Parallel machine match-up scheduling with manufacturing cost considerations

M. Selim Aktürk, Sinan Gürel, Alper Atamtürk

December, 2013

Table of contents

1. Introduction
   - Objectives
   - Literature Review
   - Contributions
2. Problem Definition
3. Numerical Example
4. Problem Formulations
   - Minimize Sum of Match-up Times
   - Minimize total manufacturing cost subject to a bound on sum of match-up times
   - Minimize maximum of match-up times
   - Minimize total manufacturing cost subject to a bound on maximum match-up time
5. Strong Conic Quadratic Formulations
6. Heuristic Approach
7. Computational Study
8. Conclusions

Rescheduling of disrupted schedules

- Match-up scheduling
  - Match-up time
  - Idle time v.s. controllable processing time
  - Match-up time v.s. manufacturing cost objectives

Idle time insertion

1. Mehta, S. V., & Uzsoy, R. M. (1998). Predictable scheduling of a job shop subject to breakdowns. *IEEE Transactions on Robotics and Automation*, *14*, 365-378.
   - Find a job sequence then apply a heuristic approach to insert idle times
2. Leus, R., & Herroelen, W. (2007). Scheduling for stability in single machine production. *Journal of Scheduling, 10(3)*, 223-235.
   - Consider minimizing expected weighted deviation between actual and planned job starting times
   - Find optimal job sequence and the optimal length of idle time following each job

Controllable processing times

1. Gürel, S., & Aktürk, M. S. (2007). Optimal allocation and processing time decisions on parallel CNC machines: ϵ-constraint approach. *European Journal of Operational Research, 183*, 591-607.
   - Consider minimizing total manufacturing cost subject to a given bound on the makespan objective in non-identical CNC machine environment

Match-up scheduling

1. Bean, J. C., Birge, J. R., Mittenthal, J., & Noon, C. E. (1991). Match-up scheduling with multiple resources, release dates and disruptions. *Operations Research, 39(3)*, 470-483.
2. Aktürk, M. S., & Görgülü, E. (1999). Match-up scheduling under a machine break down. *European Journal of Operations Research, 112*, 81-97.

- Consider heuristic approaches to find match-up times

Contributions

- Controllable processing times
- Match-up time / manufacturing cost tradeoffs
- Sum of match-up time or maximum match-up time objectives
- Handle the convex cost functions with Second Order Cone Programming
- Algorithm for approximately efficient solutions

Decision variables

x_{ij}: 1, if job j is assigned on machine i; 0, otherwise
z_j: 1, if start time of job j in S is selected as match-up time; 0, otherwise
y_{ij}: Compression on the processing time of job j on machine i

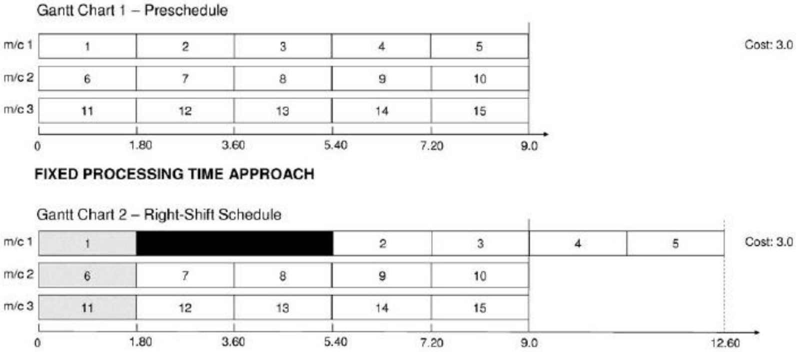
Parameters

- p_{ij}^u: Processing time upper bound that gives the minimum manufacturing cost for job j on machine i
- c_{ij}: manufacturing cost of job j on machine i with no compression
- u_{ij}: Maximum possible compression for job j on machine i
- $f_{ij}(y_{ij})$: Compression cost function for job j on machine i
- D_i: Available machining time capacity on machine i
- S: Preschedule on which a breakdown occurs
- y_{ij}^S: Compression on the processing time of job j on machine i
- s_j: Start time of job j in S
- M^*: Index of the disrupted machine
- t^*: Time of distruption on Machine M^*
- d^*: Duration of distruption on machine M^*

