Unified Modeling Approach for Scheduling Time Representations

Sylvain Mouret, Ignacio E. Grossmann and Pierre Pestiaux

Enterprise-Wide Optimization, 29 September 2009

Motivation

Representation issues

- Most optimization problems are solved using mathematical or symbolic models
- The link between problem description and model is often unclear
- In scheduling problems, the time representation used is a key element in the performance of the algorithm
- 3 types of representation
  - Relaxation approximations: planning models
  - Constraining approximations: discrete-time formulations
  - General approximations: piecewise linearization of nonlinear constraints

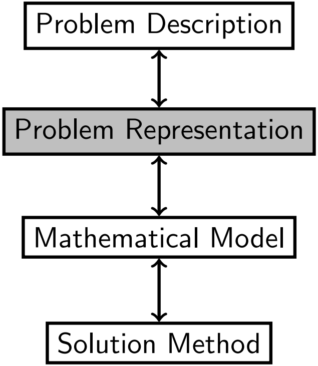


Figure: Four steps optimization method

Scheduling approach

Time representations

- Mathematical models for 4 different scheduling formulations
- Common set of variables
- Operation and priority-slot based representations

Scheduling approach

Time representations

- Mathematical models for 4 different scheduling formulations
- Common set of variables
- Operation and priority-slot based representations

Set of operations W

An operation is an action that can be executed one or several times.

Scheduling approach

Time representations

- Mathematical models for 4 different scheduling formulations
- Common set of variables
- Operation and priority-slot based representations

Set of operations W

An operation is an action that can be executed one or several times.

Set of priority-slots $T = \{1, \ldots, n\}$

Any operation assigned to priority-slot i is given scheduling priority i.

Scheduling approach

Time representations

- Mathematical models for 4 different scheduling formulations
- Common set of variables
- Operation and priority-slot based representations

Set of operations W

An operation is an action that can be executed one or several times.

Set of priority-slots $T = \{1, \ldots, n\}$

Any operation assigned to priority-slot i is given scheduling priority i.

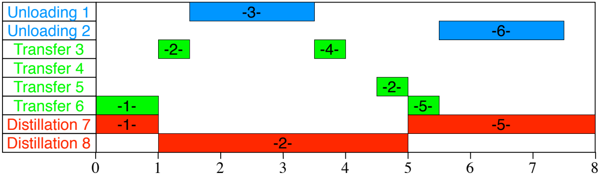
Type of scheduling problems

The proposed time representations can be used to model and solve scheduling problems in which operations can be sequenced as a whole.

- No sequencing of events such as start or end times
- Unsupported features: cumulative resource constraints, simultaneous inventory charging and discharging, ...

Multi-Operation Sequencing (MOS)

- 8 possible operations / 6 priority-slots
- Given 2 non-overlapping operations $v, w \in W$
  - v and w cannot be assigned to the same priority-slot
  - v and w are sequenced according to their scheduling priority
- Example: unloading operations 1 and 2 are assigned to slots 3 and 6

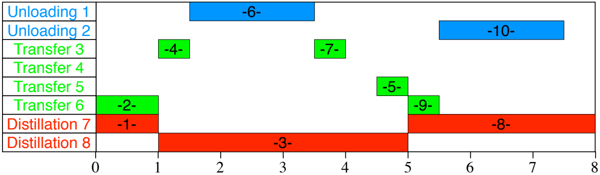


Priority-slot	-1-	-2-	-3-	-4-	-5-	-6-
Unloading 1			x			
Unloading 2						x
Transfer 3		x		x		
Transfer 4						
Transfer 5		x				
Transfer 6	x				x	
Distillation 7	x				x	
Distillation 8		x				

Figure: A solution schedule obtained using the MOS time representation

Single-Operation Sequencing (SOS)

- 8 possible operations / 10 priority-slots
- Same features as the MOS representation
- Specific feature:
  - At most one operation can be assigned to each priority-slot
  - The solution can be represented as a single sequence of operations

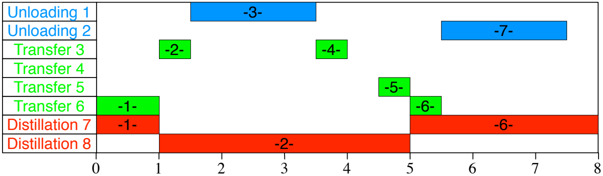


Priority-slot	-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-	-10-
Unloading 1						x				
Unloading 2										x
Transfer 3				x			x			
Transfer 4										
Transfer 5					x					
Transfer 6		x							x	
Distillation 7	x							x		
Distillation 8			x							

