Faculté des Sciences de Tunis

INSA de Toulouse

DISCREPANCY AND BACKJUMPING HEURISTICS FOR FLEXIBLE JOB SHOP SCHEDULING

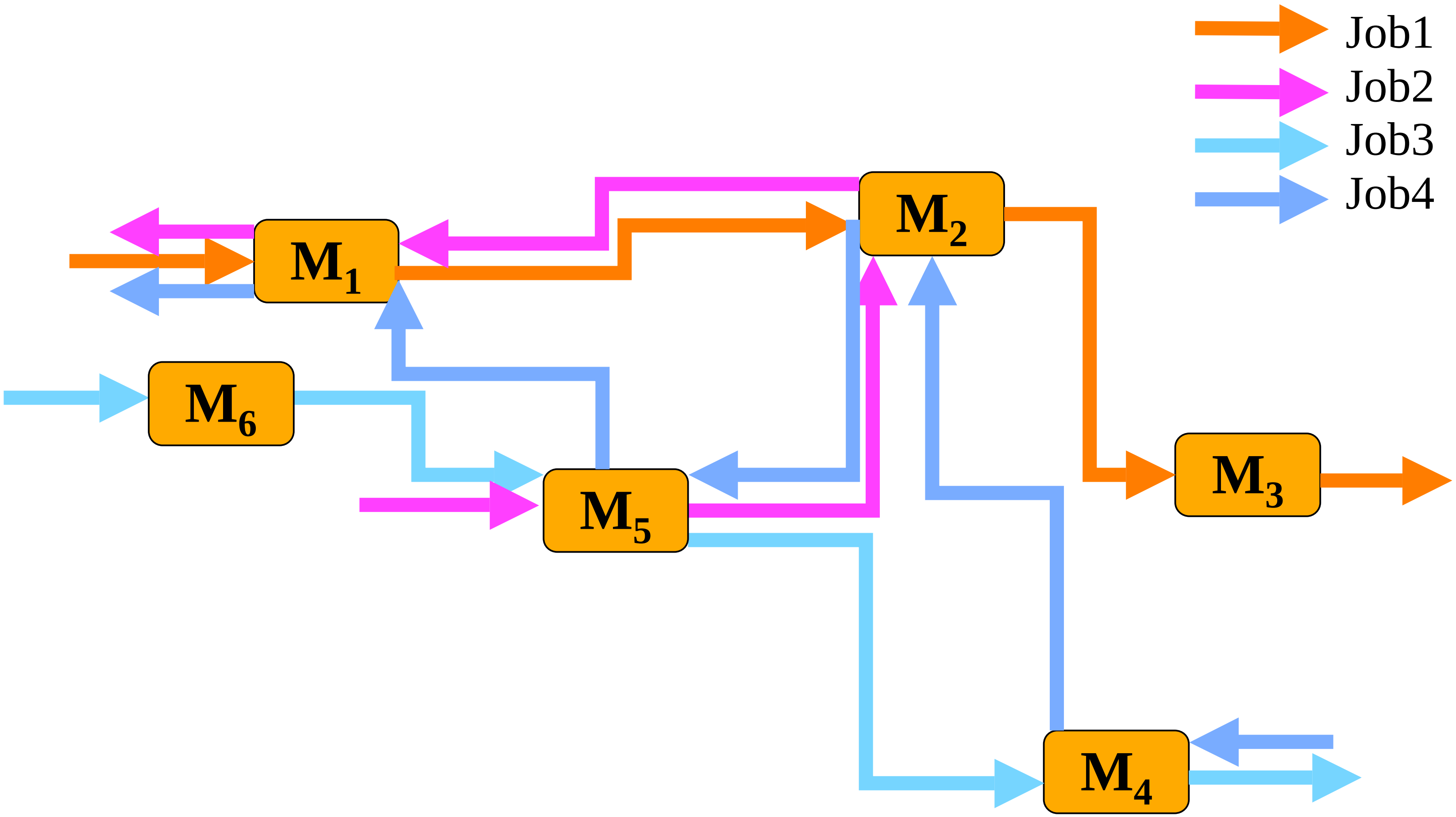
BEN HMIDA A., HAOUARI M., HUGUET M.J., LOPEZ P.