Parameters

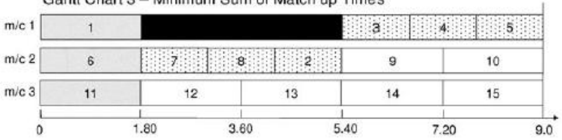
- J: Set of jobs not yet started processing at the time of disruption, i.e., $J = \{j | s_j \geq t^*\}$
- J_i: Subset of J scheduled on machine i in S
- J_i^m: Subset of J_i that can form match-up point on machine i, ie., $J_i^m = \{j \in J_i | s_j \geq t^* + d^*\}$ for $i = M^*$ and $J_i^m = \{j \in J_i | s_j \geq t^*\}$ for $i \neq \mathrm{M}^*$
- P_j: Set of jobs including job j and its predecessors which can form match-up point on the same machine in S
- E_i: End time of the last job on machine i in S

Fixed processing time approach



Controllable processing time approaches

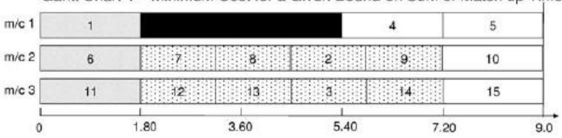
Gantt Chart 3 – Minimum Sum of Match up Times



Cost: 21.0

Sum of Match up Times: 16.2

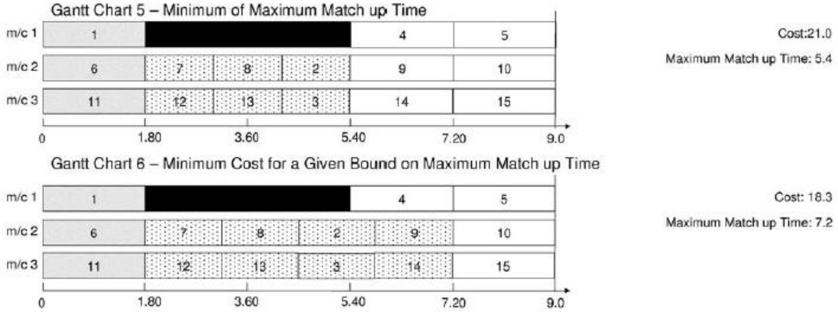
Gantt Chart 4 – Minimum Cost for a Given Bound on Sum of Match up Times



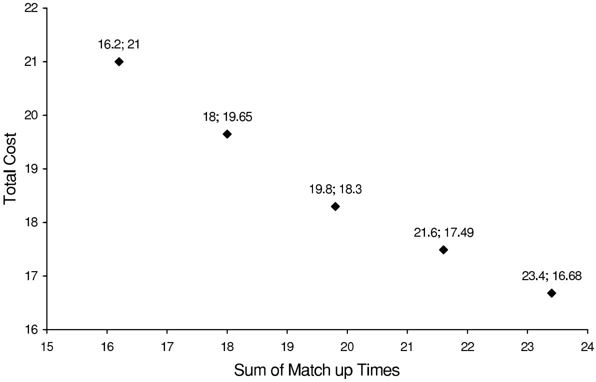
Cost:18.3

Sum of Match up Times: 19.8

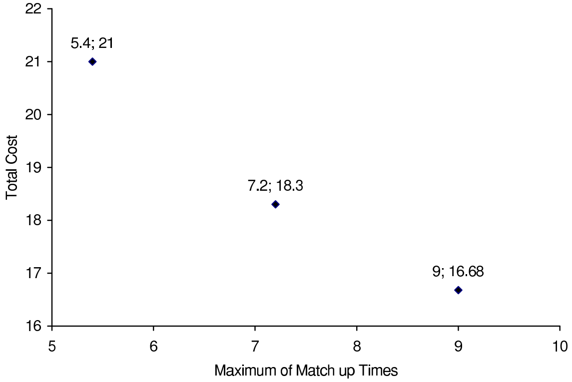
Controllable processing time approaches



Efficient solution set for total cost and sum of match-up times



Efficient solution set for total cost and minimum of maximum match- up time objectives



