A CP Based Branching Tool for Breaking Symmetries in Crude-Oil Operations Scheduling

Sylvain Mouret, Ignacio E. Grossmann and Pierre Pestiaux

Entreprise-Wide Optimization, 26 September 2008

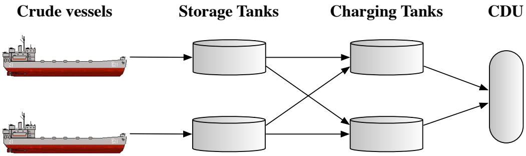
Introduction

Goals

- Optimize the schedule of operations for the crude-oil unloading and blending problem using a continuous-time mathematical formulation
- Reduce the computation expense by breaking the symmetries in the MINLP model
- Use CP inference techniques to improve performance

Crude-oil operations scheduling problem

- Scheduling horizon $[0, H]$
- 4 types of resources:
  - Crude-oil marine vessels
  - Storage tanks
  - Charging tanks
  - Crude Distillation Units (CDUs)
- 3 types of operations:
  - Unloading: Vessel unloading to storage tanks
  - Transfer: Transfer from storage tanks to charging tanks
  - Distillation: Distillation of charging tanks

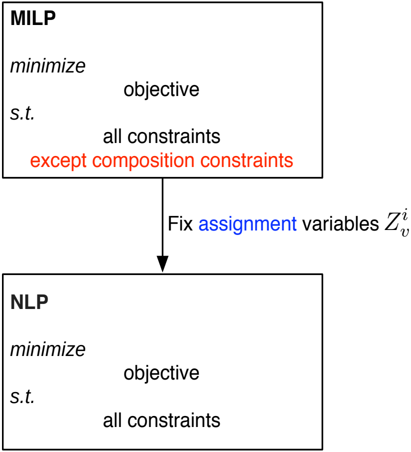


Solution method

Decomposition steps

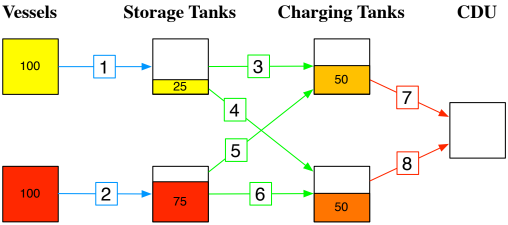
The MINLP model is solved using the following MILP - NLP decomposition procedure :

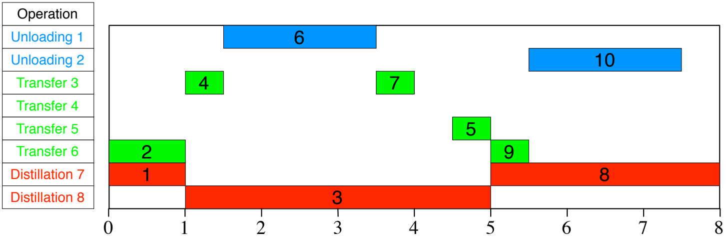
1. **Solve** the MILP relaxation
2. **Check** feasibility: the solution may not satisfy the nonlinear composition constraints
3. If infeasible, **Fix** assignment variables
4. **Solve** the resulting NLP (with nonlinear composition constraints)



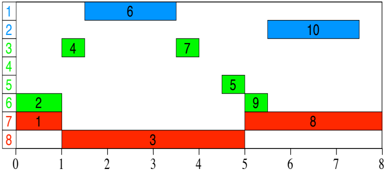
MINLP model: Basic Idea

- A schedule is represented as a sequence of operations
- The model enforces non-overlapping conditions between operations

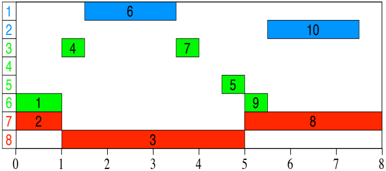




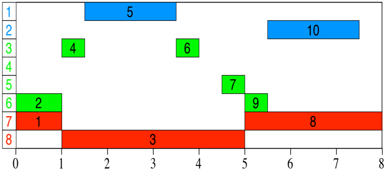
Symmetric sequences of operations



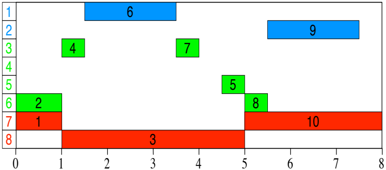
76 - 83513 - 762



67 - 83513 - 762



76 - **83135** - 762



76 - 83513 - **627**

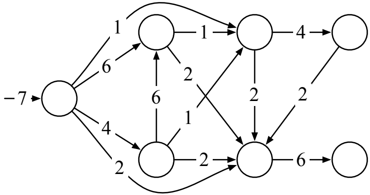
Breaking symmetries

Main Idea

- Define a **sequencing rule** to restrict the number of discrete solutions for the same schedule
- A regular language and its corresponing deterministic finite automaton (DFA) are used to express the rule
- The regular language membership constraint on the sequence of operations is derived from the DFA as a system of linear equations (Côté et al., 2007) which can be added to the original model

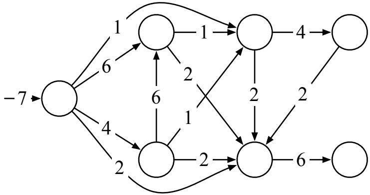
Rule derivation

- Regular language $L_7 = 7(\varepsilon + 4)(\varepsilon + 6)(\varepsilon + 1 + 14)(\varepsilon + 2 + 26)$
- DFA recognizing the regular language L_7

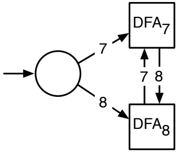


Rule derivation

- Regular language $L_7 = 7(\varepsilon + 4)(\varepsilon + 6)(\varepsilon + 1 + 14)(\varepsilon + 2 + 26)$
- DFA recognizing the regular language L_7



