A Branch-and-Bound approach for stochastic 2-machine flow shop scheduling with rework

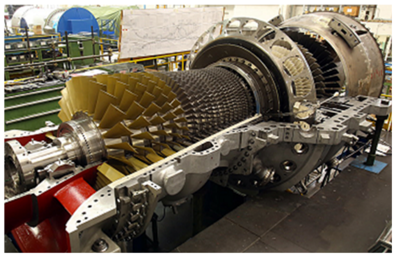
Lei Liu, Marcello Urgo
Mechanical Engineering Department
Politecnico di Milano
PMS 2022

POLITECNICO
MILANO 1863

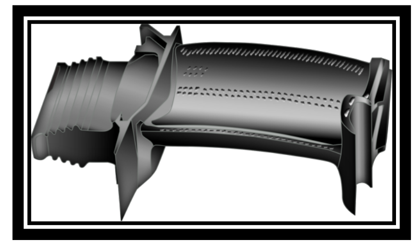
PMS 20
GNT 22

Motivation

Refurbished products are gaining importance in industrial sectors, specifically high-value products whose residual value is relevant, to guarantee the economic viability and sustainability of the remanufacturing at industrial level



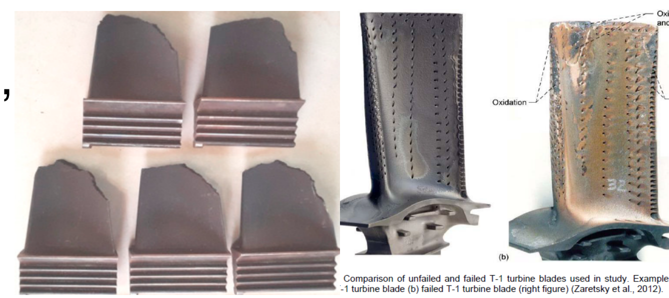
Gas Turbine for power generation



New Turbine Blade

Turbine Blades

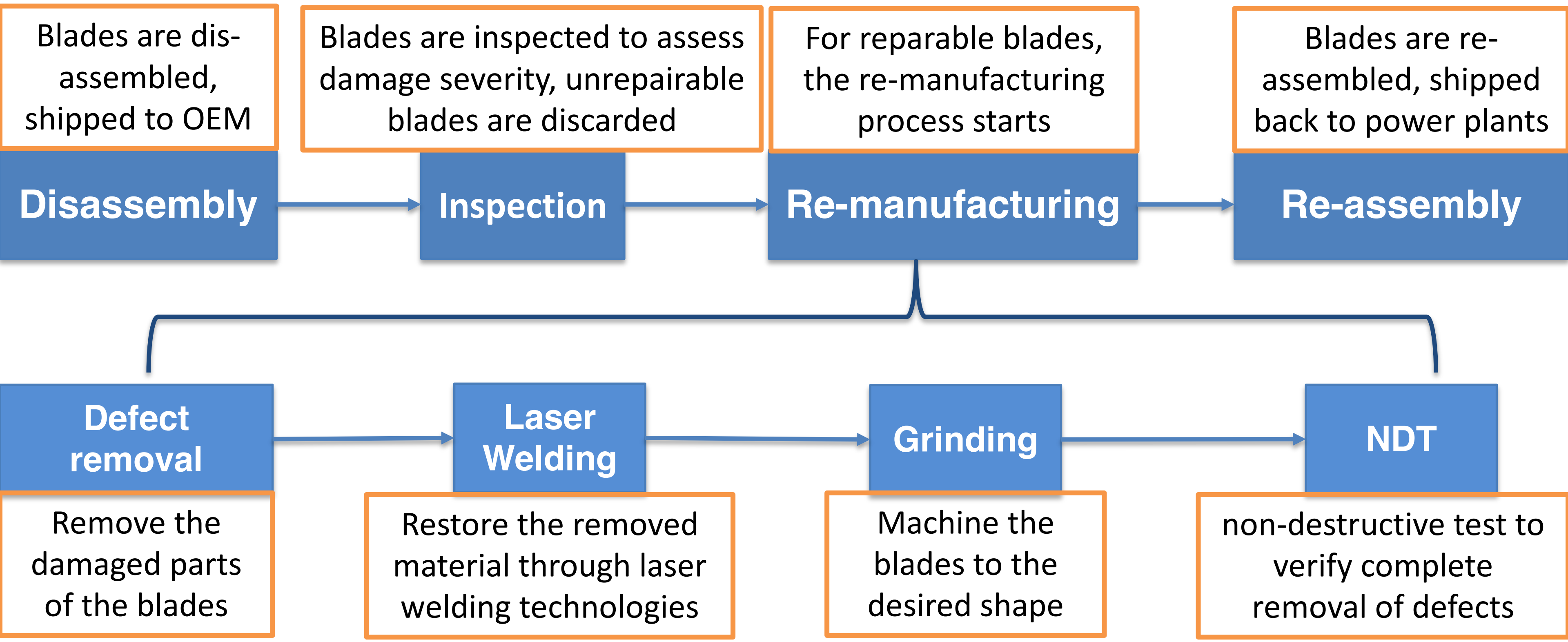
- **High-value** product (single blade ≈ VW Golf)
- The life cycle of gas turbines is 25,000 hours, after this, blades need to be disassembled and maintained
- Due to the high value, customers often require **refurbished** blades
- Matching the typical characteristic of **remanufacturing** and circular economy business model



