Solving 2-stage Hybrid Flow Shop using Climbing Depth-bounded Discrepancy Search

Abir BEN HMIDA SAKLY[1,2], Mohamed HAOUARI[2]
Marie-José HUGUET[1], Pierre LOPEZ[1]

[1] LAAS-CNRS, Université de Toulouse, France
[2] ROI, Ecole Polytechnique de Tunisie, Tunisia

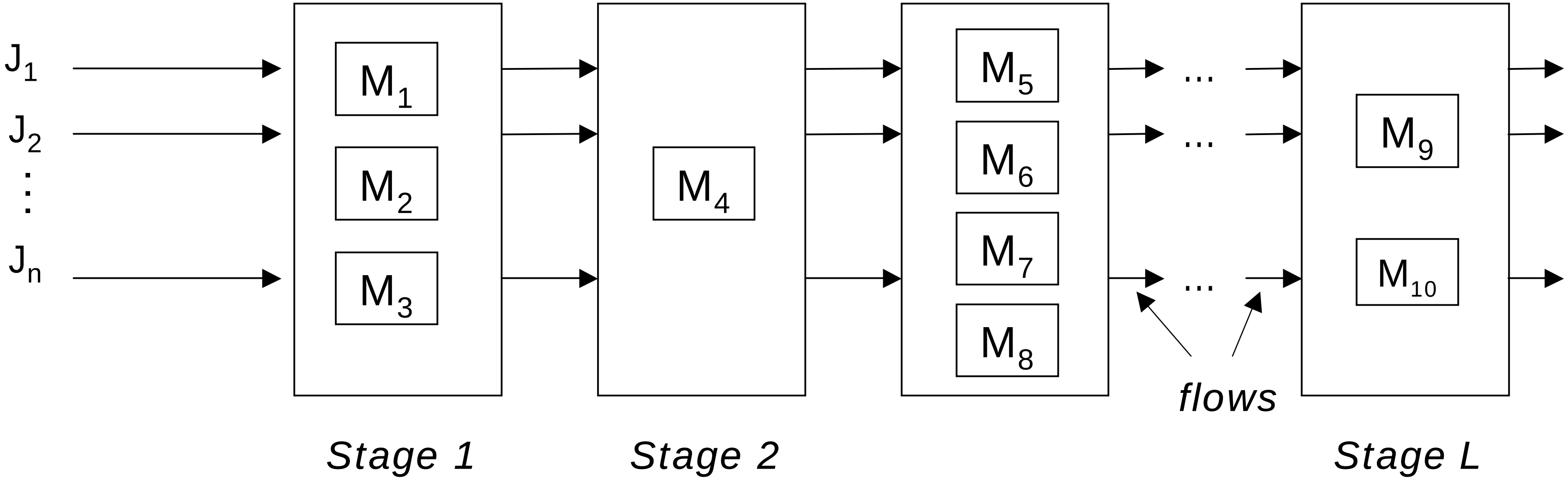
OUTLINE

- Objectives
- 2-stage Hybrid Flow Shop Problem
- Discrepancy-based search methods
- Proposed method: CDDS
- Experimental evaluation and results
- Conclusions

Objectives

- To solve 2-stage Hybrid Flow Shop ("2-HFS")
- Adapting discrepancy-based search methods for the problem under study
- Developing associated algorithms
- Experimental evaluation of the propositions

Hybrid Flow Shop



- $J=\{J_1,\ldots,J_i,\ldots,J_n\}$ jobs
- $E=\{1,\ldots,s,\ldots, L\}$ stages
- m_s: identical parallel machines / $\max(m_s) > 1$
- Application: semiconductors

  (Printed Circuit Boards)

Hybrid Flow Shop

- As in a Flow Shop:
  - Resources = machines
  - Job = sequence of operations
- But:
  - Alternative machines can process an operation
- Double problem:
  - Select a machine for each operation
  - Determine a start time for each operation
- 2 stages $\Rightarrow$ 2-HFS
- 2-HFS | max(m_1,m_2) > 1 | C_{max} is strongly NP-hard

2-stage Hybrid Flow Shop

- Background
  - 2-HFS with max(m_s)=2 for $s \in \{1,2\}$ is NP-hard: [Gupta, 1988]
  - Exact methods: [Brah & Hunsucker, 1991]; [Portmann *et al*., 1992]; [Moursli & Pochet, 2000]; [Carlier & Néron, 2000]; [Lin & Liao, 2003]; [Haouari *et al.*,2006]
  - Lower bounds: [Santos *et al.*, 1995]; [Moursli & Pochet, 2000]; [Carlier & Néron, 2000]; [Haouari and M'Hallah,1997]
  - Heuristics: [Brah & Loo, 1999]; [Engin & Döyen, 2004]; [Haouari and M'Hallah, 1997]

