.inf INSTITUTO DE INFORMÁTICA UFRGS

INCLUDING WORKERS WITH DISABILITIES IN FLOW SHOP SCHEDULING

Germano C. Carniel, Alexander J. Benavides, Marcus Ritt, Cristóbal Miralles

IEEE CASE — August 2015

AGENDA

- Workers are commonly assumed to have equal skills.
- Often wrong, in particular for persons with disabilities.
- World Health Organization (2011) estimate: 15%-20% of the world population has some disability.
- Persons with disabilities suffer from higher unemployment rates.
- Worker with disabilities
  - have usually higher processing time than regular workers;
  - may be unable to operate some machines.

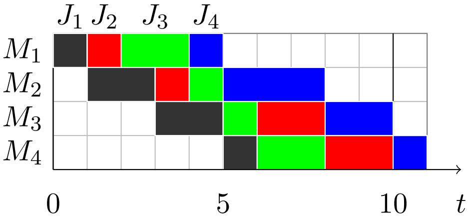
1 WORKERS WITH DISABILITIES IN FLOW SHOPS

FLOW SHOP SCHEDULING

- Flow Shop Scheduling Problem (FSSP)
  - Schedule jobs $J_1, \ldots, J_n$ on machines $M_1, \ldots, M_m$.
  - Job J_i must be processed on machine M_r in time p_{ri}.
  - No preemption.
  - Each machine processes only one job at a time.
  - Objective: minimize the *makespan*.
- Permutation Flow Shop Scheduling Problem (PFSSP)
  - Jobs are processed on all machines in the same order.
- NP-Hard for three or more machines (Garey and Johnson 1979).

FLOW SHOP SCHEDULING – EXAMPLE

	Machine			
Job	M_1	M_2	M_3	M_4
J_1	1	2	2	1
J_2	1	1	2	2
J_3	2	1	1	2
J_4	1	3	2	1



WORKERS WITH DISABILITIES IN FLOW SHOPS

WORKERS WITH DISABILITIES IN FLOW SHOPS

- Scheduling problems with
  - mainly heterogenous workers: sheltered work centres for disabled
  - a small percentage of heterogenous workers.
- *Focus here*: assign one or two parallel workers with disabilities to a machine they can operate and find an optimal schedule
  - Equals to 5%-40% of workers with disabilities in standard instances.
- Four problem variants of the Flow Shop Insertion and Scheduling Problem (FSISP)
  - FSISP: single worker, flow shop
  - HFSISP: two parallel workers, flow shop
  - PFSISP: single worker, permutation flow shop
  - HPFSISP: two parallel workers, permutation flow shop

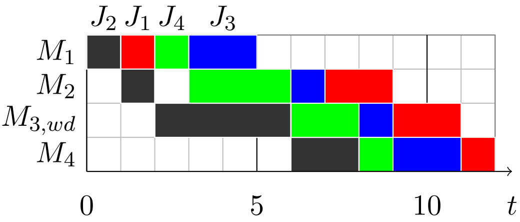
WORKERS WITH DISABILITIES IN FLOW SHOPS

- Scheduling problems with
  - mainly heterogenous workers: sheltered work centres for disabled
  - a small percentage of heterogenous workers.
- *Focus here*: assign one or two parallel workers with disabilities to a machine they can operate and find an optimal schedule
  - Equals to 5%-40% of workers with disabilities in standard instances.
- Four problem variants of the Flow Shop Insertion and Scheduling Problem (FSISP)
  - FSISP: single worker, flow shop
  - HFSISP: two parallel workers, flow shop
  - PFSISP: single worker, permutation flow shop
  - HPFSISP: two parallel workers, permutation flow shop

This paper

WORKERS WITH DISABILITIES IN FLOW SHOPS – EXAMPLE

	Regular				With disabilities			
Job	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4
J_1	1	2	2	1	2	4	2	∞
J_2	1	1	2	2	1	1	4	∞
J_3	2	1	1	2	4	2	1	∞
J_4	1	3	2	1	1	4	2	∞