Damaged Blades

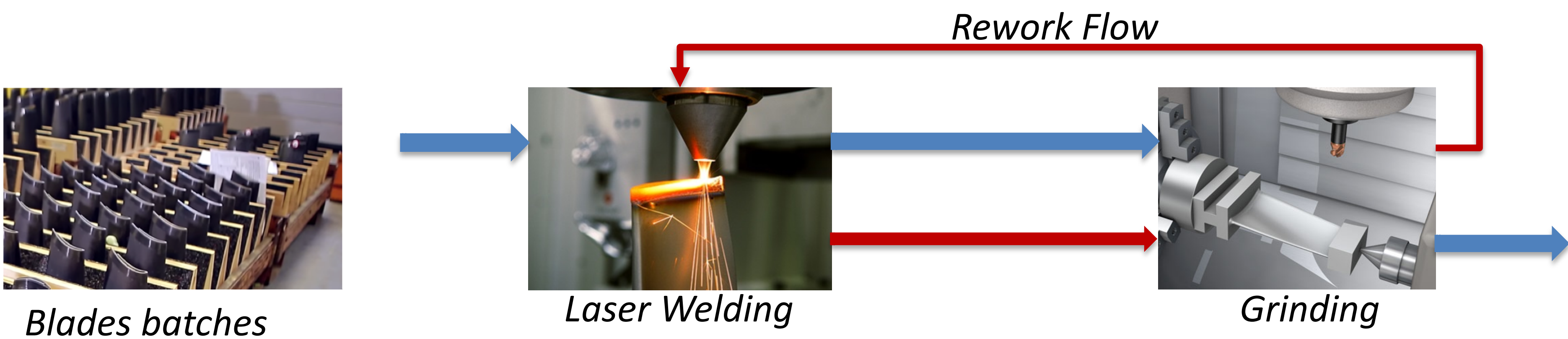
Remanufacturing Process

The blades are processed in **batches**



Remanufacturing Process

- We focus our attention on the subset of two operations of the repairing process, namely, the addition of materials through a **welding** process and the following **grinding** process, those are two bottleneck resources of the whole repair process
- A rework is needed for all the blades in a batch, thus, the same process is repeated twice on the same resources



Uncertainties

The repair process is affected by a significant degree of uncertainty

Blades entering the process with different wear states → Repairing time could be different for each blade → **Uncertain** processing time

Blades with severe defects could not be repaired → Some blades in a batch are discarded, and substituted with new ones → **Uncertain** number of blades in a repair batch