Discrepancy-based search methods (1)

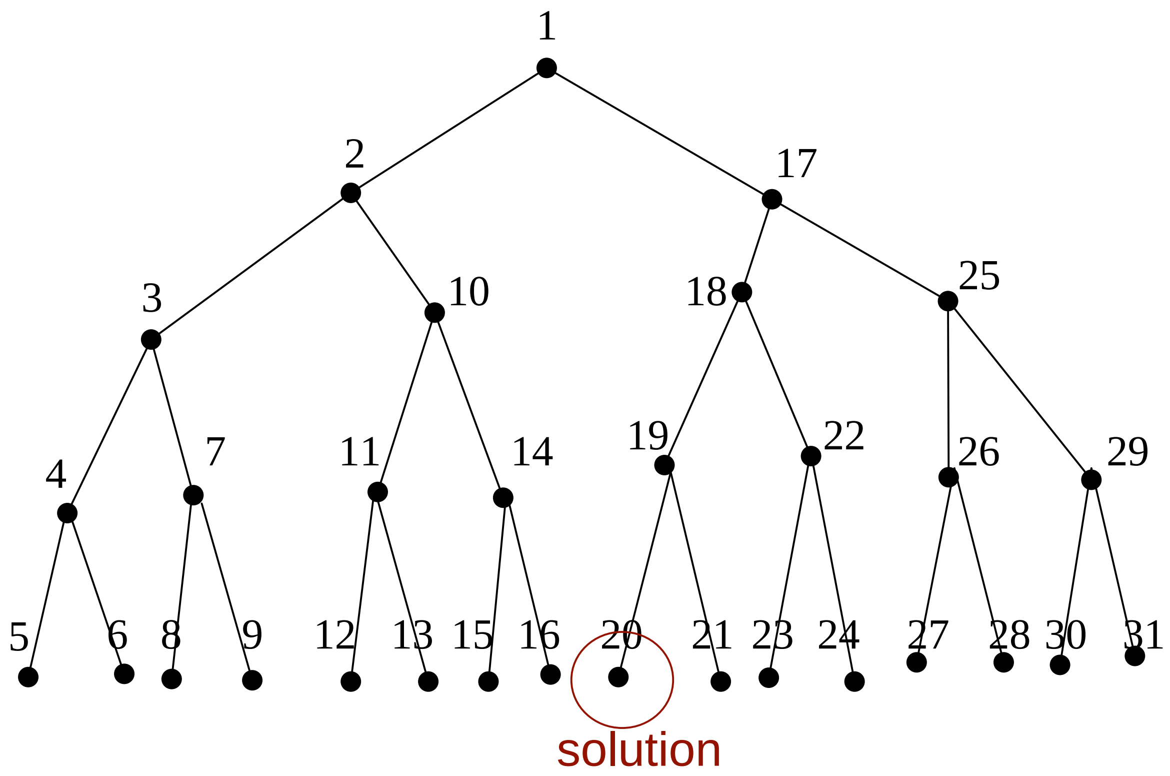
- Limited Discrepancy Search - LDS [Harvey & Ginsberg, 1995]
  - Is a problem satisfiable? → *Satisfaction*
  - Iterative tree search method
  - Instantiation heuristic to guide the search

    (the initial global instantiation is not necessarily a solution)
  - When the heuristic does not find a good solution, it is probably because it made a few poor choices → *discrepancy* then makes a choice different than heuristically top-ranked
  - Hope to find a solution before *Depth-First Search*

Depth First Search (DFS)

