0	0	0	0	0	0	0	0
$x_{1,1,2}$		0	0	0	0	0	0	0
$x_{1,1,3}$			0	0	0	0	0	0
…				0	0	0	0	0
$x_{1,2,1}$					0	0	0	0
$x_{1,2,2}$						0	0	0
$x_{1,2,3}$							0	0
…								0

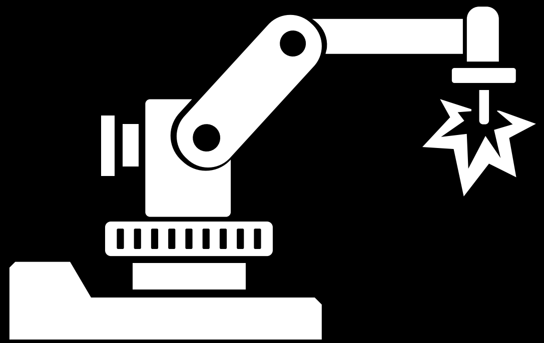
QUBO

Tuples regarding bending only



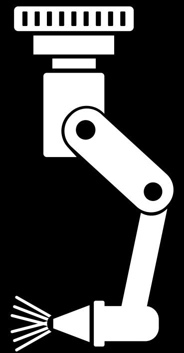
QUBO

Tuples regarding welding only with start time = end time bending



QUBO

Tuples regarding painting only with start time = end time welding



+ Tuples without bending, but have welding with start time = 0

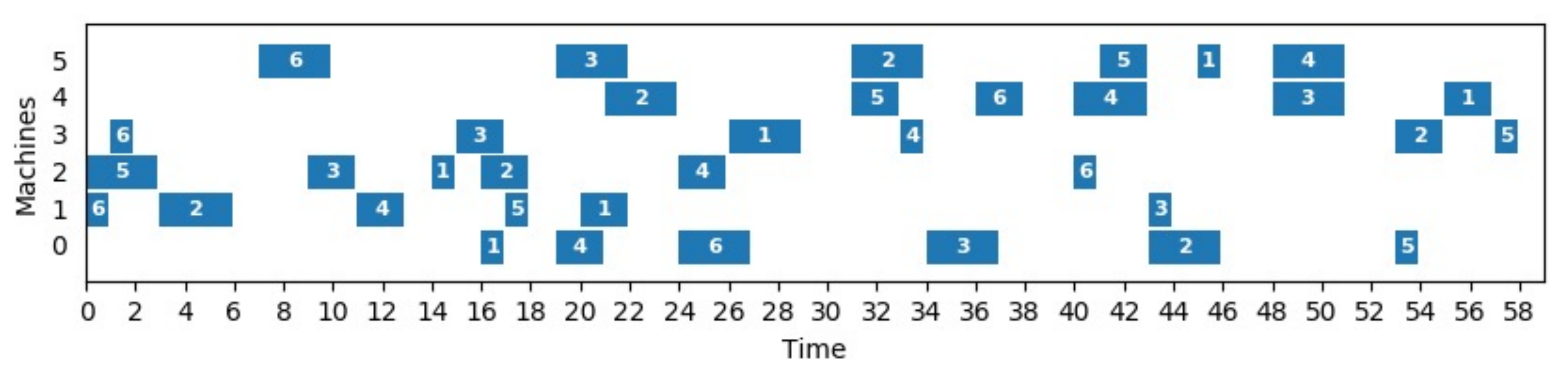
+ Tuples without bending/welding, but have painting with start time = 0

«The idea […] is that good solutions for smaller subproblems with less jobs and less operations might not necessarily be part of a good solution for the original problem, following operations can drastically change the picture. However, often in practice 'early' operations in a good solution of the smaller subproblem tend to stay rigid when compared to good solutions of the original problem.» – Denkena et al

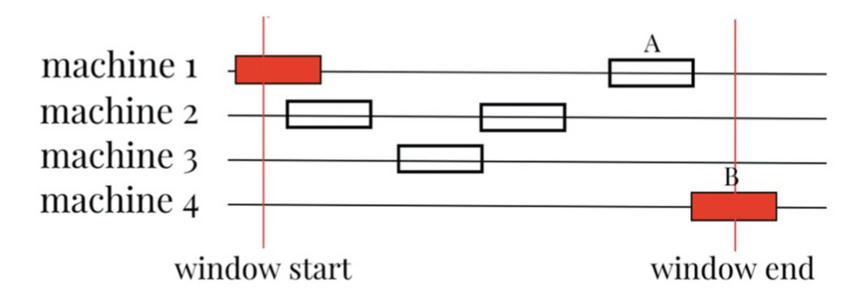
QAR QUANTUM APPLICATIONS & RESEARCH LABORATORY

Heuristic approach: Rolling Time Windows

- « The main idea behind the heuristic is to define a processing window and move it in time till the end of a schedule, so only a limited number of operations is considered. In other words, we iterate the processing window in time, and check all the operations if they fit into one of three categories, where: » – K. Kurowski et al
    - w_{begin} : start time of window
    - w_{end} : end time of window
    - s_i : start time of operation i
    - p_i : execution time of operation i



- Operations reaching out of the processing window (from the left or the right side), will be assigned additional parameters



Bottleneck Identification as Prioritization for Phase Separation and Rolling Time Window

- «In larger problems with many jobs, it is not practical to include all jobs in each iteration, so a choice about which jobs to exclude from the subproblem needs to be made each time. The idea in this approach is to assign a bottleneck factor $b_j^{(k)} > 0$ to each unfinished job *j. […]* These weight factors should reflect how much the job contributes to the overall make span of the problem.» – Denkena et al
- In our case the length of each operation in each job defines the bottleneck
- Also, the difference between make span and Due Date is decisive
- Easier alternative to bottleneck factor $b_j^{(k)}$: *sort by "length per operation"*
  and "Δ Due Date – Total Time"

TRUMPF

QAR QUANTUM APPLICATIONS & RESEARCH LABORATORY