Grounding on these, the processing time of each batch of blades is uncertain

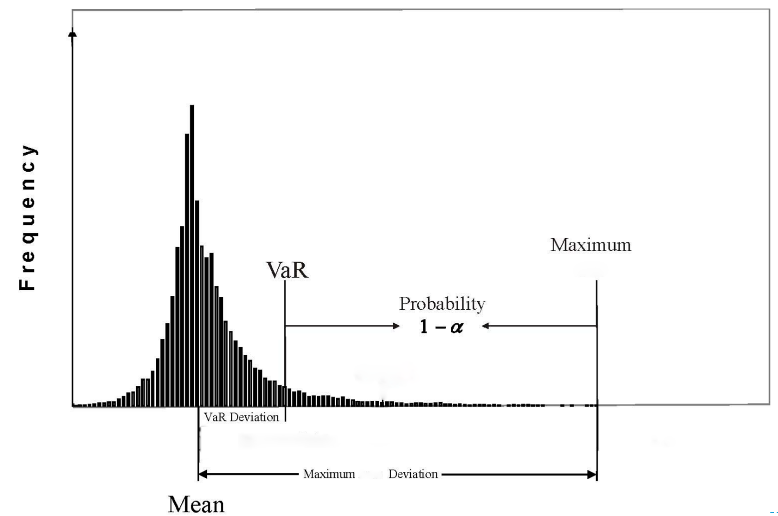
Value-at-Risk

Risk measure is one of popular robust scheduling approach which is used to determine the value of objective function in order to cover for unexpected uncertainties

1. Estimate the stochastic feature than expect objective function
2. Less conservative than minimax regret approach
3. Stem from financial optimization area and popular in scheduling field

Steps:

1. Describe the uncertainties associated to the processing times
2. Derive the distribution of objective function
3. Evaluate the value of VaR



- ➢ The maximum objective function we can experience in the best α percentage of the scenarios
- ➢ The α level confidence that the objective function will not excess VaR value

Problem Statement

Batches of blades have to undergo a two-stage process (laser welding and grinding)	**2-machine flow shop**
The **processing times** of each batch of blades on the different machines could vary according to the level of damage and number of blades in each batch	**Stochastic Processing time**
All the batches of blades need to be **reworked** by repeating the same sequence of operations	**Rework**
The objective function of the repair process is to maximize the utilization of the considered resources	**Makespan**
A robust scheduling approach is needed to mitigate the impact of uncertainty	**VaR**

A stochastic 2-machine flow shop scheduling problem with rework

$$F2 \mid (1,2,1,2) - reentrant, \boldsymbol{P}_{ij} \mid VaR(Cmax)$$

Problem Structure

A set of operations with precedence relations and stochastic durations

General distribution to match the characteristic of the industrial problem

The calculation of VaR needs the estimation of the distribution of objective function

Alternative scheduling decisions

Find optimal schedule minimizing the VaR

NP-hard problem

Solution Structure

Markov Activity Network

A set of operations with precedence relations and stochastic durations

Phase-type Distribution

General distribution to match the characteristic of the industrial problem

CTMC / Bisection

The calculation of **VaR** needs the estimation of the distribution of objective function