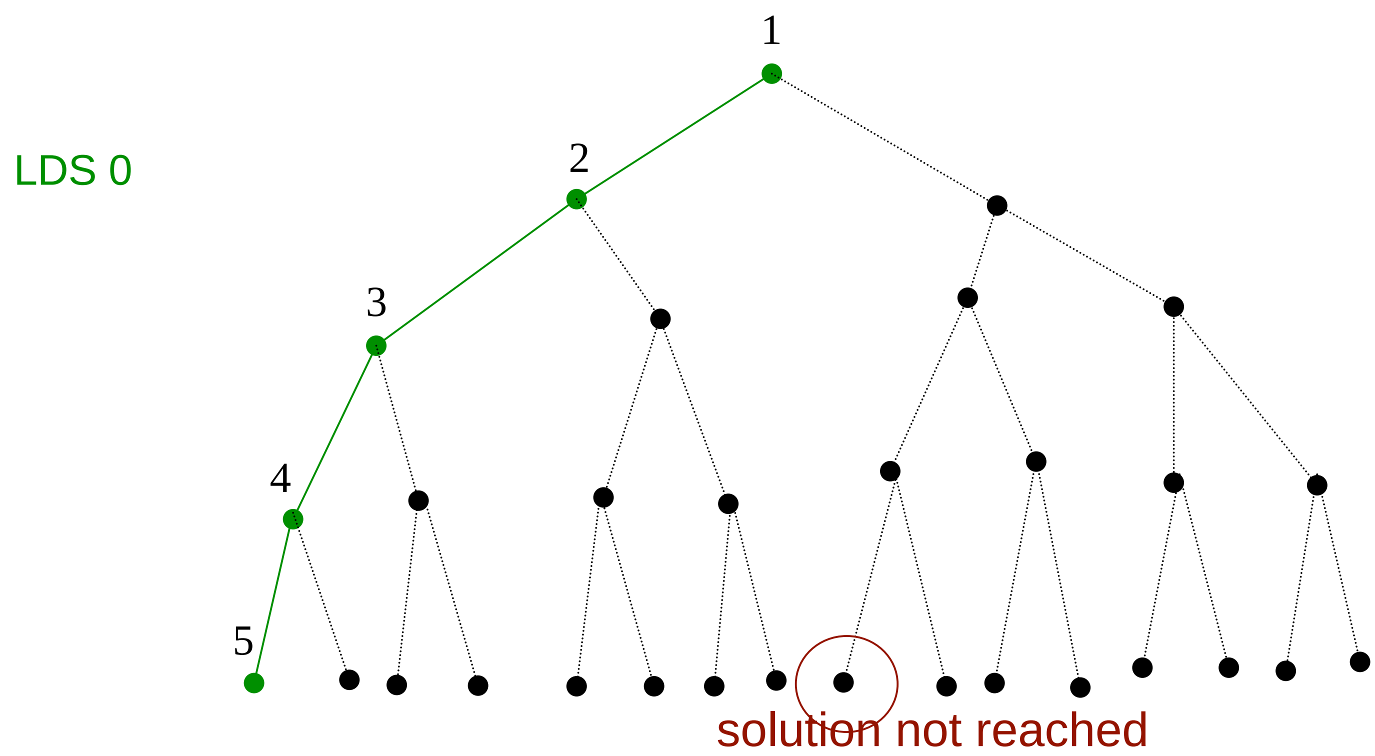
- Search principle
  - Example: binary tree

DFS



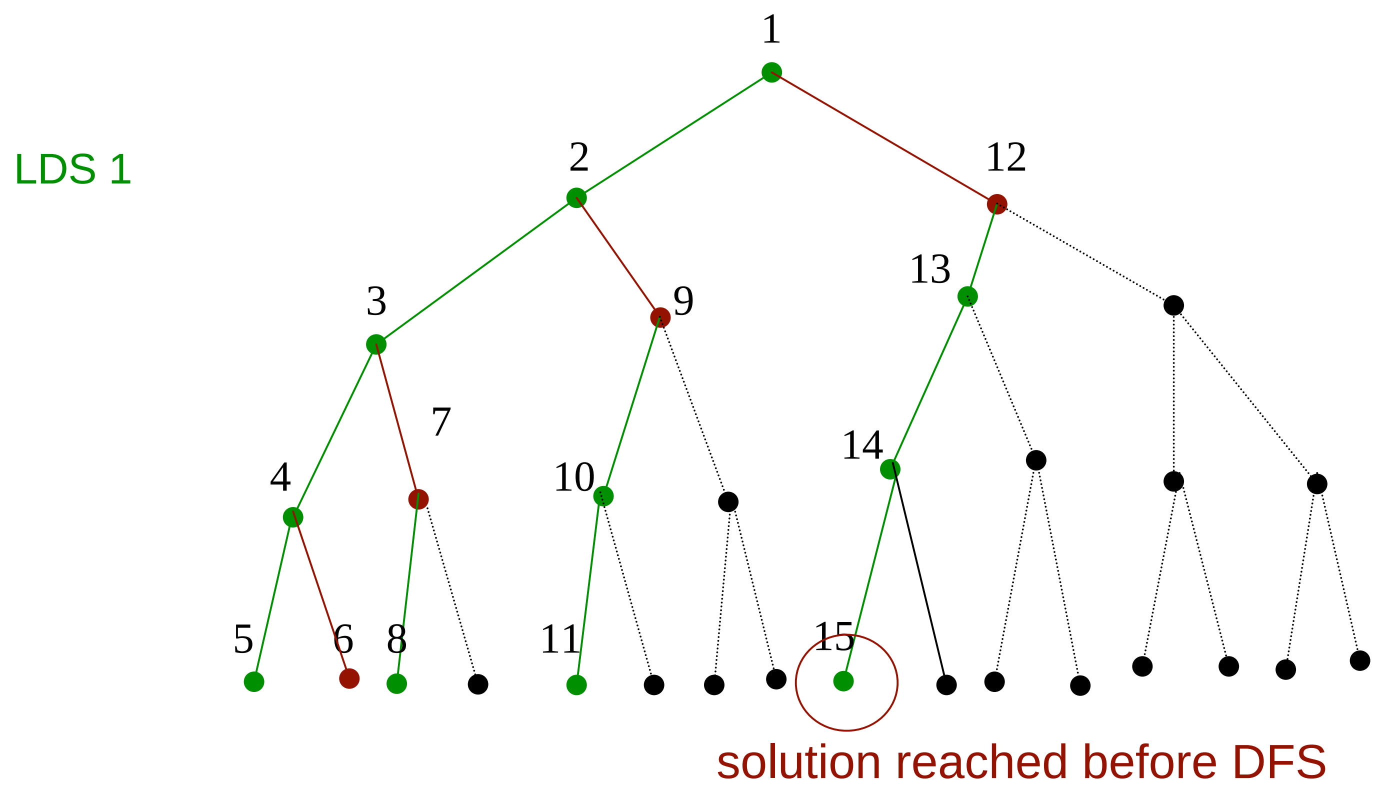
Limited Discrepancy Search (LDS)