Mathematical models

PFSISP: MATHEMATICAL MODEL

$$\textbf{min.} \quad \sum_{p\in[n]} TT_{1p} + \sum_{q\in[2,m]} TT_{qn} + \sum_{q\in[m-1]} Y_{qn}, \quad \textbf{s.t.}, \tag{1}$$

$$\sum_{i\in[n]} Z_{ij} = 1, \qquad \forall i \in [m], \tag{2}$$

$$\sum_{j\in[n]} Z_{ij} = 1, \qquad \forall i \in [n], \tag{3}$$

$$TT_{1,j-1} - TT_{r,j-1} + \sum_{q\in[r-1]} TT_{qj} - TT_{q,j-1}$$
$$+ \sum_{q\in[r-1]} Y_{qj} - Y_{q,j-1} \geq 0, \qquad \forall r \in [2,m], j \in [2,n], \tag{4}$$

$$TT_{rj} = \sum_{i\in[n]} p_{ri}(1 - X_r)Z_{ij} + d_{ri}X_rZ_{ij}, \qquad \forall r \in [m], j \in [n], \tag{5}$$

$$\sum_{r\in A} X_r = 1. \tag{6}$$

PFSISP: MATHEMATICAL MODEL

$$\textbf{min.} \quad \sum_{p\in[n]} TT_{1p} + \sum_{q\in[2,m]} TT_{qn} + \sum_{q\in[m-1]} Y_{qn}, \quad \textbf{s.t.}, \tag{1}$$

$$\sum_{i\in[n]} Z_{ij} = 1, \qquad \forall i \in [m], \tag{2}$$

$$\sum_{j\in[n]} Z_{ij} = 1, \qquad \forall i \in [n], \tag{3}$$

$$TT_{1,j-1} - TT_{r,j-1} + \sum_{q\in[r-1]} TT_{qj} - TT_{q,j-1}$$

$$+ \sum_{q\in[r-1]} Y_{qj} - Y_{q,j-1} \geq 0, \qquad \forall r \in [2,m], j \in [2,n], \tag{4}$$

- Based on the best model for the PFSSP (Tseng and Stafford 2007).
- Main ideas
  - Assign jobs to sequence positions.
  - Represent a schedule by operation waiting times Y_{rj}.
- Extended to include a worker assignment.

$$TT_{rj} = \sum_{i\in[n]} p_{ri}(1 - X_r)Z_{ij} + d_{ri}X_rZ_{ij}, \qquad \forall r \in [m], j \in [n], \tag{5}$$

$$\sum_{r\in A} X_r = 1. \tag{6}$$

HPFSISP: MATHEMATICAL MODEL

$$\textbf{min.} \quad C_{\mathsf{max}}, \qquad \textbf{s.t.} \tag{7}$$

$$C_{\mathsf{max}} \geq C_{jm}, \qquad \forall j \in [n], \tag{8}$$

$$\sum\nolimits_{l \in [2]} U_{jkl} = 1, \qquad \forall j, k \in [m], \tag{9}$$

$$U_{jk2} \leq X_k, \qquad \forall j, k, \tag{10}$$

$$C_{jk} - T_{jk} \geq C_{j,k-1}, \qquad \forall j, k, \tag{11}$$

$$Q(2 - U_{jkl} - U_{qkl} + P_{jq}) + C_{jk} - T_{jk} \geq C_{qk} \qquad \forall j, q \in [n], k, l \in [2], \tag{12}$$

$$Q(3 - U_{jkl} - U_{qkl} - P_{jq}) + C_{qk} - T_{qk} \geq C_{jk}, \qquad \forall j, q, k, l, \tag{13}$$

$$T_{jk} = p_{jk}(1 - X_r) + \sum\nolimits_{l \in [2]} (d_{jkw} X_k W_{wl}), \qquad \forall j, k, l, w \in [2], \tag{14}$$