Minimize sum of match-up times

$$\min \sum_i \sum_{j \in J_i^m} s_j z_j + \sum_i E_i (1 - \sum_{j \in J_i^m} z_j)$$

$$\text{s.t.} \quad \sum_{j \in J} (p_{ij}^u x_{ij} - y_{ij}) \leq D_i, \qquad i = 1, \ldots, m \tag{1}$$

$$y_{ij} \leq x_{ij} u_{ij}, \qquad i = 1, \ldots, m \text{ and } j \in J \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad \forall j \in J \tag{3}$$

$$(\text{SM}) \qquad \sum_{j \in J_i^m} z_j \leq 1, \qquad i = 1, \ldots, m \tag{4}$$

Minimize sum of match-up times

$$\sum_{j_2 \in P_{j_1}} z_{j_2} \leq x_{ij_1}, \qquad i = 1, \ldots, m \text{ and } \forall j_1 \in J_i^m \tag{5}$$

$$(u_{ij_1} - y_{ij_1}^{\mathcal{S}}) \sum_{j_2 \in P_{j_1}} z_{j_2} \leq u_{ij_1} - y_{ij_1}, \qquad i = 1, \ldots, m \text{ and } \forall j_1 \in J_i^m \tag{6}$$

$$y_{ij_1}^{\mathcal{S}} \sum_{j_2 \in P_{j_1}} z_{j_2} \leq y_{ij_1}, \qquad i = 1, \ldots, m \text{ and } \forall j_1 \in J_i^m \tag{7}$$

$$x_{ij} \in \{0, 1\}, \ y_{ij} \in R_+, \qquad i = 1, \ldots, m, \text{ and } j \in J \tag{8}$$

$$z_j \in \{0, 1\}, \qquad j \in \cup_i J_i^m. \tag{9}$$

Minimize total manufacturing cost subject to a bound on sum of match-up times

change in the machining cost due to processing time compression $y \geq 0$

$$f(y) = ky^{a/b},$$

where a and b are integers satisfying $a \geq b > 0$ and $k > 0$, so that f is an increasing and convex function of the compression.

$$\text{(CSM)} \quad \begin{aligned} \min \; & \sum_i \sum_{j \in J} (c_{ij} x_{ij} + f_{ij}(y_{ij})) \\ \text{s.t.} \; & \sum_i \sum_{j \in J_i^m} s_j z_j + \sum_i E_i (1 - \sum_{j \in J_i^m} z_j) \leq T \\ & \text{and } (1) - (9). \end{aligned} \qquad (10)$$

Minimize maximum of match-up times

$$
\begin{aligned}
& \min \quad W \\
(\mathrm{MM}) \quad & \text{s.t.} \sum_{j \in J_i^m} s_j z_j + E_i \Big(1 - \sum_{j \in J_i^m} z_j\Big) \leq W, \quad i = 1, \ldots, m \qquad (11) \\
& \text{and } (1) - (9).
\end{aligned}
$$

Minimize total manufacturing cost subject to a bound on maximum match-up time

$$
\begin{aligned}
\text{min} \quad & \sum_{i} \sum_{j \in J^T} (c_{ij} x_{ij} + f_{ij}(y_{ij})) & & \\
\text{s.t.} \quad & \sum_{j \in J^T} (p^u_{ij} x_{ij} - y_{ij}) \leq T_i, & i = 1, \ldots, m & \quad (12) \\
\text{(CMM)} \quad & y_{ij} \leq x_{ij} u_{ij}, & i = 1, \ldots, m \text{ and } j \in J^T & \quad (13) \\
& \sum_{i=1}^{m} x_{ij} = 1, & j \in J^T & \quad (14) \\
& x_{ij} \in \{0, 1\}, \ y_{ij} \in R_+, & i = 1, \ldots, m \text{ and } j \in J^T. & \quad (15)
\end{aligned}
$$