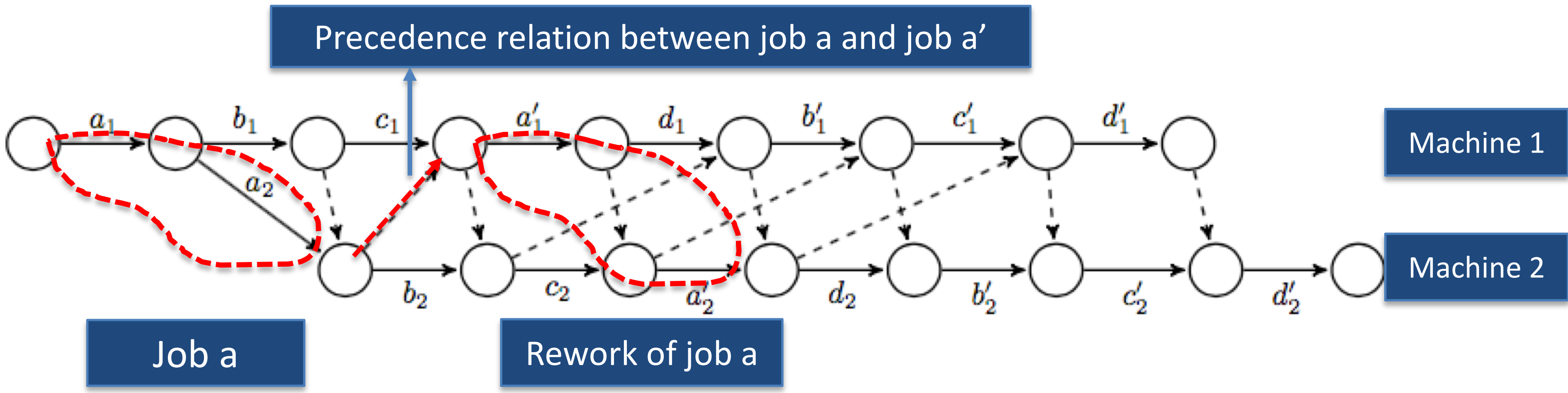
Branch-and-Bound

Alternative scheduling decisions

Find **optimal** schedule minimizing the VaR

Rework

We make proper hypothesis to model the rework activities



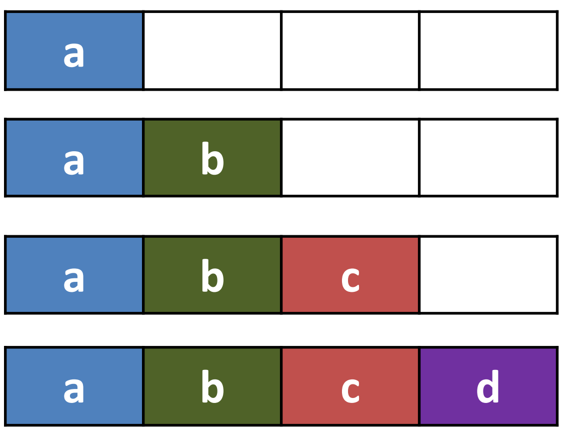
- Rework activities are modelled as additional jobs
- Specific constraints apply:
  - Rework job should be scheduled after the processing of the job, e.g., a1 → a1'
  - An offline inspection step is operated before the rework, hence, rework jobs cannot be processed earlier than 2 jobs after the corresponding original job (unless the jobs to be processed are less than 2)

Algorithm

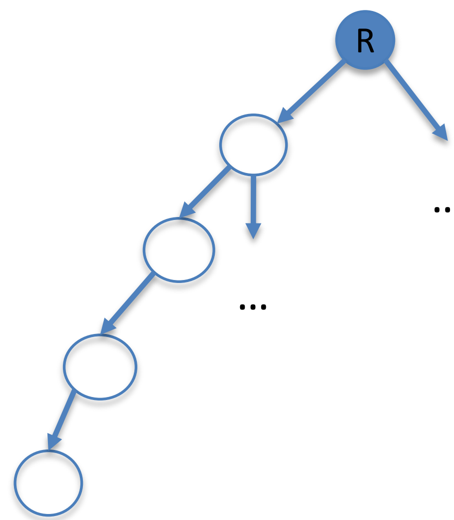
Branch-and-Bound

- Branching Scheme
- Full Schedule Evaluation
- Partial Schedule Evaluation
- Initial Schedule

Branching scheme: Sequencing the jobs starting from the beginning of the schedule