Figure: A solution schedule obtained using the SOS time representation

MOS with Synchronized Start Time (MOS-SST)

- 8 possible operations / 7 priority-slots
- Same features as the MOS representation
- Specific feature:
  - All operations assigned to priority-slot i must start at the same time
  - The scheduling horizon is divided into variable adjacent time intervals

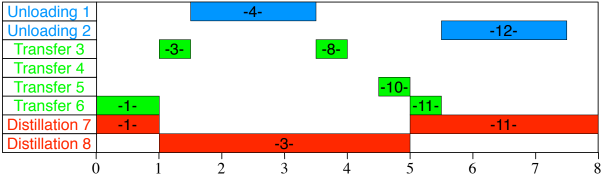


Priority-slot	-1-	-2-	-3-	-4-	-5-	-6-	-7-
Unloading 1			x				
Unloading 2							x
Transfer 3		x		x			
Transfer 4							
Transfer 5					x		
Transfer 6	x					x	
Distillation 7	x					x	
Distillation 8		x					

Figure: A solution schedule obtained using the MOS-SST time representation

MOS with Fixed Start Time (MOS-FST)

- 8 possible operations / 16 priority-slots
- Same features as the MOS-SST representation
- Specific feature:
  - All operations assigned to priority-slot *i* must start at fixed time point
  - The scheduling horizon is divided into fixed adjacent time intervals



Priority-slot	-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-	-10-	-11-	-12-	-13-	-14-	-15-	-16-
Unloading 1				x												
Unloading 2												x				
Transfer 3			x					x								
Transfer 4																
Transfer 5										x						
Transfer 6	x										x					
Distillation 7	x										x					
Distillation 8			x													

Figure: A solution schedule obtained using the MOS-FST time representation

Time Representation Summary

Required number of priority-slots

For a given solution s of the scheduling problem

- MOS finds s with fewer priority-slots than
  - SOS
  - MOS-SST
  - MOS-FST
- MOS-SST finds s with fewer priority-slots than MOS-FST

Time Representation Summary

Required number of priority-slots

For a given solution *s* of the scheduling problem

- MOS finds *s* with fewer priority-slots than
  | SOS
  | MOS-SST
  | MOS-FST
- MOS-SST finds *s* with fewer priority-slots than MOS-FST

Integer feasible space

For a given number of priority-slots *n*

- The integer feasible space of MOS includes the integer feasible space of
  | SOS
  | MOS-SST
  | MOS-FST
- The integer feasible space of MOS-SST includes the integer feasible space of MOS-FST

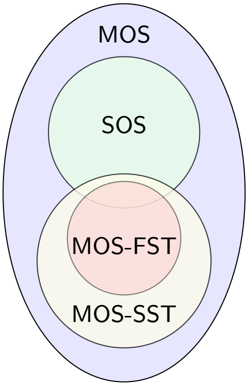


Figure: Integer feasible space of the 4 time representations (fixed number of priority-slots *n*)

Mathematical Model: MOS

- Assignment variables

$$Z_{iv} \in \{0,1\} \qquad i \in T, v \in W$$

- Start, duration, end variables

$$S_{iv}, D_{iv}, E_{iv} \geq 0 \qquad i \in T, v \in W$$

- Variable definition constraints

$$E_{iv} = S_{iv} + D_{iv} \quad i \in T, v \in W$$

$$\underline{S_v} \cdot Z_{iv} \leq S_{iv} \leq \overline{S_v} \cdot Z_{iv} \quad i \in T, v \in W$$

$$\underline{D_v} \cdot Z_{iv} \leq D_{iv} \leq \overline{D_v} \cdot Z_{iv} \quad i \in T, v \in W$$

$$\underline{E_v} \cdot Z_{iv} \leq E_{iv} \leq \overline{E_v} \cdot Z_{iv} \quad i \in T, v \in W$$

- Assignment constraint

$$Z_{iv_1} + Z_{iv_2} \leq 1 \qquad i \in T, v_1, v_2 \in W, NO_{v_1 v_2} = 1$$

- Cardinality constraint

$$\underline{N_{W'}} \leq \sum_{v \in W'} Z_{iv} \leq \overline{N_{W'}} \quad i \in T, W' \subset W$$