Strong conic quadratic formulations

Replace each term $y_{ij}^{a_{ij}/b_{ij}}$ in the objective with an auxiliary variable t_{ij} and add $y_{ij}^{a_{ij}/b_{ij}} \leq t_{ij}$ into the constraints as below:

$$\text{(CSM1)} \quad \begin{aligned} \min \ & \sum_i \sum_{j \in J} (c_{ij} x_{ij} + k_{ij} t_{ij}) \\ \text{s.t.} \ & y_{ij}^{a_{ij}/b_{ij}} \leq t_{ij}, \quad i = 1, \ldots, m, \ j \in J \\ & \text{and } (1) - (9) \text{ and } (10). \end{aligned} \tag{16}$$

As $y_{ij}, t_{ij} \geq 0$ and $b_{ij} > 0$ for all i, j, inequality (16) is equivalent to

$$y_{ij}^{a_{ij}} \leq t_{ij}^{b_{ij}}. \tag{17}$$

which can be strengthened as

$$y_{ij}^{a_{ij}} \leq t_{ij}^{b_{ij}} x_{ij}^{a_{ij} - b_{ij}}. \tag{18}$$

Example on strong conic quadratic formulation

Let $f(y) = y^{5/4}$. Write inequality $y^5 \leq t^4 x,\ \ y, t, x \geq 0$ then put it in the form

$$y^8 \leq t^4 x y^3,\ \ y, t, x \geq 0.$$

Express equivalently by using the following hyperbolic constraints:

$$v_1^2 \leq xy,\ \ x, y \geq 0,$$
$$v_2^2 \leq y v_1,\ y, v_1 \geq 0,$$
$$y^2 \leq t v_2,\ t, v_2 \geq 0.$$

The hyperbolic constraints are then written in conic quadratic form as

$$||2v_1,\ y - x|| \leq y + x,$$
$$||2v_2, y - v_1|| \leq y + v_1,$$
$$||2y, t - v_2|| \leq t + v_2.$$

Heuristic search algorithm

- Generates a set of approximately efficient solutions
- Consider two bi-criteria rescheduling problems
  1. Total manufacturing cost/Sum of match-up times tradeoffs
  2. Total manufacturing cost/Maximum match-up time tradeoffs

Heuristic search algorithm

Algorithm 1 Heuristic algorithm for finding approximate efficient solutions.

Require: A preschedule $\mathcal{S}$ and a disruption on one of the machines.

Solve the problem SM (MM) to find an initial schedule S, match-up times T_i, and job pool P.

repeat

Apply `1-move` Algorithm.

Apply `2-swap` Algorithm.

Report the generated solution.

until `Augment Job Pool` is `False`.

A subproblem

Given a match-up time T_i and a set of jobs J_c to be scheduled before T_i on machine i, find optimal compression levels for the jobs

$$\text{(COMPi)} \quad \begin{aligned} \min \quad & \sum_{j \in J_c} f_{ij}(y_{ij}) \\ \text{s.t.} \quad & \sum_{j \in J_c} (p_{ij}^u - y_{ij}) \leq T_i \qquad (19) \\ & 0 \leq y_{ij} \leq u_{ij}, \;\; j \in J_c. \qquad (20) \end{aligned}$$

Job pool

Defn: Set of jobs to be rescheduled; if it is not yet started at the time of the breakdown and its start time precedes the given match-up time on its machine.

- Adding a new job $\Rightarrow$ increases the sum of match-up times, but may decrease the manufacturing cost, after reallocating the jobs and solving the subproblems on each machine

Algorithm for augment job pool

Algorithm 2 Augment Job Pool

Require: Given match-up times T_i for each machine and job pool P.
  if $T_i = E_i$ for all i **then**
    return .
  else
    Calculate Δ_i for the machines with $T_i \neq E_i$.
    Select i^* with minimum Δ_i.
    $T_{i^*} \leftarrow s_j$, where j is the next job on i^*; $P \leftarrow P \cup j$.
    return .

Δ_i**:** an estimate for the ratio of the cost change to the match-up time change that will be obtained by moving the match-up point to the start time of the next job.

$$\Delta_i := \frac{k_{ij}\hat{y}_{ij}^{a_{ij}/b_{ij}} - k_{ij}{y_{ij}^*}^{a_{ij}/b_{ij}} - \lambda_i^*(\hat{y}_{ij} - y_{ij}^*)}{p_{ij}^u - y_{ij}^*},$$

where $\hat{y}_{ij} = \min((\partial f_{ij}/\partial y_{ij})^{-1}(\lambda_i^*), u_{ij})$.