- Search principle
  - Example: binary tree
  - LDS 0: The choices of the heuristic are satisfied



Limited Discrepancy Search (LDS)

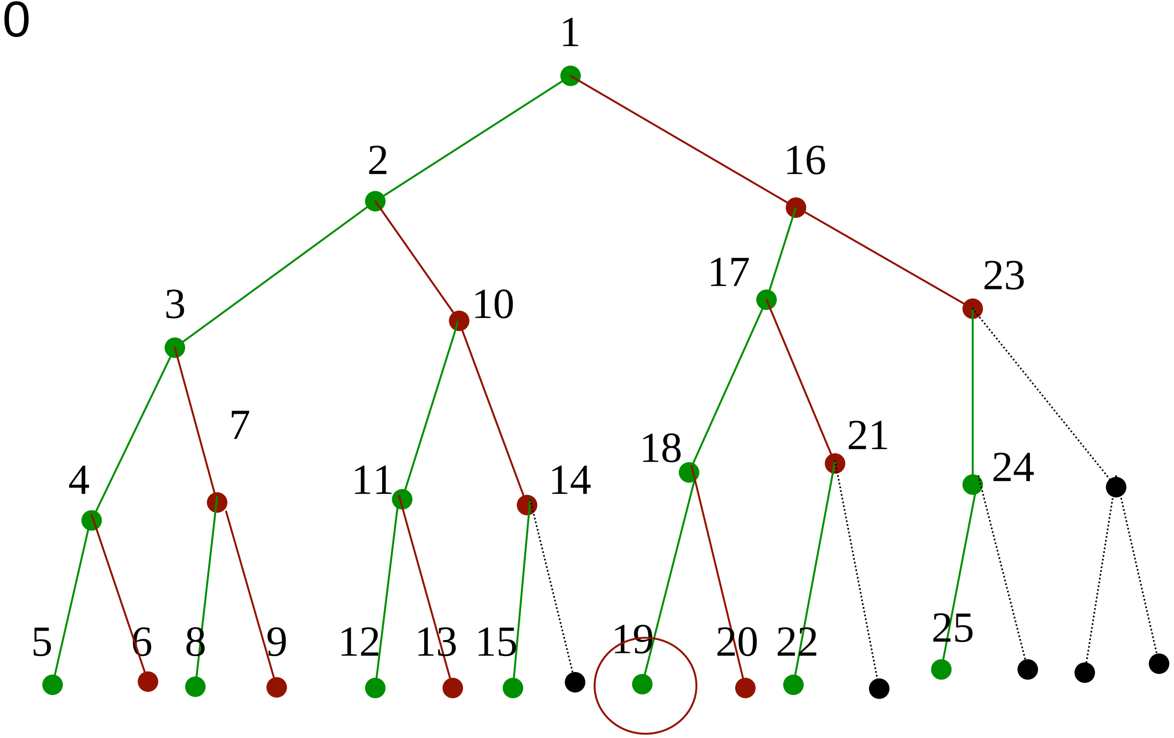
- Search principle
  - Example: binary tree
  - LDS 1: All the paths differ of ONE decision from LDS 0



Limited Discrepancy Search (LDS)

- Search principle
  - Example: binary tree
  - LDS 2: All the paths differ of TWO decisions (for TWO variables) from LDS 0