$$\sum\nolimits_{k \in A} X_k = 1, \tag{15}$$

$$\sum\nolimits_{l \in [2]} W_{wl} = 1, \qquad \forall w, \tag{16}$$

$$\sum\nolimits_{w \in [2]} W_{wl} = 1, \qquad \forall l, \tag{17}$$

$$C_{jk} \geq 0 \qquad \forall j, k. \tag{18}$$

HPFSISP: MATHEMATICAL MODEL

$$\textbf{min.} \quad C_{\max}, \qquad \textbf{s.t.} \tag{7}$$

$$C_{\max} \geq C_{jm}, \qquad \forall j \in [n], \tag{8}$$

$$\sum_{l \in [2]} U_{jkl} = 1, \qquad \forall j, k \in [m], \tag{9}$$

$$U_{jk2} \leq X_k, \qquad \forall j, k, \tag{10}$$

$$C_{jk} - T_{jk} \geq C_{j,k-1}, \qquad \forall j, k, \tag{11}$$

$$Q(2 - U_{jkl} - U_{qkl} + P_{jq}) + C_{jk} - T_{jk} \geq C_{qk} \qquad \forall j, q \in [n], k, l \in [2], \tag{12}$$

$$Q(3 - U_{jkl} - U_{qkl} - P_{jq}) + C_{qk} - T_{qk} \geq C_{jk}, \qquad \forall j, q, k, l, \tag{13}$$

$$T_{jk} = p_{jk}(1 - X_r) + \sum_{l \in [2]} (d_{jkw} X_k W_{wl}), \qquad \forall j, k, l, w \in [2], \tag{14}$$

$$\sum_{k \in A} X_k = 1, \tag{15}$$

$$\sum_{l \in [2]} W_{wl} = 1, \qquad \forall w, \tag{16}$$

$$\sum_{w \in [2]} W_{wl} = 1, \qquad \forall l, \tag{17}$$

$$C_{jk} \geq 0 \qquad \forall j, k. \tag{18}$$

- Uses dichotomous constraints (binary variables P_{ij}) for ordering the jobs.
- Extended to include a double worker assignment.

Heuristics for flow shop insertion problems

- Using an iterated greedy algorithm (Ruiz and Stützle 2007).
  - Construct an initial solution by the procedure of Nawaz, Enscore, and Ham (1983).
  - Repeatedly perturb the solution and apply a local search.
  - New solution is accepted with

$$P[\text{accept}(\pi, \pi')] = \min\{e^{-\Delta(\pi,\pi')/T}, 1\}$$
$$T = \alpha\overline{p}/10$$

- Two strategies for worker assignment
  - Allocation to every possible machine.
  - Pooled allocation.

ITERATED GREEDY ALGORITHM

Input: A permutation schedule π.
Output: An improved permutation schedule π'.

function $\textsc{IGA}(\pi)$
  $\pi :=$ shift-localsearch(π)
  repeat
    remove d random jobs $j_1, \ldots, j_d$ from π to get π'
    for $i \in [d]$ **do**
      insert j_i into π' at the pos. of minimal $C_{\max}(\pi')$
    end for
    $\pi' :=$ shift-localsearch(π')
    if $\mathrm{accept}(\pi, \pi')$ **then**
      $\pi := \pi'$
    end if
  until some stopping criterion is satisfied
  return the best solution π^* found during the search
end function

Input: A permutation schedule π.
Output: An improved permutation schedule π'.

function $\textsc{IGA}(\pi)$
  $\pi :=$ shift-localsearch(π)
  repeat
    Perturbation
    remove d random jobs $j_1, \dots, j_d$ from π to get π'
    for $i \in [d]$ **do**
      insert j_i into π' at the pos. of minimal $C_{\max}(\pi')$
    end for
    $\pi' :=$ shift-localsearch(π')
    if $\mathrm{accept}(\pi, \pi')$ **then**
      $\pi := \pi'$
    end if
  until some stopping criterion is satisfied
  return the best solution π^* found during the search
end function