1-move improvement search

- Move of a job in the job pool from its current machine to another machine

Lemma

(Lower Bound for a 1-move) For a given schedule let λ_{i_1} and λ_{i_2} be optimal dual prices for COMPi$_1$ and COMPi$_2$, respectively, and y_{i_1j} be the compression of job j on machine i_1. Then, a lower bound for the cost change that will result by moving job j from machine i_1 to i_2 is as stated below:

$$\begin{aligned} LB(j:(i_1 \to i_2)) = \quad & -\lambda_{i_1}(p_{i_1j} - y_{i_1j}) - c_{i_1j} - f_{i_1j}(y_{i_1j}) \\ & + c_{i_2j} + f_{i_2j}(\hat{y}_{i_2j}) + \lambda_{i_2}(p_{i_2j} - \hat{y}_{i_2j}) \end{aligned}$$

where $\hat{y}_{i_2j} = \min((\partial f_{i_2j}/\partial y_{i_2j})^{-1}(\lambda_{i_2}), u_{i_2j})$.

- cost reduction by removing j from machine i_1
- cost increase by inserting job j into machine i_2

1-move improvement search algorithm

Algorithm 3 `1-move` Search Algorithm

```
Require: A given schedule S and a job pool P.
  repeat
    Generate all feasible 1-moves for each j in P in S;
    Calculate LB for all moves;
    if LB ≥ 0 for all feasible moves then
      BREAK;
    else
      Make a list of moves with LB < 0 in nondecreasing order of LB's.
      Initialize: found_improving_move ← False,
      end_of_list ← False;
      while NOT found_improving_move and NOT end_of_list do
        Do the next move in the list;
        if It is the last move in the list then
          end_of_list ← True
        Solve COMPi for affected machines;
        New schedule is S′;
        if COST(S′) < COST(S) then
          S ← S′;
          found_improving_move ← True; improved ← True;
      if NOT found_improving_move then
        improved ← False
  until NOT improved.
```

2-swap improvement search

- Exchange of two jobs, j_1 and j_2, between two machines i_1 and i_2
- Combination of two `1-move`'s

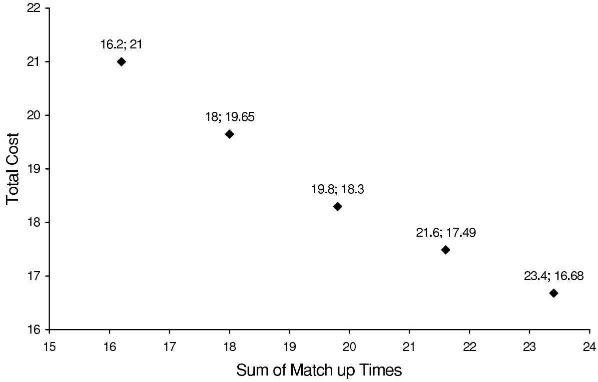
Lemma

(Lower Bound for a 2-swap) For a given schedule let λ_{i_1} and λ_{i_2} be optimal dual prices for COMPi_1 and COMPi_2, respectively, and $y_{i_1 j_1}$ and $y_{i_2 j_2}$ be the compression of the jobs j_1 and j_2 on machine i_1 and i_2, respectively. Then, a lower bound for the cost change that will result by swapping jobs j_1 and j_2 between machines i_1 and i_2 is calculated as below:

$$LB(j_1 \leftrightarrow j_2) = \lambda_{i_1}(p_{i_1 j_1} - y_{i_1 j_1} - p_{i_1 j_2} + \hat{y}_{i_1 j_2}) - c_{i_1 j_1} - f_{i_1 j_1}(y_{i_1 j_1}) + c_{i_1 j_2} + f_{i_1 j_2}(\hat{y}_{i_1 j_2})$$
$$+ \lambda_{i_2}(p_{i_2 j_2} - y_{i_2 j_2} - p_{i_2 j_1} + \hat{y}_{i_2 j_1}) - c_{i_2 j_2} - f_{i_2 j_2}(y_{i_2 j_2}) + c_{i_2 j_1} + f_{i_2 j_1}(\hat{y}_{i_2 j_1}),$$