LDS 2



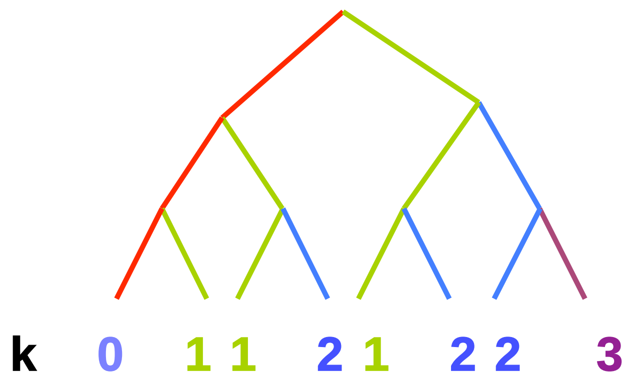
solution reached before DFS

(Improved) Limited Discrepancy Search (I-LDS)

[Harvey & Ginsberg, 1995] and [Korf, 1996]

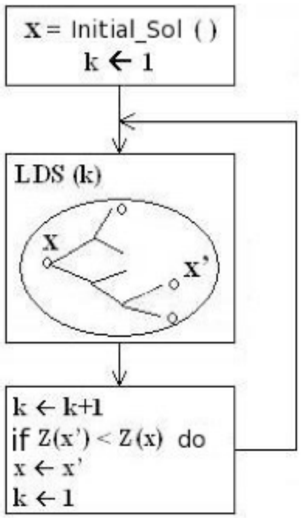
Algorithm

```
k ← 0
kmax ← N
I ← Initial_instantiation()
While no_solution() and (k ≤ kmax) do
     k ← k+1
     -- Generate leaves at discrepancy k from I
     -- Stop when a solution is found
     I ← compute_Leaves (I, k)
End while
```



Discrepancy-based search methods (2)

- Depth-bounded Discrepancy Search - DDS [Walsh, 1997]
  - → *Satisfaction*
  - To correct “early mistakes” (the most important)
  - Principle: discrepancies on top of the search tree
  - Stop: a solution is found
- Climbing Discrepancy Search - CDS [Milano & Roli, 2005]
  - To improve current solution → *Optimization*
  - Principle:
    - `Initial solution (Reference)`
    - `Apply LDS principle to explore the neighborhood from this reference`
    - `Reference ← Improved_Solution`
    - `Restart with #discrepancy := 0`
  - Stop: no more improvement, limit on time or #iterations reached
  - CDS is close to VNS [Hansen & Mladenovic, 2001]



Proposed method

- To combine 2 discrepancy-based methods
  - Climbing DS (neighborhood search)
  - Depth-bounded DS (neighborhood restricted at the top of the tree)

  → *Climbing Depth-bounded Discrepancy Search (CDDS)*

- Optimization method: approximate solutions
  - Criterion = *makespan* minimization
  - A solution = UB

$$\min C_{\max}$$

- LBs to fathom nodes in the search tree

Proposed method

- Exploration strategy
  - Instantiation heuristics
    - First Stage:
      1. Job Selection : Extension of Johnson's rule for the 2-machine FS having $\{ p_{1,j} / m_1 , p_{2,j} / m_2 \}$ as processing times
      2. Machine selection (allocation): First Available machine (FAM)
    - Second Stage:
      1. Job Selection : Earliest release date (and longest $p_{2,j}$)
      2. Machine selection (allocation): First Available machine (FAM)
  - Simple propagation
    - Forward-Checking over:
      - Earliest start time of subsequent operations
      - Release date of selected machine
  - Discrepancies
    - On job selection variables

Proposed method

- Lower bounds: SPT-rule [Haouari and M'Hallah, 1997]

1. On second stage:

$$LB_{SPT}^{2}(S)=\left\lceil \frac{I_2(S)+\sum_{j\in S} p_{2,j}}{m_2} \right\rceil$$

2. On first stage :

$$LB_{SPT}^{1}(S)=\left\lceil \frac{I_1(S)+\sum_{j\in S} p_{1,j}}{m_1} \right\rceil$$

3. So,

$$LB=\max\left(LB_{SPT}^{1},LB_{SPT}^{2}\right)$$

Experiments

- Instances:
  - Three sets of instances generated in a similar way as in [Lee and Vairaktarakis, 1994]:
  - Set A:
    - n ={10, 20, 30, 40, 50, 100, 150}
    - M: (m_1,m_2) are (2, 2), (2, 4), (4, 2), and (4, 4)
    - processing times: uniform distribution in [1, 20] for the first stage and in [1, 40] for the second stage
    - 560 instances
  - Set B:
    - processing times: uniform distribution in [1, 40] for the first stage and in [1, 20] for the second stage
  - Set C:
    - processing times: uniform distribution in [1, 40] for both stages
  - 1680 instances

Experiments

- Comparison of *CDDS*[2] (2-HFS) with:
  - *CDDS*[L] of [Ben Hmida *et al.*, 2007] (HFS)
  - LBs of [Haouari *et al.*, 2006]
  - TS of [Haouari and M'Hallah, 1997]

$$\text{Dev} = \frac{C\max\_best - LowerBound}{LowerBound} \times 100$$

- Stop:
  - limit on CPU time=15 sec.

