Limited Discrepancy-Based Neighborhood Search for Parallel Machine Scheduling

B. Gacias, C. Artigues and P. Lopez

LAAS-CNRS, Université de Toulouse, France
emails: {bgacias,artigues,lopez}@laas.fr

SCRO/JOPT 2008

Outline

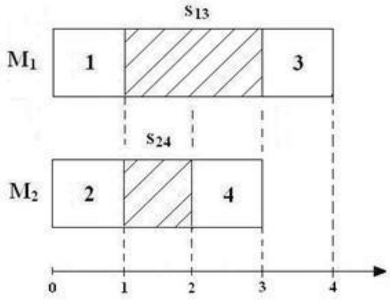
Outline

Introduction

- Parallel machine scheduling problem [Pearn et al., 2007]
- Solved by tree and local search methods [Néron et al., 2006]
- Problem with setup times and precedence constraints between jobs
- Difficulty for list scheduling algorithm [Hurink & Knust, 2001]

Example

- The problems can't be efficiently solved by a list scheduling algorithm



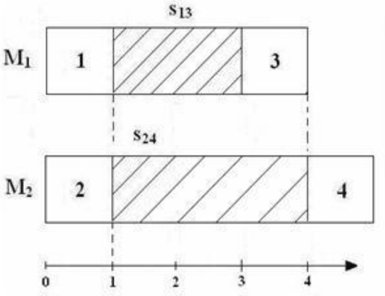
$$\begin{pmatrix} 0 & 10 & 2 & 10 \\ 10 & 0 & 0 & 1 \\ 10 & 10 & 0 & 10 \\ 10 & 10 & 10 & 0 \end{pmatrix}$$

$$\pi = (1,2,4,3)$$

$$\pi' = (2,1,4,3)$$

Example

- Precedence constraint $3 \prec 4$



$$\begin{pmatrix} 0 & 10 & 2 & 10 \\ 10 & 0 & 0 & 1 \\ 10 & 10 & 0 & 10 \\ 10 & 10 & 10 & 0 \end{pmatrix}$$

$$\pi = (1,2,4,3)$$

$$\pi' = (2,1,4,3)$$

- π and $\pi^{'}$ don't respect the precedence constraint
- Branching Structure: complete enumeration (best list of jobs and best resource allocation)

Outline

Problem statement

Problem data

Decision variables

Optimization criteria

Problem statement

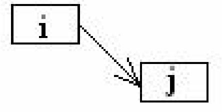
Problem data

- Jobs: $J = \{1, \ldots, n\}$
- Resources/Machines: $M = \{M_1, \ldots, M_m\}$
- For every job i: p_i, r_i, d_i
- Precedence constraints
- Setup times s_{ij}

Problem statement

Problem data

- Jobs: $J = \{1, \ldots, n\}$
- Resources/Machines: $M = \{M_1, \ldots, M_m\}$
- For every job i: p_i, r_i, d_i
- Precedence constraints

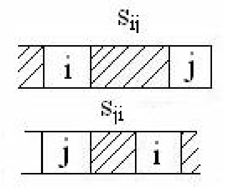


- Setup times s_{ij}

Problem statement

Problem data

- Jobs: $J = \{1, \ldots, n\}$
- Resources/Machines: $M = \{M_1, \ldots, M_m\}$
- For every job i: p_i, r_i, d_i
- Precedence constraints
- Setup times s_{ij}



Problem statement

Decision variables

- S_i $(C_i = S_i + p_i)$

Optimization criteria

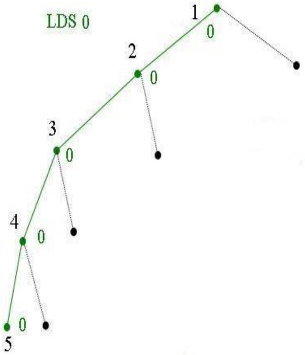
- $\min \sum C_i$ [Chu et al., 2005]
- $\min L_{\max}$ [Uzsoy & Velásquez, 2006]

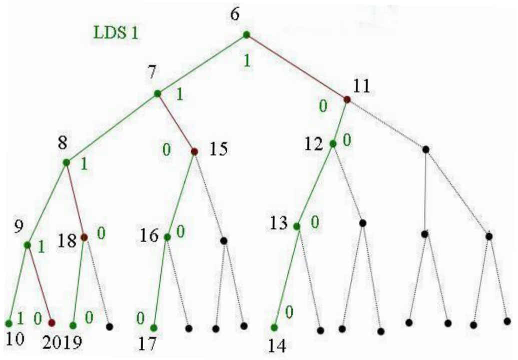
Outline

Discrepancy-based search methods

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

- Iterative tree search method
- Instantiation heuristic to guide the search (discrepancies)





Discrepancy-based search methods

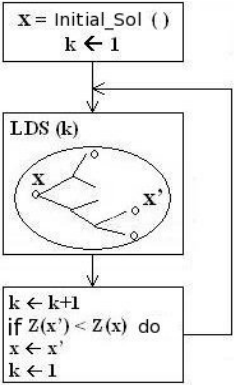
Methods based on LDS in order to increase efficiency:

- Improved LDS (ILDS) [Korf, 1996]
- Depth-bounded Discrepancy Search (DDS) [Walsh, 1997]
- Discrepancy-bounded Depth-First Search (DDFS) [Beck & Perron, 2000]
- Yet Improved LDS (YIELDS) [Karoui et al., 2007]