- Non-overlapping constraint

$$E_{i_1 v_1} \leq S_{i_2 v_2} + \overline{E_{v_1}} \cdot (1 - Z_{i_2 v_2}) \qquad i_1, i_2 \in T, i_1 \leq i_2, v_1, v_2 \in W, NO_{v_1 v_2} = 1$$

Mathematical Model: MOS-FST

- Assignment variables

$$Z_{iv} \in \{0,1\} \qquad i \in T, v \in W$$

- Start, duration, end variables

$$S_{iv}, D_{iv}, E_{iv} \geq 0 \qquad i \in T, v \in W$$

- Time point parameter $t_i = \frac{i-1}{n} \cdot H$
- Variable definition constraints

$$E_{iv} = S_{iv} + D_{iv} \quad i \in T, v \in W$$

$$\underline{S_v} \cdot Z_{iv} \leq S_{iv} \leq \overline{S_v} \cdot Z_{iv} \quad i \in T, v \in W$$

$$\underline{D_v} \cdot Z_{iv} \leq D_{iv} \leq \overline{D_v} \cdot Z_{iv} \quad i \in T, v \in W$$

$$\underline{E_v} \cdot Z_{iv} \leq E_{iv} \leq \overline{E_v} \cdot Z_{iv} \quad i \in T, v \in W$$

- Assignment constraint

$$Z_{iv_1} + Z_{iv_2} \leq 1 \qquad i \in T, v_1, v_2 \in W, NO_{v_1 v_2} = 1$$

- Cardinality constraint

$$\underline{N_{W'}} \leq \sum_{v \in W'} Z_{iv} \leq \overline{N_{W'}} \quad i \in T, W' \subset W$$

- Non-overlapping constraint

$$E_{i_1 v_1} \leq S_{i_2 v_2} + \overline{E_{v_1}} \cdot (1 - Z_{i_2 v_2}) \qquad i_1, i_2 \in T, i_1 \leq i_2, v_1, v_2 \in W, NO_{v_1 v_2} = 1$$

- Synchronization constraint

$$S_{iv} = t_i \cdot Z_{iv} \qquad i \in T, v \in W$$

Single stage batch scheduling problem

- Up to 29 batch orders $O = \{o_1, \ldots, o_{29}\}$
- 4 parallel units $U = \{u_1, \ldots, u_4\}$ with unit-dependent transition times
- The set of operations is defined by

$$W = \{v = (o, u), o \in O, u \in U_o\}$$

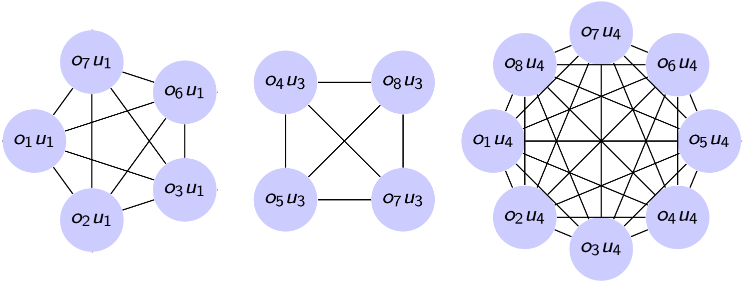
- Non-overlapping graph $G_{NO} = (V, E)$ where
  - $V = W$
  - $E = \{\{v_1, v_2\}, v_1, v_2 \in W, NO_{v_1 v_2} = 1\}$

Single stage batch scheduling problem

- Up to 29 batch orders $O = \{o_1, \dots, o_{29}\}$
- 4 parallel units $U = \{u_1, \dots, u_4\}$ with unit-dependent transition times
- The set of operations is defined by

$$W = \{v = (o, u), o \in O, u \in U_o\}$$

- Non-overlapping graph $G_{NO} = (V, E)$ where
  - $V = W$
  - $E = \{\{v_1, v_2\}, v_1, v_2 \in W, NO_{v_1 v_2} = 1\}$
- For the 8 orders problem, it contains 3 isolated cliques



Strengthened reformulations: MOS

Strengthened constraints

- Assignment constraint using cliques

$$\sum_{v \in W_u} Z_{iv} \leq 1 \qquad i \in T, u \in U, W_u \in clique(W)$$

- Non-overlapping constraint using cliques

$$\sum_{v \in W_u} E_{i_1 v} \leq \sum_{v \in W_u} S_{i_2 v} + \left(\max_{v \in W_u} \overline{E_v}\right) \cdot \left[1 - \sum_{v \in W_u} Z_{i_2 v}\right] \qquad i_1, i_2 \in T, i_1 \leq i_2, u \in U, W_u \in clique(W)$$