- Instance and its reverse are considered

Results

Table 1. Performance on Set A

		CDDS^2			*CDDS*^{L}		
n	(m_1, m_2)	US	Dev	Time	US	Dev	Time
10	(2, 2)	3	0.21	3.62	11	1.20	8.79
	(2, 4)	10	2.47	5.28	17	6.24	12.83
	(4, 2)	0	0.00	0.07	3	0.06	2.31
	(4, 4)	0	0.00	0.13	4	0.12	3.10
20	(2, 2)	2	0.05	2.03	3	0.16	3.11
	(2, 4)	9	0.91	8.42	20	5.79	15.00
	(4, 2)	0	0.00	0.94	4	0.12	3.75
	(4, 4)	4	0.21	4.60	11	1.34	8.77
30	(2, 2)	1	0.02	0.92	7	1.61	5.85
	(2, 4)	6	0.86	6.89	17	5.61	12.92
	(4, 2)	0	0.00	0.45	3	0.12	2.63
	(4, 4)	3	0.07	3.78	8	0.32	6.76
40	(2, 2)	0	0.00	0.24	2	0.04	1.72
	(2, 4)	4	0.21	5.18	8	0.97	6.78
	(4, 2)	0	0.00	0.96	4	0.56	3.77
	(4, 4)	2	0.05	2.89	4	0.12	4.16
50	(2, 2)	0	0.00	0.40	3	0.07	2.59
	(2, 4)	2	0.15	2.37	6	0.70	5.33
	(4, 2)	0	0.00	0.60	4	0.11	3.48
	(4, 4)	2	0.06	3.91	4	0.08	4.56
100	(2, 2)	1	0.05	1.88	4	0.23	3.75
	(2, 4)	2	0.06	3.68	6	0.30	5.79
	(4, 2)	0	0.00	0.91	5	0.09	4.43
	(4, 4)	2	0.02	3.71	6	0.15	5.80
150	(2, 2)	0	0.00	4.40	3	0.30	5.99
	(2, 4)	0	0.00	2.41	2	0.16	3.67
	(4, 2)	0	0.00	5.27	3	0.18	6.73
	(4, 4)	1	0.01	8.10	2	0.05	8.79
Average		**54**	**0.19**	**3**	**174**	**0.96**	**5.83**

Results

Table 2. Performance on Set B

		$CDDS^2$			$CDDS^L$		
n	(m_1, m_2)	US	Dev	Time	US	Dev	Time
10	(2, 2)	2	0.09	1.51	5	0.12	4.32
	(2, 4)	0	0.00	0.44	5	0.38	4.08
	(4, 2)	9	1.60	6.03	19	5.92	14.28
	(4, 4)	2	0.29	1.52	7	1.18	5.74
20	(2, 2)	1	0.03	0.79	5	0.08	4.34
	(2, 4)	0	0.00	0.03	3	0.09	2.28
	(4, 2)	5	0.64	4.91	13	5.70	10.09
	(4, 4)	3	0.14	2.91	4	0.15	3.78
30	(2, 2)	0	0.00	0.03	4	0.16	3.02
	(2, 4)	0	0.00	0.10	3	0.12	2.34
	(4, 2)	7	0.93	6.05	11	5.69	8.64
	(4, 4)	2	0.11	2.49	3	0.19	3.31
40	(2, 2)	0	0.00	0.19	2	0.06	1.67
	(2, 4)	0	0.00	0.09	3	0.53	2.33
	(4, 2)	3	0.28	3.06	9	1.01	7.31
	(4, 4)	2	0.05	13.09	4	0.55	8.24
50	(2, 2)	0	0.00	0.65	3	0.56	2.80
	(2, 4)	0	0.00	0.10	2	0.43	1.59
	(4, 2)	8	0.37	7.23	10	0.69	7.95
	(4, 4)	1	0.02	2.94	3	0.06	4.75
100	(2, 2)	0	0.00	0.52	3	0.09	2.69
	(2, 4)	0	0.00	0.55	2	0.09	2.00
	(4, 2)	2	0.03	4.30	7	0.35	6.65
	(4, 4)	2	0.02	3.30	8	0.17	6.99
150	(2, 2)	0	0.00	1.39	3	0.08	3.43
	(2, 4)	0	0.00	0.90	4	0.13	3.72
	(4, 2)	1	0.03	3.03	5	0.22	6.02
	(4, 4)	1	0.01	5.06	3	0.03	6.55
Average		**51**	**0.17**	**2.62**	**153**	**0.89**	**5.03**