Large neighborhood local search based on LDS

Climbing Discrepancy Search (CDS) [Milano & Roli, 2002]

- Neighborhood exploration by LDS
- Discrepancies around the best current solution x

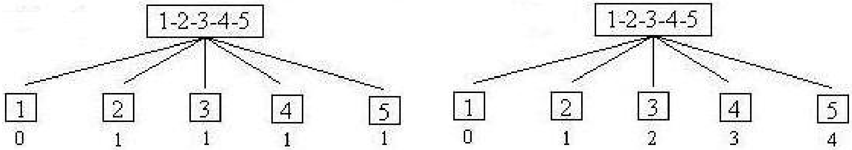


CDDS: the neighborhood is also limited by the depth in the tree [Ben Hmida et al., 2007]

Large neighborhood local search based on LDS

We propose two methods:

- Hybrid Discrepancy CDDS
  1. CDS search until k discrepancies
  2. $k \leftarrow k+1$
  3. CDDS search (not only on top of the tree), to restrict the neighborhood
- Mix Counting CDS
  - Different counting discrepancies ways$\longrightarrow$ Depth tree level



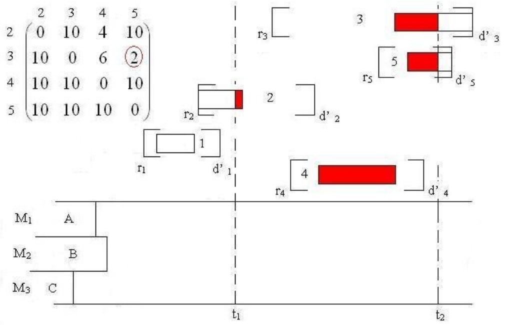
Outline

Branching strategy

- Initial heuristic
  - *Earliest Start Time (EST)* $\rightarrow$ minimize setup times
- Branching strategy
  - *DDFS (Discrepancy-bounded Depth First Search)*
  - *LDS-top*
  - *LDS-low*

Node evaluation

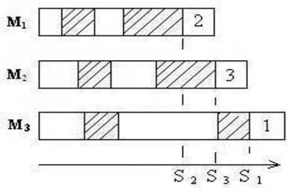
- min $\sum C_i$: Lower bound [Chu et al., 2005]
- min $L_{\max}$: Energetic reasoning [Lopez et al., 1992]
  - In a time interval $\Delta = [t_1, t_2] \rightarrow E_{consumed} + E_{setup} \leq E_{produced}$
  - $[r_i, d'_i]$, time windows for the set of not yet allocated jobs



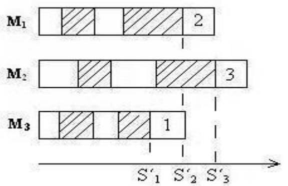
$E_{produced} = m \times (t_2 - t_1)$
$E_{consumed} = \sum_i E_{min}, \ i \in F$
$E_{setup} \rightarrow k - m$ shortest
$s_{ij}, \ i, j \in F$
where F is the set of jobs that $E_{consumed} > 0$

Node evaluation

Dominance rules for local search



Evaluated node



Partial dominant schedule

- Goal: To find a partial dominant schedule ($S'_i < S_i$ and $S'_j \leq S_j$ for the rest of the last jobs executed on every resource)
- It has to be included in the explored neighborhood (authorised discrepancies)

Outline

Computational experiments

Comparison on $P|r_i, q_i|C_{max}$ problems [Néron et al.]($n = 100$, $m = 10$, $p_i = [1 - 10]$)

$CPU_{limit} = 30\ s$	*Best Solution*	*Best Sol. Strict*	$CPU(s)$
$LDS^{TW}_{z=1}$	1	0	29.64
$LDS^{CHR}_{z=2}$	7	0	28.40
$BS^{TW}_{\omega=3}$	25	3	20.37
$BS^{CHR}_{\omega=4}$	22	0	28.40
CDS	35	6	30 (8.03)
HD-CDDS	38	9	30 (7.02)

Computational experiments

- *ECT* rule evaluation
  - We find the optimal solution for 85 % of instances
  - Consider the job order discrepancies separately of machine allocation discrepancies
- Comparison with local search methods ($n = 100$, $m = \{2, 4\}$, $s_{ij} = \{[1-10], [1-20]\}$, $p_i = \{[1-10], [1-5]\}$, $OS = \{0.75, 0.5, 0.25\}$)

$\sum C_i$	*Best Solution*	*Mean Dev.*
CDS	12 (20.0 %)	1.4 %
CDDS	2 (3.3 %)	4.8 %
HD-CDDS	27 (45.0 %)	0.9 %
MC-CDS	40 (66.7 %)	1.0 %

L_{max}	*Best Solution*	*Mean Dev.*
CDS	18 (30.0 %)	2.5 %
CDDS	0 (0.0 %)	11.6 %
HD-CDDS	36 (60.0 %)	1.8 %
MC-CDS	34 (56.7 %)	1.7 %

Conclusions and further work

- *LDS-top* branching strategy and *Binary counting* mode are the best strategies for *LDS* implementation
- Hybrid local-tree discrepancy based methods are efficient for parallel machine scheduling problems
- The computation of upper bounds improve the results (*EST*)
- We have efficiently introduced the setup times in the energetic reasoning and we have adapted the dominance rules for local search
- VRP problem: strategies and resolution methods