SCHEDULE

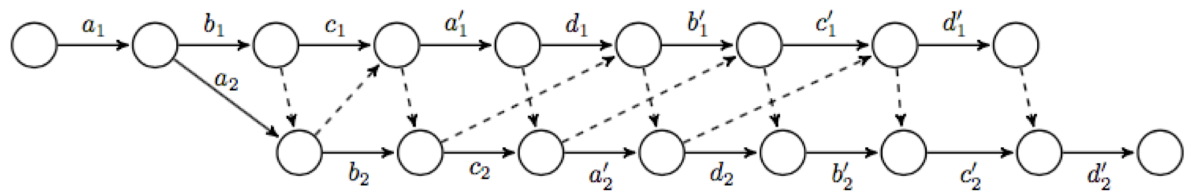


BRANCHING TREE

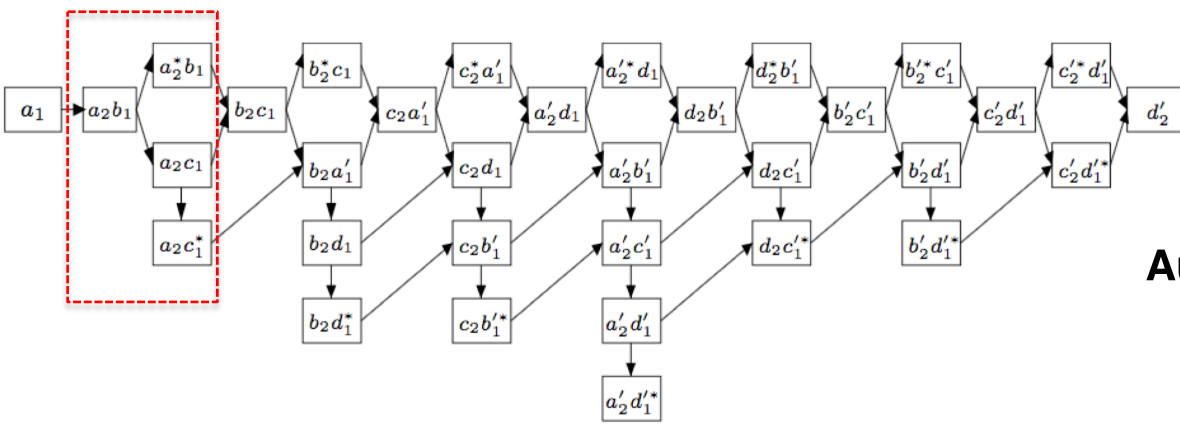
Evaluation – Full Schedule

Given a full schedule

a	b	c	a’	d	b’	c’	d’



AoA network



Automatic CTMC generation scheme

Infinitesimal generator **Matrix T** from the CTMC with advantage of Kronecker operations[1]

1. Angius, Alessio, András Horváth, and Marcello Urgo. "A Kronecker Algebra Formulation for Markov Activity Networks with Phase-Type Distributions." Mathematics 9.12 (2021): 1404.

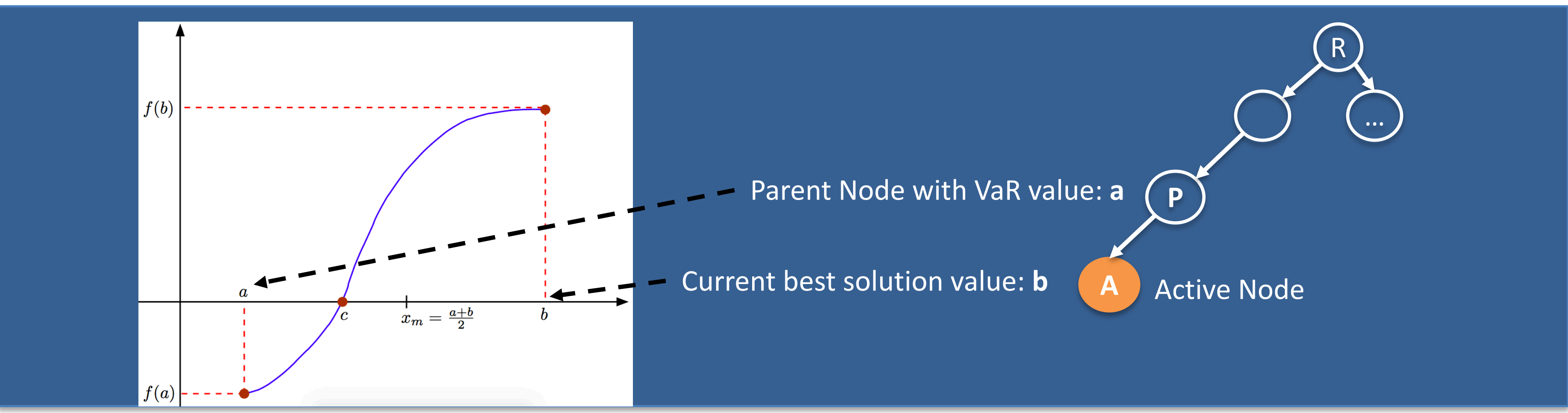
Evaluation – Full Schedule

Cdf of makespan **distribution**: $\mathrm{F_t(z)} = 1 - \beta * e^{\boldsymbol{T}z}\mathbf{1},$ β: initial state vector