Results

Table 3. Performance on Set C

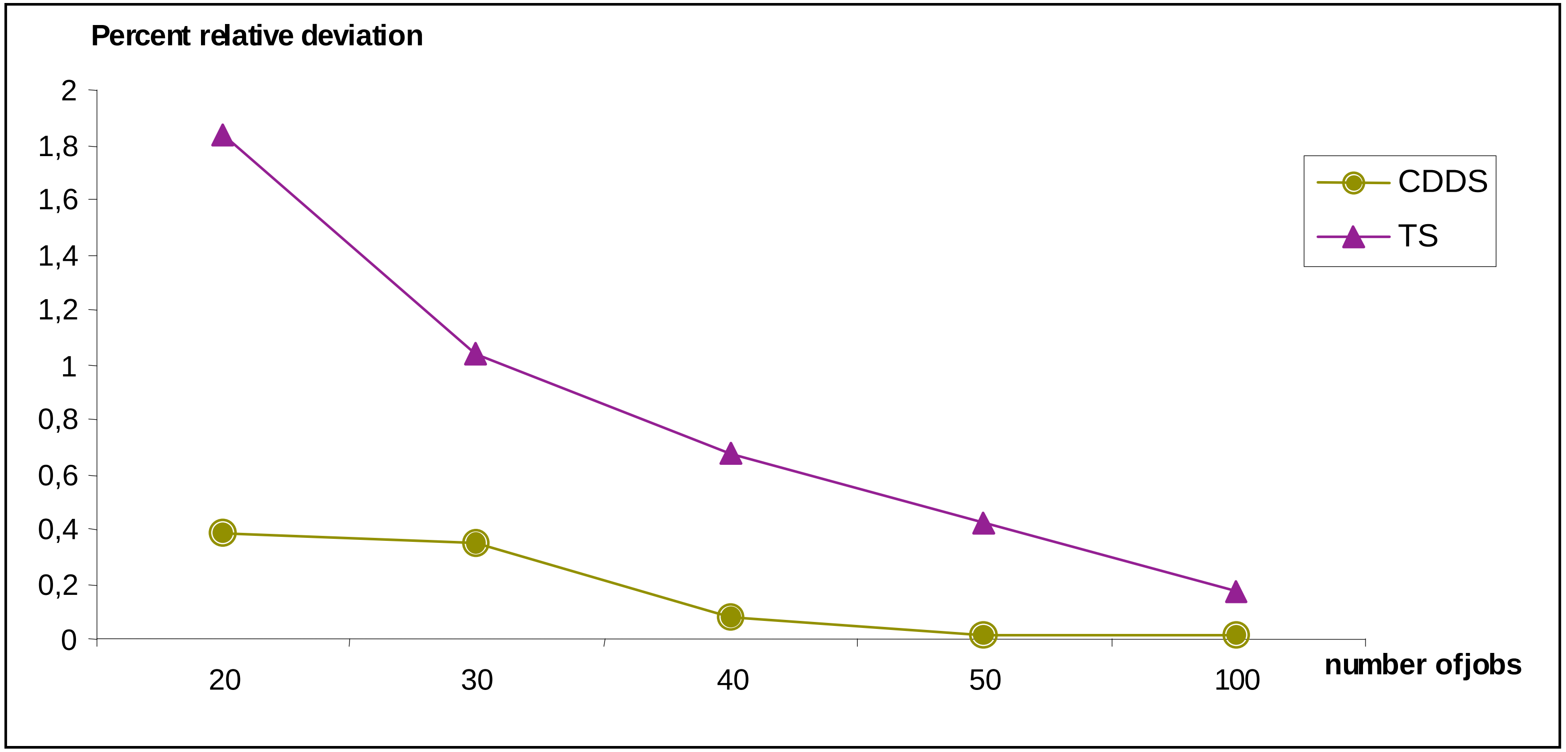
		CDDS^2			*CDDS*^{L}		
n	(m_1, m_2)	US	Dev	Time	US	Dev	Time
10	(2, 2)	3	0.20	2.35	9	0.26	7.18
	(2, 4)	1	0.34	0.81	6	0.45	5.07
	(4, 2)	0	0.00	0.06	5	0.05	3.80
	(4, 4)	9	1.89	6.75	18	1.98	13.58
20	(2, 2)	5	0.39	3.85	11	0.48	8.60
	(2, 4)	0	0.00	0.07	7	0.07	5.30
	(4, 2)	0	0.00	0.91	3	0.23	3.02
	(4, 4)	13	1.19	9.87	20	1.47	15.00
30	(2, 2)	4	0.10	3.07	12	0.63	9.31
	(2, 4)	0	0.00	0.44	8	0.08	6.26
	(4, 2)	1	0.02	1.18	5	0.05	4.64
	(4, 4)	9	1.42	9.10	14	1.95	10.80
40	(2, 2)	3	0.08	2.38	9	0.26	7.19
	(2, 4)	1	0.01	0.81	7	0.14	5.78
	(4, 2)	1	0.01	2.09	6	0.17	5.96
	(4, 4)	5	0.46	5.39	11	0.61	8.74
50	(2, 2)	0	0.00	0.19	6	0.06	4.63
	(2, 4)	0	0.00	0.26	8	0.28	6.16
	(4, 2)	2	0.02	1.88	5	0.05	4.46
	(4, 4)	5	0.84	4.89	12	1.20	9.39
100	(2, 2)	0	0.00	0.73	9	0.16	7.15
	(2, 4)	0	0.00	0.56	11	0.15	8.50
	(4, 2)	2	0.01	3.44	9	0.09	7.70
	(4, 4)	6	0.22	6.73	7	0.32	5.98
150	(2, 2)	1	0.01	4.28	8	0.08	8.57
	(2, 4)	0	0.00	1.62	9	0.09	7.64
	(4, 2)	0	0.00	5.04	7	0.07	8.53
	(4, 4)	7	0.11	9.69	10	0.14	8.19
Average		**78**	**0.26**	**3.16**	**252**	**0.41**	**7.40**