Strengthened reformulations: MOS

Strengthened constraints

- Assignment constraint using cliques

$$\sum_{v \in W_u} Z_{iv} \leq 1 \qquad i \in T, u \in U, W_u \in clique(W)$$

- Non-overlapping constraint using cliques

$$\sum_{v \in W_u} E_{i_1 v} \leq \sum_{v \in W_u} S_{i_2 v} + \left(\max_{v \in W_u} \overline{E_v} \right) \cdot \left[1 - \sum_{v \in W_u} Z_{i_2 v} \right] \qquad i_1, i_2 \in T, i_1 \leq i_2, u \in U, W_u \in clique(W)$$

Symmetry-breaking constraints using isolated cliques

$$\sum_{v \in W_u} Z_{iv} \leq \sum_{v \in W_u} Z_{(i-1)v} \qquad i \in T, i \neq 1, u \in U$$

Strengthened reformulations: MOS

Strengthened constraints

- Assignment constraint using cliques

$$\sum_{v \in W_u} Z_{iv} \leq 1 \qquad i \in T, u \in U, W_u \in clique(W)$$

- Non-overlapping constraint using cliques

$$\sum_{v \in W_u} E_{i_1 v} \leq \sum_{v \in W_u} S_{i_2 v} + \left(\max_{v \in W_u} \overline{E_v} \right) \cdot \left[1 - \sum_{v \in W_u} Z_{i_2 v} \right] \qquad i_1, i_2 \in T, i_1 \leq i_2, u \in U, W_u \in clique(W)$$

Symmetry-breaking constraints using isolated cliques

$$\sum_{v \in W_u} Z_{iv} \leq \sum_{v \in W_u} Z_{(i-1)v} \qquad i \in T, i \neq 1, u \in U$$

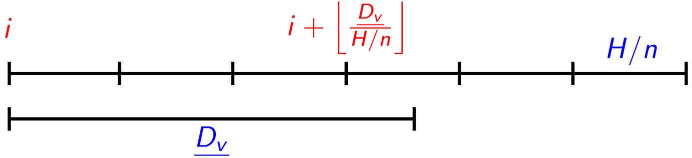
Minimum priority-slot usage constraint

$$\sum_{v \in W} Z_{iv} \geq 1 \qquad i \in T$$

Strengthened reformulations: MOS-FST

Assignment constraint using time points

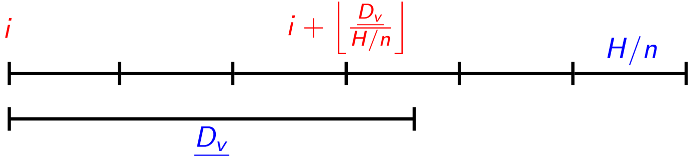
$$\sum_{\substack{j \in T \\ i \le j \le i + \left\lfloor \frac{D_v}{H/n} \right\rfloor}} Z_{jv} \le 1 \qquad i \in T, v \in W$$



Strengthened reformulations: MOS-FST

Assignment constraint using time points

$$\sum_{\substack{j \in T \\ i \leq j \leq i + \left\lfloor \frac{D_v}{H/n} \right\rfloor}} Z_{jv} \leq 1 \qquad i \in T, v \in W$$



Assignment constraint using cliques

$$\sum_{\substack{j \in T \\ i \leq j \leq i + \left\lfloor \frac{D_{W_u}}{H/n} \right\rfloor}} \sum_{v \in W_u} Z_{jv} \leq 1 \quad i \in T, u \in U, W_u \in clique(W), \underline{D_{W_u}} = \min_{v \in W_u} \underline{D_v}$$

Single stage batch scheduling MOS results

- The scheduling objective is to minimize earliness by maximizing completion times: $\max \sum_{i \in T} \sum_{v \in W} E_{iv}$
- MILP solver: Ilog Cplex 11.1.1
- For 29 orders and 10 priority-slots, the MILP model has 529 binary variables, 1,752 continuous variables, 2,007 constraints and 26,790 nonzeroes.
- The global optimum is obtained using the following algorithm:

`For` $n = 1$ `to` $|O|$, `solve` $MOS(n)$ `using cutoff=`$\max_{m<n} MOS^*(m)$

Orders	n^*	LP	MILP	Nodes	CPU	Total CPU
8	3	189.000	189.000	0	0s	0s
12	4	299.000	297.974	544	1s	2s
18	7	467.350	451.504	1557	5s	29s
25	9	607.849	579.570	51835	157s	1040s
29	10	662.728	635.104	37042	116s	639s