LAAS-CNRS, Toulouse, France
&
ROI, Ecole Polytechnique de Tunisie, Tunisia

OUTLINE

- Problem description
- Discrepancy-based methods
  - Limited Discrepancy Search (LDS)
  - Climbing Discrepancy Search (CDS)
- Proposed discrepancy-based method and its adaptation for the problem under consideration
- Numerical results
- Conclusion and further works

1. Problem Description (1/3)



1. Problem Description (2/3)

- $J = \{1, \dots, n\}$ jobs
- $M = \{1, \dots, m\}$ machines
- $J_i = \{O_{i,1}, O_{i,2}, \dots, O_{i,ni}\}$
- $M_{i,j} \subseteq M \ (\forall i, \bigcap M_{i,j} \neq \varnothing)$

- One operation at a time on a machine
- One machine at a time by operation
- No preemption

- Objective: Minimize the makespan $C_{max} = \max(ct_j)$
- Complexity: JMPM|| C_{max} is Strongly *NP-hard* [Brucker, 2004]

1. Problem Description (3/3)

Literature review

Exact algorithms

($JMPM|n=2|C_{max}$)

Brucker and Schlie (1990)

Jurisch (1992)

Heuristics

Mono-criteria

Mastrolilli and Gambardella (2000)

Ho, Tay and Lai (2007)

Pezella, Morganti and Ciaschetti (2007)

Multi-criteria

Zhang and Gen (2005)

Gao, Gen, Sun and Zhao (2007)

Vilcot (2007)

Gao, Sun and Gen (2008)

2. Discrepancy-Based Methods (1/3)

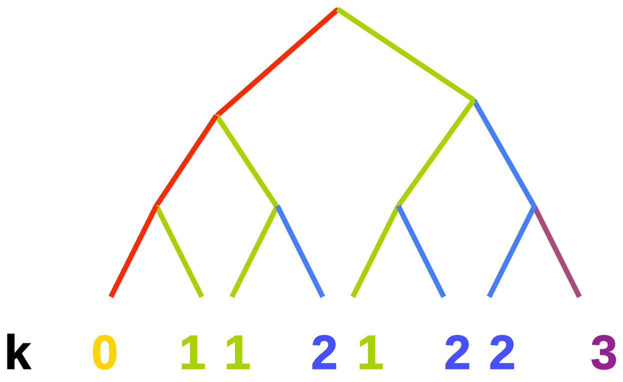
- Intuition: Heuristic usually finds good solution; when it doesn’t, it is probably because it made a *few* poor choices
- Discrepancy is when search makes choice other than heuristically top-ranked
- So... systematically introduce discrepancies as needed to find solution

2. Discrepancy-Based Methods (2/3)

- Limited Discrepancy Search (LDS) [Harvey & Ginsberg 1995]