Results

Table 4. Performance comparison between
the proposed CDDS and TS [Haouari and M'Hallah, 1997]

n		***Performance on Set A***			***Performance on Set B***			***Performance on Set C***			***Average***
		(2, 4)	(4, 4)	(4, 2)	(2, 4)	(4, 4)	(4, 2)	(2, 4)	(4, 4)	(4, 2)	
20	*CDDS*	0.95	0.26	0.00	0.03	0.14	0.73	0.00	1.48	0.05	0.40
	TS	2.90	1.20	0.35	0.92	5.72	0.13	0.56	3.43	1.22	1.83
30	*CDDS*	**0.92**	**0.10**	**0.00**	**0.00**	**0.11**	**0.96**	**0.07**	**1.45**	**0.02**	**0.40**
	TS	1.43	0.85	0.06	0.57	3.10	0.05	0.27	1.45	1.46	1.03
40	*CDDS*	**0.21**	**0.05**	**0.00**	**0.00**	**0.05**	**0.28**	**0.02**	**0.46**	**0.01**	**0.12**
	TS	0.96	0.43	0.12	0.5	1.57	0.12	0.34	1.08	0.89	0.67
50	*CDDS*	**0.15**	**0.06**	**0.00**	**0.00**	**0.02**	**0.37**	**0.00**	**0.88**	**0.02**	**0.16**
	TS	0.54	0.30	0.02	0.26	1.09	0.04	0.20	0.95	0.42	0.42
100	*CDDS*	**0.06**	**0.02**	**0.00**	**0.00**	**0.02**	**0.03**	**0.00**	**0.22**	**0.01**	**0.04**
	TS	0.19	0.15	0.02	0.11	0.39	0.01	0.07	0.41	0.18	0.17
Average	*CDDS*	**0.46**	**0.10**	**0.00**	**0.01**	**0.07**	**0.48**	**0.02**	**0.90**	**0.02**	**0.22**
	TS	1.20	0.59	0.11	0.47	2.37	0.07	0.29	1.46	0.83	0.82

Results



Percent relative deviation
2
1,8
1,6
1,4
1,2
1
0,8
0,6
0,4
0,2
0
20
30
40
50
100
number ofjobs
CDDS
TS

Conclusions

- Novel method to solve the 2-HFS
- Promising results
  - outperforms the previous algorithm (*CDDS*^{L})
  - outperforms the TS procedure of [Haouari and M'Hallah, 1997]
  - produces optimal solutions for more than 89.2% (1498 instances out of 1680)
  - provides a tiny gap (≤ 0.22 if all instances are considered)

- Good results for solving Flexible Job Shop problems ([Ben Hmida *et al.*, 2009] submitted)

instances	#	*CDDS*	*GA*	*TS*	*hGA*
Brandimarte	10	15.0	17.5	15.1	14.9
Barnes/Chambers	21	22.5	29.6	22.5	22.6
Hurink Edata	43	2.3	6.0	2.2	2.1
Hurink Rdata	43	1.3	4.4	1.2	1.2
Hurink Vdata	43	0.1	2.0	0.1	0.08

Limit on CPU time=15 sec.

HFS: Example

Processing times of a 3×2 Hybrid Flow Shop

Jobs	Stage 1		Stage 2	
1	O_{11}	8	O_{12}	7
2	O_{21}	7	O_{22}	8
3	O_{31}	8	O_{32}	8

$m_1 = 1;\ m_2 = 2$

EST-SPT (1st stage):
J_2 ; J_1 ; J_3