VaR calculation by exploiting **bisection method**: $VaR_{\alpha}(\mathrm{t}) = \min\{\mathrm{z} \mid F_t(z) - \alpha \geq 0\}$

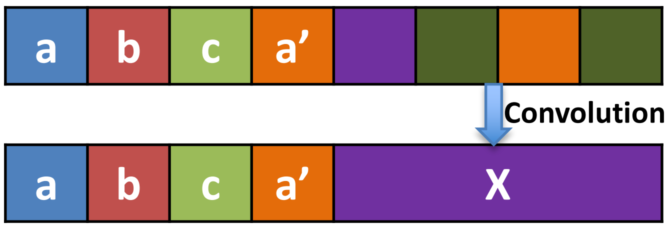
For an **active node A**, to reduce the iteration steps in the bisection method:

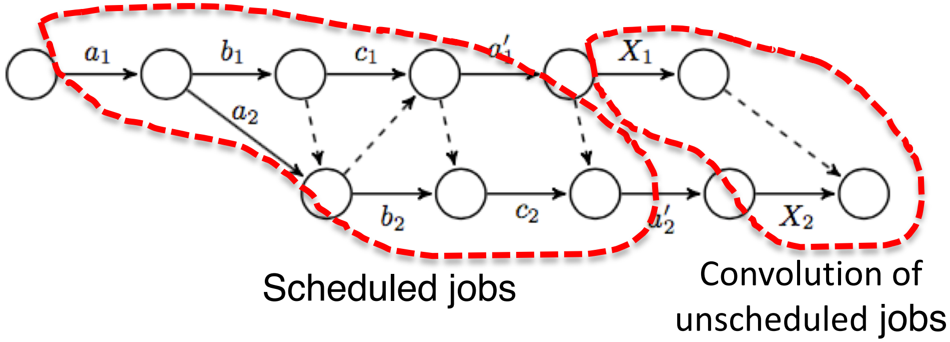
- The **lower bound support** is set as the VaR value of its **parent node P**
- The **upper bound support** is set as the **current best solution**
- If the root cannot be found in this support interval, this node can be pruned since it must be larger than the current best solution



Evaluation- Partial Schedule

In order to evaluate non-leaves nodes, we must be able to evaluate partial schedules





- **Precedence constraints** among operations for un-sequenced jobs are **relaxed**
- VaR of the makespan is a regular objective function, providing a lower bound

- CTMC generation scheme for leaf nodes can be used to estimate the lower bound of the VaR
- This lower bound of VaR will guide the search in the branch process of the algorithm to cut the branches without possibility finding optimal schedule

Initial Solution Schedule

To set the initial upper bound for the branch tree, an initial solution schedule is generated

Heuristic rule

This schedule is obtained by:
Arranging all the jobs according to the **decreasing order** of $E(J_1) - E(J_2)$

$E(J_1)$ and $E(J_2)$ are the expected value of the processing times of job j on machine 1 and 2.

This is the optimal schedule for 2-machine flow shop with exponential processing time[1]

Note: If the resulting schedule is in conflict with the constraints affecting the sequencing of rework jobs, they are shifted towards the right until the conflicts are eliminated.

The result schedule will be set as the **initial** solution schedule, with the corresponding VaR as the initial upper bound of algorithm.

1. Baker, K.R. and Trietsch, D., 2011. Three heuristic procedures for the stochastic, two-machine flow shop problem. Journal of Scheduling, 14(5), pp.445-454.

Algorithm Implementation

- Phase-type distributions are modeled using the **BuTools** library **:** a rich toolbox for Markovian performance evaluation
- The Branch-and-Bound algorithm is implemented with the support of the **Bob++** library: a framework for solving optimization problems with branch-and-X methods
- Matrix related operations are performed by means of the **Eigen library:** a high-level C++ library of template headers for linear algebra, matrix and vector operations
- The whole algorithm is coded in **C++**, and the experiments are run on a Windows 7 workstation with a 2.6 GHz processor and 64 GB of RAM