where
$\hat{y}_{i_2 j_1} = \min((\frac{\partial f_{i_2 j_1}}{\partial p_{i_2 j_1}})^{-1}(\lambda_{i_2}), u_{i_2 j_1})$ *and* $\hat{y}_{i_1 j_2} = \min((\frac{\partial f_{i_1 j_2}}{\partial y_{i_1 j_2}})^{-1}(\lambda_{i_1}), u_{i_1 j_2})$.

Efficient frontier of sum of match-up time versus manufacturing cost tradeoff



Experimental factors

Factor	Description	Levels: Low	Levels: High
ld	length of distruption	2.0	5.0
κ	capacity factor	0.25	0.30
n	number of jobs	50	100
m	number of machines	2	3

$$D_i = \kappa \times \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}^{u}}{m} .$$

duration of breakdown generated from Uniform $[ld - 1.0, ld + 1.0]$

Table : Results for Strong Conic Quadratic Formulations

				Sum of Match-Up Times CSM2				Maximum Match-Up Time CMM2			
ld	κ	n	m	opt	egap (%)	node	cpu	opt	egap (%)	node	cpu
2.0	0.25	50	2	15	0	35.7	9.8	15	0	2.9	1.3
			3	15	0	188.7	71.1	15	0	14.6	4.1
		100	2	15	0.01	106	55	15	0	0	0.6
			3	14	0.01	311.3	340.2	15	0	6.9	6.4
	0.30	50	2	15	0.01	41.3	12.6	15	0	2.6	1
			3	13	0.07	280.9	176.1	15	0	15.9	4.5
		100	2	15	0.01	82.5	58.1	15	0	2.4	2.1
			3	13	0.01	463.4	502.9	15	0	14	10.6
5.0	0.25	50	2	15	0	54.2	13.9	15	0	1.1	0.7
			3	14	0.03	306.9	161.8	15	0	10.2	3.5
		100	2	15	0.01	120.1	80	15	0	2.2	2
			3	11	0.05	624.4	567.6	15	0.01	11.7	7.9
	0.30	50	2	15	0	35.9	9.6	15	0	2.8	1
			3	14	0.01	496.1	213.7	15	0	25.1	7.7
		100	2	14	0.02	213.8	157.1	15	0	0.9	1.3
			3	8	0.05	710.2	792.7	15	0.01	17.6	14.2
Average				13.8	**0.02**	254.5	**201.4**	15	**0.001**	8.2	**4.3**

Analysis on heuristic algorithm

- For each problem instances, first run heuristic which generates a set of approximately efficient solutions
- Then, solve the strong conic quadratic formulations of CSM (CMM) problems for the three selected solutions generated by the heuristic.

$$\text{Relative Gap} = \frac{\text{Cost of heuristic - Optimal cost of CSM}}{\text{Optimal cost of CSM}}$$