ITERATED GREEDY ALGORITHM

Input: A permutation schedule π.
Output: An improved permutation schedule π'.

```
function IGA(π)
    π := shift-localsearch(π)
    repeat                                                   Perturbation
        remove d random jobs j_1, ..., j_d from π to get π'
        for i ∈ [d] do
            insert j_i into π' at the pos. of minimal C_max(π')
        end for                                              Local search
        π' := shift-localsearch(π')
        if accept(π, π') then
            π := π'
        end if
    until some stopping criterion is satisfied
    return the best solution π* found during the search
end function
```

Output: A solution (π, k) for the PFSISP or HPFSISP

1: $P := \{(\mathrm{NEH}(k), k) \mid k \in [m]\}$ ▷ create the solution pool
2: **while** $|P| > 1$ **do**
3: **for all** $(\pi, k) \in P$ **do**
4: $(\pi, k) := (\mathrm{IGA}(\pi, t), k)$
5: **end for**
6: $(\pi_0, k_0) := \mathrm{argmax}_{(\pi,k)\in P}\, C_{\max}(\pi)$
7: $P := P \setminus \{(\pi_0, k_0)\}$
8: **end while**
9: **return** the single solution (π, k) in the pool

A POOLED IGA

Output: A solution (π, k) for the PFSISP or HPFSISP

```
P := {(NEH(k), k) | k ∈ [m]}                    ▷ create the solution pool
while |P| > 1 do
    for all (π, k) ∈ P do
        (π, k) := (IGA(π, t), k)
    end for
    (π_0, k_0) := argmax_{(π,k)∈P} C_max(π)
    P := P \ {(π_0, k_0)}
end while
return the single solution (π, k) in the pool
```

Apply IGA to pool

A POOLED IGA

Output: A solution (π, k) for the PFSISP or HPFSISP

```
P := {(NEH(k), k) | k ∈ [m]}                      ▷ create the solution pool
while |P| > 1 do                     Apply IGA to pool
    for all (π, k) ∈ P do
        (π, k) := (IGA(π, t), k)
    end for                          Remove worst solution
    (π_0, k_0) := argmax_{(π,k)∈P} C_max(π)
    P := P \ {(π_0, k_0)}
end while
return the single solution (π, k) in the pool
```

- Reduction to a head-body-tail problem

$$r_{ij} = \max\{r_{i,\pi(\pi^{-1}(j)-1)}, r_{i-1,j}\} + p_{ij}$$
$$q_{ij} = \max\{q_{i,\pi(\pi^{-1}(j)+1)}, q_{i+1,j}\} + p_{ij}$$

- Solution by dynamic programming

$$\begin{aligned} C(j,t_1,t_2) = \min\{ &\max\{C_1(t_1,j) + q_j, C(j+1, C_1(t_1,j), t_2)\}, \\ &\max\{C_2(t_2,j) + q_j, C(j+1, t_1, C_2(t_2,j))\}\} \end{aligned}$$

with earliest starting time

$$C_l(t,j) = \max\{t, r_j\} + p_{jl}$$

on machine l.

Computational experiments

TEST INSTANCES AND METHODOLOGY

- Nine small instances from Carlier (1978).
- 60 large instances from Taillard (1993) with up to 50 jobs and 20 machines.
- We created heterogeneous instances with
  - 0%, 10%, and 20% of incompatibilities per worker;
  - processing times chosen uniformly at random in $[p, 2p]$ or $[p, 5p]$ for regular time p.
- In total 408 test instances.
- Parameter setting according to Ruiz and Stützle (2007)

$$d = 4; \qquad \alpha = 0.4.$$

- Running time $3nm\,\mathrm{ms}$.
- Five replications in all tests.

CARLIER INSTANCES – SINGLE WORKER

		CPLEX		LOMPEN		Heuristics		
Var.	Inc.	$\bar{t}$	Rd.	$\bar{t}$	Rd.	S	P	PL
2	0	26.7	7.4	0.1	7.4	7.4	7.4	7.4
2	10	17.8	7.9	0.1	7.9	7.9	7.9	7.9
2	20	14.5	9.2	0.1	9.2	9.2	9.3	9.3
5	0	55.7	75.8	0.0	75.8	75.8	75.8	75.8
5	10	46.7	75.8	0.0	75.8	75.8	75.8	75.8
5	20	11.3	77.7	0.0	77.7	77.7	77.7	77.7
Avg.		28.8	42.3	0.0	42.3	42.3	42.3	42.3

- All instances solved optimally.
- Easy to solve for the state-of-the-art B&B solver LOMPEN (Companys and Mateo 2007) and the heuristics.
- Confirms a high overhead for a single worker.