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
```



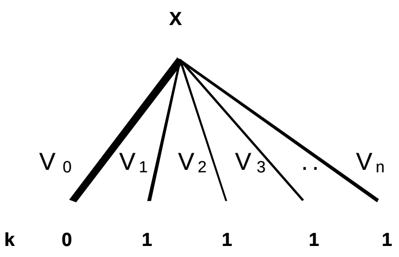
2. Discrepancy-Based Methods (3/3)

- LDS: searching for solution at *k* discrepancies, solutions with *0* to *k-1* discrepancies are revisited

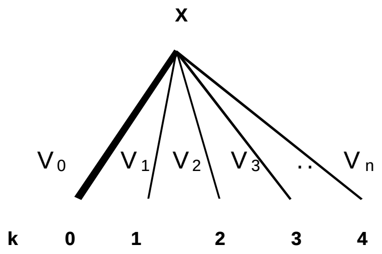
  ILDS: Improved Limited Discrepancy Search [Korf 1996]
- Depth-bounded Discrepancy Search (DDS): applying discrepancy first at the top of the tree to correct early mistakes in the instantiation heuristic [Walsh 1997]
- Discrepancy-Bounded Depth First Search (DBDFS): gives the priority to the depth [Beck and Perron, 2000]
- Climbing Discrepancy Search (CDS): reference solution is updated and the number of discrepancies is reset to zero (VND) [Milano and Roli, 2005]

3. Proposed Discrepancy-based Methods: Adaptation for the problem under consideration (1/3)

- Adaptation of discrepancy:



Binary counting



Non-Binary counting

- Strategy:
  - Select a job using *job selection heuristic: EST-LDJ*
  - Allocate a resource using a *heuristic for assignment of operations to machines: ECT*
  - Fix a start time
  - Applying the forward checking
    - Starting time of the following operation
    - Availability date of the chosen resource

3. Proposed Discrepancy-based Methods: Adaptation for the problem under consideration (2/3)

- Applying discrepancy on promising choice points chosen by using two types of *backjumping heuristics*:

  1. Permutation of two adjacent critical operations carried out by the same resource (discrepancy on selection variable). (van Laarhoven et al., 1992)

  2. Replacement of a critical operation on another resource (discrepancy on allocation variable but restricted to critical operations).

3. Proposed Discrepancy-based Methods: Adaptation for the problem under consideration (3/3)

- Applying discrepancy only at the top of the tree
- Limit the tree search expansion

Depth-bounded Discrepancy Search (DDS)

+

Climbing Discrepancy Search

=

Climbing Depth-bounded Discrepancy Search

(***CDDS*** = DDS + CDS)

4. Numerical results (1/3)

- ***Test beds***

  Brandimarte’s benchmarks

  -- 10 problems

  -- N=[10,20]; M=[4,15]; n_i=[5,15]

  Hurink’s benhmarks

  -- 129 problems (43 JSP)

  -- Edata, Rdata, Vdata: N=[6,30];M=[5,15]

  Mastrolilli and Gambardella’s results and lower bounds

- ***Results***

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

4. Numerical results (2/3)

Table 1. Comparison with the Tabu Search of Mastrolilli and Gambardella (M.G.) on 10 FJSP instances from Brandimarte

instances	*n*	*m*	*LB*	*M.G.*	*CDDS*	*%dev*	*CPU(M.G.)*	*CPU(CDDS)*
Mk01	10	6	36	**40**	**40**	0.0	0.01	0.1
Mk02	10	6	24	**26**	**26**	0.0	0.73	0.2
Mk03	15	8	204	**204***	**204***	0.0	0.01	0.2
Mk04	15	8	48	**60**	**60**	0.0	0.08	0.03
Mk05	15	4	168	173	182	5.2	0.96	0.2
Mk06	10	15	33	58	60	3.4	3.26	0.1
Mk07	20	5	133	144	**139**	-3.5	8.91	0.3
Mk08	20	10	523	**523***	**523***	0.0	0.02	0.8
Mk09	20	10	299	**307**	**307**	0.0	0.15	0.4
Mk10	20	15	165	198	212	7.1	7.69	0.3
Average						1.2	2.18	**0.26**

4. Numerical results (3/3)

Table 2. Deviation percentage over the best known lower bound

Data set	num	alt	CDDS (%)
Brandimarte	10	2.59	17.02
Hurink Edata	43	1.15	15.81
Hurink Rdata	43	2	9.85
Hurink Vdata	43	4.31	1.11

num: number of instances; alt:machine's number per job

5. Conclusions And Further Works (1/2)

- Climbing Depth-bounded Discrepancy Search (CDDS= DDS+CDS)
- Heuristics
  - Job selection heuristics (EST-LDJ)
  - Heuristic for assignment of operations to machines (ECT)
- Backjumping heuristics
  - Permutation of two adjacent critical operations carried out by the same resource
  - Replacement of a critical operation on another resource
- Constraint propagation Forward Checking
- The test problems are Brandimarte and Hurink's benchmarks

5. Conclusions And Further Works (2/2)

- Our results are compared with the best known TS procedure and LBs of Mastrolilli and Gambardella (2000)
- CDDS gives promising results
- Designing a diversification mechanism
- Other problems:
  - Flexible job shop problem with multi-criteria