Table : Heuristic Algorithm Performance

				Sum of Match-Up Times					Maximum Match-Up Time				
				# sol	cpu	gap (%)			# sol	cpu	gap(%)		
ld	κ	n	m	mean	mean	mean	min	max	mean	mean	mean	min	max
2.0	0.25	50	2	19.8	1.26	0.04	0.00	0.21	27.6	2.05	0.28	0.00	1.06
			3	13.8	0.37	1.21	0.00	6.23	31.8	1.38	0.35	0.00	1.43
		100	2	70.2	8.58	0.21	0.00	0.66	67.6	12.03	0.17	0.08	0.24
			3	49	3.77	0.09	0.00	0.18	78.8	12.51	0.11	0.00	0.37
	0.30	50	2	18.6	0.64	0.39	0.00	1.92	31.4	2.30	0.21	0.00	0.62
			3	14.4	1.15	0.82	0.00	3.04	34.4	4.63	0.73	0.10	2.08
		100	2	50.4	3.77	0.09	0.00	0.55	72.6	10.22	1.17	0.00	15.75
			3	27.6	1.35	0.03	0.00	0.09	80.6	27.16	0.08	0.00	0.23
5.0	0.25	50	2	19.4	1.94	0.14	0.00	0.48	20.6	2.00	0.48	0.00	1.60
			3	21.2	1.16	0.49	0.00	1.41	25.0	1.72	1.94	0.00	18.00
		100	2	53.6	11.74	0.30	0.00	0.93	57.0	9.97	0.41	0.16	0.86
			3	57	10.33	0.26	0.00	0.92	70.2	15.60	0.38	0.07	1.04
	0.30	50	2	66.2	2.10	0.21	0.00	0.61	24.6	2.28	1.89	0.00	17.94
			3	19.4	1.23	1.58	0.17	4.11	31.2	2.44	0.36	0.00	1.25
		100	2	69.4	12.89	0.17	0.01	0.50	64.8	9.05	0.46	0.25	0.73
			3	48	6.26	0.97	0.00	2.68	74.0	24.80	2.64	0.09	24.04
Average				38.6	**4.28**	**0.44**	0.01	1.53	49.5	**8.76**	**0.73**	0.05	5.45

Conclusions

- Introduce an important flexibility with controllable processing times to rescheduling problems under a single machine breakdown.
- Creating alternative time/cost tradeoffs to the decision maker
- Developing a heuristic algorithm to provide alternative solutions
- Handling convex cost function with new advances in conic mixed integer programming

Thank you for listening.

Karush-Kuhn-Tucker conditions

$$\frac{\partial f_{ij}}{\partial y_{ij}} - \lambda - \nu_j + \eta_j = 0, \;\; j \in J_c \tag{21}$$

$$\lambda \left(\sum_{j \in J_c} (p_{ij}^u - y_{ij}) - T_i \right) = 0 \tag{22}$$

$$\nu_j y_{ij} = 0, \;\; j \in J_c \tag{23}$$

$$\eta_j (y_{ij} - u_{ij}) = 0, \;\; j \in J_c \tag{24}$$

$$\nu_j \geq 0, \;\; \eta_j \geq 0, \;\; j \in J_c \tag{25}$$

$$\lambda \geq 0 \tag{26}$$

and inequalities (19)–(20).

Lemma

Let y_{ij}^* and $(\lambda^*, \eta^*, \nu^*)$ be an optimal pair of solutions to COMPi. For each job j, we have the following:

$$(\partial f_{ij}/\partial y_{ij})(y_{ij}^*) \begin{cases} \geq \lambda^*, & \text{if } y_{ij}^* = 0; \\ = \lambda^*, & \text{if } 0 < y_{ij}^* < u_{ij}; \\ \leq \lambda^*, & \text{if } y_{ij}^* = u_{ij}. \end{cases}$$

Because $\lambda^* \geq 0$ and $(\partial f_{ij}/\partial y_{ij})(0) = 0$, the first part holds only when $\lambda^* = 0$. in which case the other parts imply that $y_{ij}^* = 0$ for j. Thus Lemma 3 states that whenever $\lambda^* > 0$, the partial derivative of the cost function for each job must be equal unless its compression is at its upper bound u_{ij}. For the considered compression cost function f, $(\partial f_{ij}/\partial y_{ij})(0) = 0$ always holds, so condition iii holds only if $\lambda^* = 0$ and $y_{ij}^* = 0$ for all i and j. Here, λ^* is the rate of change of the optimal cost as T_i changes. But also from the lemma, the rate of change of the optimal cost for each job with $0 < y_{ij}^* < u_{ij}$ is also equal to λ^*.

Strong conic quadratic formulations

$$
\text{(CSM2)} \quad \begin{aligned} \min \ & \sum_{i} \sum_{j \in J} (c_{ij} x_{ij} + k_{ij} t_{ij}) \\ \text{s.t.} \ & y_{ij}^{a_{ij}} \leq t_{ij}^{b_{ij}} x_{ij}^{a_{ij}-b_{ij}}, \quad i = 1, \ldots, m, \ j \in J \\ & \text{and } (1) - (9) \text{ and } (10). \end{aligned} \qquad (27)
$$