CARLIER INSTANCES – TWO WORKERS

		CPLEX			Heuristics			
Var.	Inc.	Gap	$\bar{t}$	Rd.	S	P	PL	PD
2	0	6.9	1499.5	-4.2	-4.1	-4.0	-4.0	-3.6
2	10	6.4	1151.0	-2.2	-2.1	-2.1	-2.1	-2.1
2	20	5.1	1524.7	-0.6	-0.5	-0.5	-0.5	-0.4
5	0	4.1	875.8	3.6	4.6	4.3	4.3	4.6
5	10	3.7	899.0	5.0	5.8	5.5	5.6	5.5
5	20	3.7	788.2	5.4	6.1	5.9	5.8	5.7
Avg.		5.0	1123.0	1.2	1.6	1.5	1.5	1.6

- 80% of the instances solved in one hour.
- Heuristics in average 0.4% longer in $1/500$ of the time.
- Makespan close to optimum of regular PFSSP.

TAILLARD INSTANCES – TIME VAR. $[p, 2p]$

		LOMPEN		One worker			Two workers		
G	I	$\bar{t}$	Rd.	S	P	PL	S	P	PL
1	0	0.4	14.7	14.7	14.7	14.7	-3.4	-3.6	-3.7
2		3141.9	2.5	3.6	3.7	3.6	-0.3	-0.9	-1.3
3		54258.4	1.2	1.4	1.4	1.3	-0.0	-0.4	-0.9
4		6.9	27.4	27.4	27.4	27.4	-0.8	-1.3	-1.3
5		127.0	19.2	19.4	19.4	19.4	3.0	0.7	-0.4
6		21934.4	5.2	5.1	5.0	4.2	4.1	3.0	0.8
Avg.		13244.9	11.7	11.9	11.9	11.8	0.4	-0.4	-1.1
1	10	0.4	17.0	17.0	17.0	17.0	-3.5	-3.6	-3.7
2		3795.5	3.5	3.7	3.7	3.7	-0.3	-0.7	-1.2
3		48653.9	1.3	1.5	1.5	1.4	-0.1	-0.2	-0.9
4		6.2	28.4	28.4	28.4	28.4	-0.3	-0.9	-1.0
5		80.4	20.2	20.2	20.2	20.2	3.2	0.9	-0.3
6		19483.3	5.5	5.3	5.3	4.7	4.1	3.1	0.8
Avg.		12003.3	12.6	12.7	12.7	12.6	0.5	-0.2	-1.0
1	20	0.4	18.3	18.3	18.3	18.3	-1.7	-1.9	-2.0
2		1872.0	5.2	5.3	5.5	5.3	-0.1	-0.6	-0.9
3		44304.2	1.3	1.4	1.5	1.4	0.0	-0.2	-0.8
4		4.7	28.7	28.7	28.7	28.7	-0.0	-0.7	-0.7
5		76.4	20.5	20.5	20.6	20.6	3.2	1.3	0.2
6		16410.8	5.8	5.8	5.6	5.0	4.1	3.1	0.8
Avg.		10444.8	13.3	13.3	13.4	13.2	0.9	0.2	-0.6