- Regular language $L = (\varepsilon + L_7)(L_8 L_7)^*(\varepsilon + L_8)$
- DFA recognizing the regular language L



Regular language membership constraint

The regular language membership constraint, noted `regular`, initially developped by Pesant (2003) for CP, has been adapted by Côté et al. (2007) for MILP. It is formulated as a network flow problem.

$$Z_{iv} = \sum_{q} S_{ivq} \qquad \forall i, v$$

$$\sum_{v} S_{1vq_0} = 1$$

$$\sum_{v,q',q=\delta(q',v)} S_{(i-1)vq'} = \sum_{v} S_{ivq} \qquad \forall i, q$$

$$\sum_{v,q,\delta(q,v)\in F} S_{nvq} = 1$$

Computational Results: Motivation for using CP

- Approach tested on the 4 problems from Lee et al. (1996) using Xpress (MILP) and CONOPT (NLP)

Pb	Operations	Slots	MILP	NLP	Gap	Nodes	CPU
1	8	13	79.75	79.75	0%	17	2s
2	14	21	101.17	101.17	0%	23	16s
3	14	21	87.4	84.5	3.3%	45	47s
4	19	26	132.5	132.55	0%	21	82s

- Basic MILP model without symmetry-breaking constraints -12 slots
  - Size: 1,189 variables (96 binary), 2,383 constraints
  - CPU time: +3,600s
  - Number of nodes: +1,990,700
- Extended MILP model with symmetry-breaking constraints -12 slots
  - Size: 2,629 variables (96 binary), 2,646 constraints
  - CPU time: 2s
  - Number of nodes: 63

Branch & Bound search

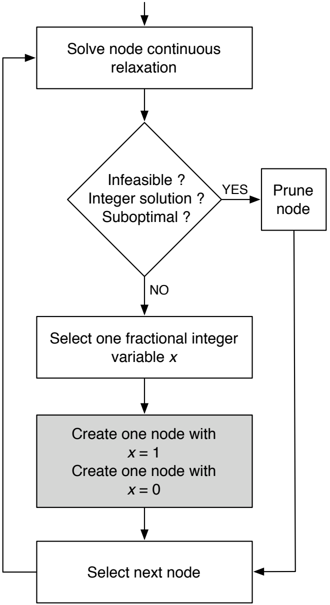
Branch & Bound search

The B&B algorithm processes a node queue by successively

- solving continuous relaxations to potentially prune nodes
- branching on fractional integer variables to create new nodes

CP branching

CP inference techniques are used in order to extend the branching decisions (grey step)



CP based branching

CP model

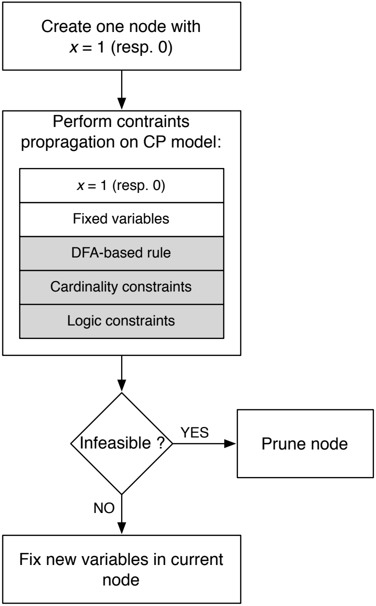
The CP model includes only integer variables and is composed of two main submodels:

- Variable submodel with fixed variables and branching decision
- Fixed submodel with logic constraints (in grey)

CP branching

Constraint propagation is performed on CP model leading to

- infeasibility due to logic constraints on integer variables,
- or extension of the branching decision (new fixed variable)



CP based branching features

Global performance

+ Constraint propagation is not expensive
+ Some constraints can be removed from the MILP
– Can lead to less tight MILP formulation

CP based branching features

Global performance

- \+ Constraint propagation is not expensive
- \+ Some constraints can be removed from the MILP
- – Can lead to less tight MILP formulation

Implementation

- \+ Can be implemented within commercial MILP solver using callbacks
- – Need to write two models
- – Communication between MILP and CP solvers (Ilog Concert used to communicate between Ilog Cplex and Ilog CP)
- ▶ Implemented in C++ but could be implemented in Ilog OPL

Computational results

- Approach tested on the MILP relaxation of 4 problems from Lee et al. (1996) using Ilog Cplex (MILP) and Ilog Solver (CP)
- The hybrid MILP-CP strategy shows a 55% average decrease in CPU time, in spite of larger number of nodes explored

			Pure MILP		MILP-CP	
Pb	Operations	Slots	Nodes	CPU	Nodes	CPU
1	8	13	18	5s	21	2s
2	14	21	27	124s	71	58s
3	14	21	54	179s	189	144s
4	19	26	27	451s	91	146s

Conclusion and future work

Conclusion

- New MINLP formulation for the crude-oil operations problem
- Handles logistics constraints and maximization of crude-oil gross margins
- Includes DFA-based symmetry-breaking constraints relying on the structure of the scheduling problem
- CP based branching algorithm to improve performances

Conclusion and future work

Conclusion

- New MINLP formulation for the crude-oil operations problem
- Handles logistics constraints and maximization of crude-oil gross margins
- Includes DFA-based symmetry-breaking constraints relying on the structure of the scheduling problem
- CP based branching algorithm to improve performances

Future work

- Contraint Programming as a continuous variable bounding tool
- Enhance the MILP-NLP decomposition (LP/NLP based branch and bound, Quesada and Grosmann, 1992)
- Take into account stochastic parameters (vessels arrival time)
- Integration with refinery planning