1. Horváth G, Telek M (2016) Butools 2: a rich toolbox for markovian performance evaluation.. URL http://webspn.hit.bme.hu/ telek/tools/butools/
2. DjerrahAetal(2006) Bob++:Framework for solving optimization problems with branch-and-bound methods. In: 2006 15th IEEE international conference on high performance distributed computing. IEEE.
3. Guennebaud G, Jacob B, et al. (2010) Eigen v3. http://eigen.tuxfamily.org

Data Generation

The proposed algorithm is tested on randomly generated instances

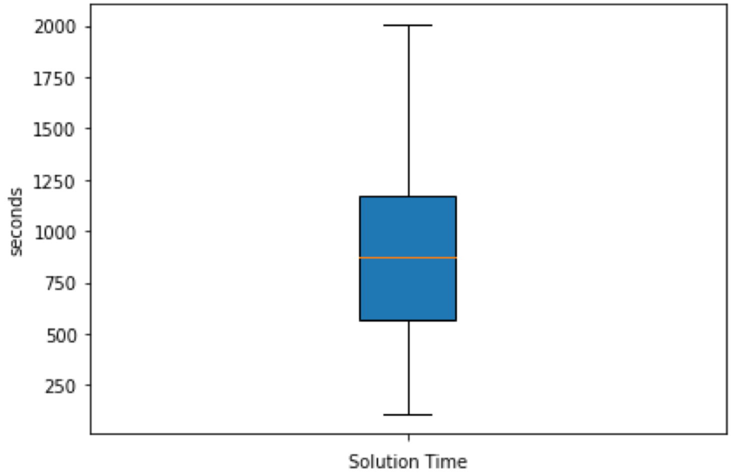
Data:

- n = 6 jobs, i.e., 12 jobs including rework jobs
- processing time of each job on each machine follows a **general phase-type** distribution, with **mean value** randomly sampled from [0,20],[30,50] and [60,80] and **number of phases** randomly sampled between [1, 4]
- risk level (10 and 20%) --> 90, 80% confidence level
- 50 instances in total are generated for 50 experiments

Algorithm Performance

We use the experiments to assess the performance for the proposed approach in terms of solution time / number of nodes explored

Job No.	Risk level	CPU time(s)		
		mean	min	max
6	10	877.6	110.8	1383.8
	20	900.9	143.6	2009.3
All		889.3	110.8	2009.1

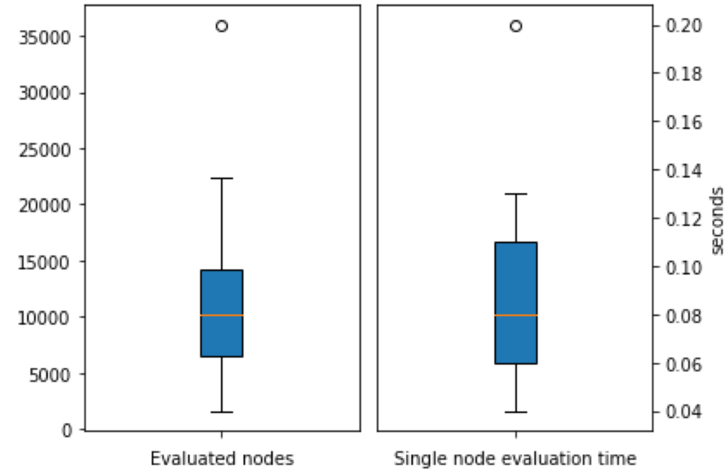


The proposed algorithm is able to find the optimal schedule in about 15 minutes on average (2 min – 30 min)

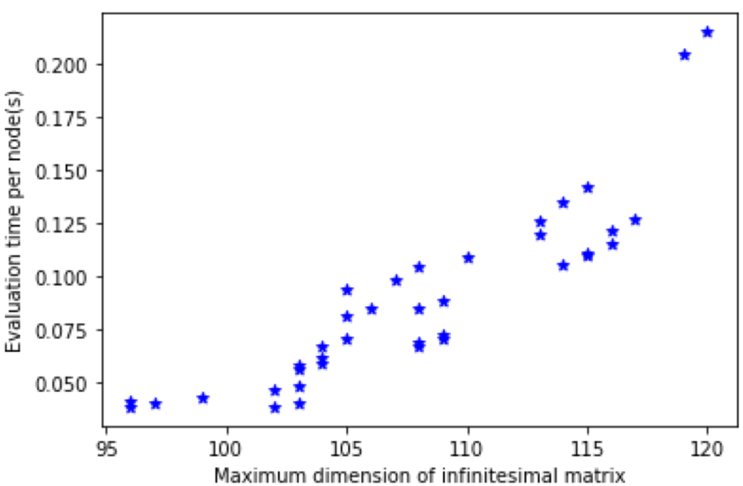
Algorithm Performance

- About 10,000 nodes are evaluated for each instance in the branching tree
- The average evaluation time for single node is about 0.08 seconds