TAILLARD INSTANCES – TIME VAR. $[p, 2p]$

		LOMPEN		One worker			Two workers		
G	I	$\bar{t}$	Rd.	S	P	PL	S	P	PL
1	0	0.4	14.7	14.7	14.7	14.7	-3.4	-3.6	-3.7
2	0	3141.9	2.5	3.6	3.7	3.6	-0.3	-0.9	-1.3
3	0	54258.4	1.2	1.4	1.4	1.3	-0.0	-0.4	-0.9
4	0	6.9	27.4	27.4	27.4	27.4	-0.8	-1.3	-1.3
5	0	127.0	19.2	19.4	19.4	19.4	3.0	0.7	-0.4
6	0	21934.4	5.2	5.1	5.0	4.2	4.1	3.0	0.8
Avg.		[illegible]	11.7	11.9	11.9	[illegible]	0.4	-0.4	-1.1
1	10	0.4	17.0	17.0	17.0	17.0	-3.5	-3.6	-3.7
2	10	3795.5	3.5	3.7	3.7	3.7	-0.3	-0.7	-1.2
3	10	48653.9	1.3	1.5	1.5	1.4	-0.1	-0.2	-0.9
4	10	6.2	28.4	28.4	28.4	28.4	0.3	[illegible]	1.0
5	10	80.4	20.2	20.2	20.2	20.2	3.2	0.9	-0.3
6	10	19483.3	5.5	5.3	5.3	4.7	4.1	3.1	0.8
Avg.		12003.3	12.6	12.7	12.7	12.6	0.5	-0.2	-1.0
1	20	0.4	18.3	18.3	18.3	18.3	-1.7	-1.9	-2.0
2	20	1872.0	5.2	5.3	5.5	5.3	-0.1	-0.6	-0.9
3	20	44304.2	1.3	1.4	1.5	1.4	0.0	-0.2	-0.8
4	20	4.7	28.7	28.7	28.7	28.7	-0.0	-0.7	-0.7
5	20	76.4	20.5	20.5	20.6	20.6	3.2	1.3	0.2
6	20	16410.8	5.8	5.8	5.6	5.0	4.1	3.1	0.8
Avg.		10444.8	13.3	13.3	13.4	13.2	0.9	0.2	-0.6

- Single worker
  - Very good solutions, close to optimality.
  - Solver needs $3\,\mathrm{h}$, heuristic at most $1\,\mathrm{min}$.
  - No notable difference between the strategies.
- Two workers: pool and longer time limit improve.

TAILLARD INSTANCES – TIME VAR. $[p, 5p]$

		LOMPEN		One worker			Two workers		
G	I	$\bar{t}$	Rd.	S	P	PL	S	P	PL
1	0	0.2	106.5	106.5	106.5	106.5	6.7	5.8	5.3
2		3.7	58.0	58.0	58.0	58.0	5.5	4.1	3.7
3		390.8	24.6	24.6	24.7	24.7	3.7	2.8	2.2
4		6.8	149.2	149.2	149.2	149.2	5.6	2.8	2.3
5		30.8	125.5	125.5	125.5	125.5	9.3	5.7	4.0
6		491.5	77.0	77.0	77.0	77.0	7.8	6.2	3.5
Avg.		154.0	90.1	90.1	90.2	90.2	6.4	4.6	3.5
1	10	0.2	108.3	108.3	108.4	108.4	6.6	5.8	5.4
2		3.6	58.0	58.0	58.0	58.0	5.5	4.4	3.6
3		389.5	26.6	26.6	26.6	26.6	3.6	2.7	2.1
4		5.8	154.7	154.7	154.7	154.7	5.2	3.0	2.4
5		27.5	125.5	125.5	125.6	125.5	9.6	6.6	4.3
6		476.9	77.8	77.8	77.8	77.8	8.0	6.4	3.4
Avg.		150.6	91.8	91.8	91.8	91.8	6.4	4.8	3.6
1	20	0.1	119.1	119.1	119.1	119.1	6.6	5.8	5.4
2		3.4	68.2	68.2	68.2	68.2	5.4	4.3	3.7
3		383.6	27.6	27.6	27.6	27.6	3.9	3.4	2.6
4		4.4	154.7	154.7	154.7	154.7	10.9	7.9	7.0
5		24.1	126.0	126.0	126.0	126.0	9.4	6.6	4.8
6		462.1	78.5	78.5	78.5	78.5	7.9	6.4	3.5
Avg.		146.3	95.7	95.7	95.7	95.7	7.3	5.7	4.5