Table: Computational results obtained during iterations leading to global optimal solutions for each problem with total CPU time

Single stage batch scheduling MOS results

- 3 types of MOS models have been compared for the 18 orders problem with varying number of slots
  - MOS-sr-sb-mu is the MOS model with **s**trengthened **r**eformulation, **s**ymmetry-**b**reaking constraints, and **m**inimum priority-slot **u**sage
  - MOS-sr-sb
  - MOS-sr-mu

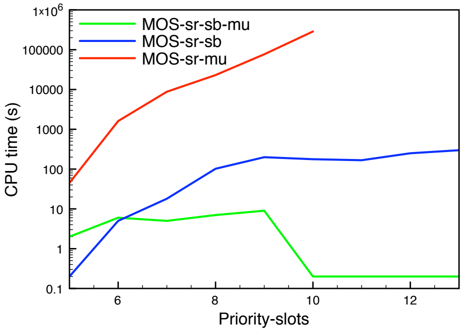
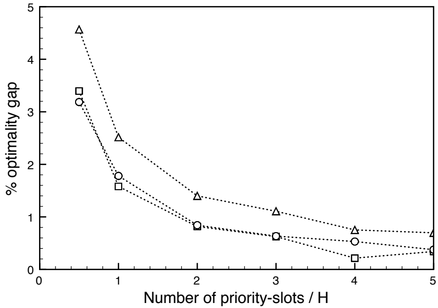
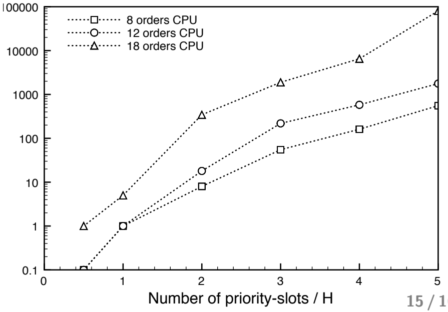


Figure: CPU times for different MOS models

Single stage batch scheduling MOS-FST results

- The performance of the MOS-FST model has been tested for the 8, 12, and 18 orders problems with varying number of slots ($\frac{1}{2}H$, H, $2H$, $3H$, $4H$, $5H$ slots)
- Model size for 8 orders problems with 60 slots: 1,020 binary variables, 3,061 continuous variables, 9,415 constraints and 1,153,151 nonzeroes.
- Near-optimal solutions are obtained quickly although global optimal solutions are difficult to obtain
- No proof of global optimality





Conclusions

- Four different priority-slot based time representations can be designed using identical concepts

Conclusions

- Four different priority-slot based time representations can be designed using identical concepts
- Corresponding MILP scheduling models can be efficiently solved using
  - Strengthened reformulations
  - Symmetry-breaking constraints
  - Minimum priority-slot usage constraint
  - Iterative algorithm using MILP cutoffs

Conclusions

- Four different priority-slot based time representations can be designed using identical concepts
- Corresponding MILP scheduling models can be efficiently solved using
  - Strengthened reformulations
  - Symmetry-breaking constraints
  - Minimum priority-slot usage constraint
  - Iterative algorithm using MILP cutoffs
- MOS representations can be solved faster and to global optimality as opposed to MOS-FST representations

Conclusions

- Four different priority-slot based time representations can be designed using identical concepts
- Corresponding MILP scheduling models can be efficiently solved using
  - Strengthened reformulations
  - Symmetry-breaking constraints
  - Minimum priority-slot usage constraint
  - Iterative algorithm using MILP cutoffs
- MOS representations can be solved faster and to global optimality as opposed to MOS-FST representations
- MOS models require stronger modeling skills than MOS-FST

Conclusions

- Four different priority-slot based time representations can be designed using identical concepts
- Corresponding MILP scheduling models can be efficiently solved using
  - Strengthened reformulations
  - Symmetry-breaking constraints
  - Minimum priority-slot usage constraint
  - Iterative algorithm using MILP cutoffs
- MOS representations can be solved faster and to global optimality as opposed to MOS-FST representations
- MOS models require stronger modeling skills than MOS-FST
- SOS and MOS-SST will be considered later, as well as other scheduling applications: multi stage batch scheduling and crude-oil operations scheduling