Job No.	Risk level	Evaluated Nodes		
		mean	min	max
6	10	12262	1627	22433
	20	11442	2665	36028
All		11852	1627	36028



Node evaluation time entails the largest part of the solution time

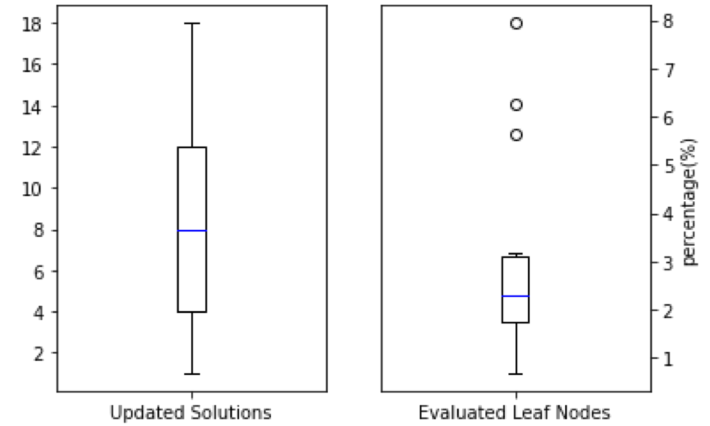


- There is a clear positive correlation between the average time to **evaluate a single node** and the maximum **dimension** of the infinitesimal generator **matrix**

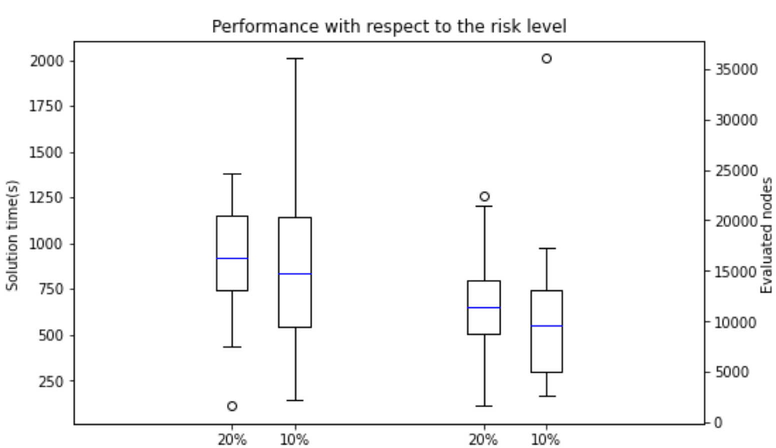
Algorithm Performance

We investigate the efficiency of initial solution schedule and lower bound

- The **Initial Solution** provides a reasonably good schedule, the optimal solution is updated less than 20 times.
- The **Leaf Nodes** evaluated count < 3% of all evaluated nodes, proving the efficiency of lower bound



We analyse the performance in terms of solution time and number of evaluated nodes, under different **risk levels**



- ANOVA, p-value 0.76, 0.84, respectively
- No statistical evidence to state that the solution time/number of evaluated nodes is affected by the risk level

Conclusion

1. Scheduling of remanufacturing activities for the repair of turbine blades
2. 2-machine flow shop scheduling problem with uncertain processing times and rework
3. Minimization of the VaR of the Makespan
4. Phase-type distributions to fit the uncertain processing times
5. Markov activity network(MAN) exploited to evaluate the distribution of the objective function
6. Branch-and-bound (B&B) algorithm to find the optimal schedule

Future Research

- The computation time to solve larger instance is rather high
  - Speed up the node evaluation
  - Tighter lower bounds
  - …
- The current rework is deterministic, probability of rework may be considered