TAILLARD INSTANCES – TIME VAR. $[p, 5p]$

		LOMPEN		One worker			Two workers		
G	I	$\bar{t}$	Rd.	S	P	PL	S	P	PL
1		0.2	106.5	106.5	106.5	106.5	6.7	5.8	5.3
2		3.7	58.0	58.0	58.0	58.0	5.5	4.1	3.7
3	0	390.8	24.6	24.6	24.7	24.7	3.7	2.8	2.2
4		6.8	149.2	149.2	149.2	149.2	5.6	2.8	2.3
5		[illegible]	125.5	125.5	125.5	125.5	9.3	5.7	4.0
6		491.5	77.0	77.0	77.0	77.0	7.8	6.2	3.5
Avg.		154.0	90.1	90.1	90.2	90.2	6.4	4.6	3.5
1		0.2	108.3	108.3	108.4	108.4	6.6	5.8	5.4
2		5.6	58.0	58.0	58.0	58.0	5.5	4.4	3.0
3	[illegible]	389.5	26.6	26.6	26.6	26.6	3.6	2.7	2.1
4		5.8	154.7	154.7	154.7	154.7	5.2	3.0	2.4
5		[illegible]	125.5	125.5	125.6	125.5	9.6	6.6	4.3
6		476.9	77.8	77.8	77.8	77.8	8.0	6.4	3.4
Avg.		150.6	91.8	91.8	91.8	91.8	6.4	4.8	3.6
1		0.1	119.1	119.1	119.1	119.1	6.6	5.8	5.4
2		3.4	68.2	68.2	68.2	68.2	5.4	4.3	3.7
3	20	383.6	27.6	27.6	27.6	27.6	3.9	3.4	2.6
4		4.4	154.7	154.7	154.7	154.7	10.9	7.9	7.0
5		24.1	126.0	126.0	126.0	126.0	9.4	6.6	4.8
6		462.1	78.5	78.5	78.5	78.5	7.9	6.4	3.5
Avg.		146.3	95.7	95.7	95.7	95.7	7.3	5.7	4.5

- Single worker
  - Again: good solutions in a short time with all strategies.
  - Easier to solve, due to strong bottleneck machine.
- Two workers: pool and longer time limit again improve.

CONCLUSIONS

- Variants of IGAs show very good results for PFSISP and HPFSISP.
- The pooling strategy is useful for the hybrid problem variant.
- Inserting a single worker has a visible overhead
  - 12% for a small time variation $[p, 2p]$;
  - 90% for a large time variation $[p, 5p]$.
- Inserting two workers at a hybrid machine effectively hides disabilities
  - Shorter makespan than PFSSP for small time variation $[p, 2p]$
  - Never more than 7% longer for large time variation $[p, 5p]$

Thanks for your attention!

For more on the general case:

European Journal of Operational Research 237 (2014) 713–720

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Application of O.R.

Flow shop scheduling with heterogeneous workers

Alexander J. Benavides [a], Marcus Ritt [a,*], Cristóbal Miralles [b]

[a] Instituto de Informática, Universidade Federal do Rio Grande do Sul, Brazil
[b] ROGLE, Departamento de Organización de Empresas, Universitat Politècnica de València, Spain

REFERENCES I

Carlier, J. (1978). "Ordonnancements a contraintes disjonctives". In: *R.A.I.R.O. Recherche operationelle/Operations Research* 12.4, pp. 333–351.

Companys, Ramón and Manuel Mateo (2007). "Different behaviour of a double branch-and-bound algorithm on $Fm \mid prmu \mid C_{\max}$ and $Fm \mid block \mid C_{\max}$ problems". In: *Comput. Oper. Res.* 34.4, pp. 938–953. DOI: 10.1016/j.cor.2005.05.018.

Garey, Michael R. and David S. Johnson (1979). *Computers and intractability: A guide to the theory of NP-completeness*. Freeman.

Nawaz, M., E.E. Enscore, and I. Ham (1983). "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem". In: *Omega* 11.1, pp. 91–95. DOI: 10.1016/0305-0483(83)90088-9.

REFERENCES II

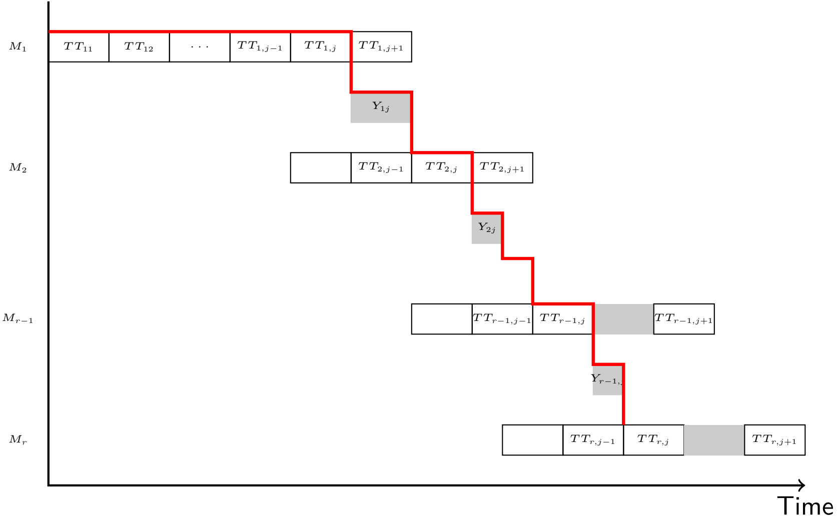
Ruiz, Rubén and Thomas Stützle (2007). "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem". In: *Eur. J. Oper. Res.* 177.3, pp. 2033–2049. DOI: 10.1016/j.ejor.2005.12.009.

Taillard, E. (1993). "Benchmarks for basic scheduling problems". In: *Eur. J. Oper. Res.* 64.2, pp. 278–285. DOI: 10.1016/0377-2217(93)90182-M.

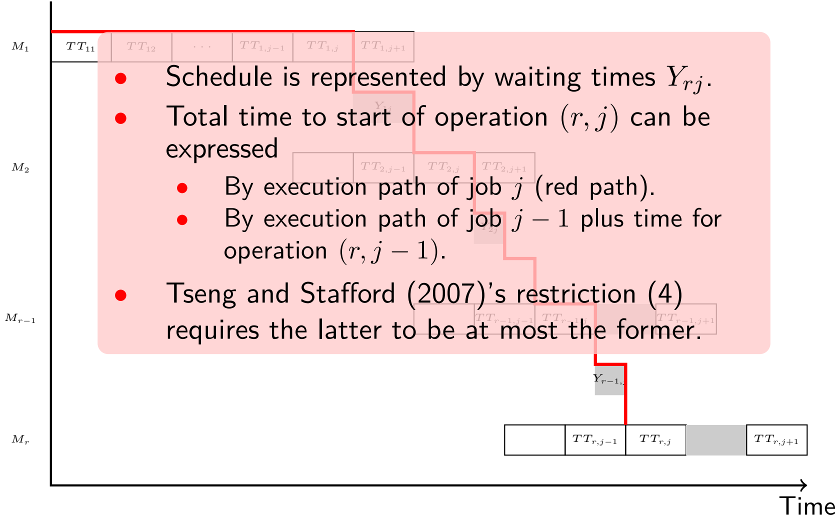
Tseng, F T and E F Stafford (2007). "New MILP models for the permutation flowshop problem". In: *J. Oper. Res. Soc.* 59.10, pp. 1373–1386. ISSN: 0160-5682. DOI: 10.1057/palgrave.jors.2602455.

World Health Organization (2011). *World report on disability*.

MODELING DEPENDENCIES AFTER Tseng and Stafford (2007)



MODELING DEPENDENCIES AFTER Tseng and Stafford (2007)



- Schedule is represented by waiting times Y_{rj}.
- Total time to start of operation (r, j) can be expressed
  - By execution path of job j (red path).
  - By execution path of job $j-1$ plus time for operation $(r, j-1)$.
- Tseng and Stafford (2007)'s restriction (4) requires the latter to be at most the former.