Recursive column generation for the Tactical Berth Allocation Problem

Ilaria Vacca[1] Matteo Salani[2] Michel Bierlaire[1]

[1]Transport and Mobility Laboratory, EPFL, Lausanne, Switzerland

[2]IDSIA, Lugano, Switzerland

TRISTAN VII - Seventh Triennal Symposium on Transportation Analysis

June 24, 2010

Outline

- Container terminals
- Berth Allocation & Quay Crane Assignment
- Tactical Berth Allocation Problem (TBAP)
- Dantzig-Wolfe decomposition
- Recursive column generation for TBAP
- Conclusions

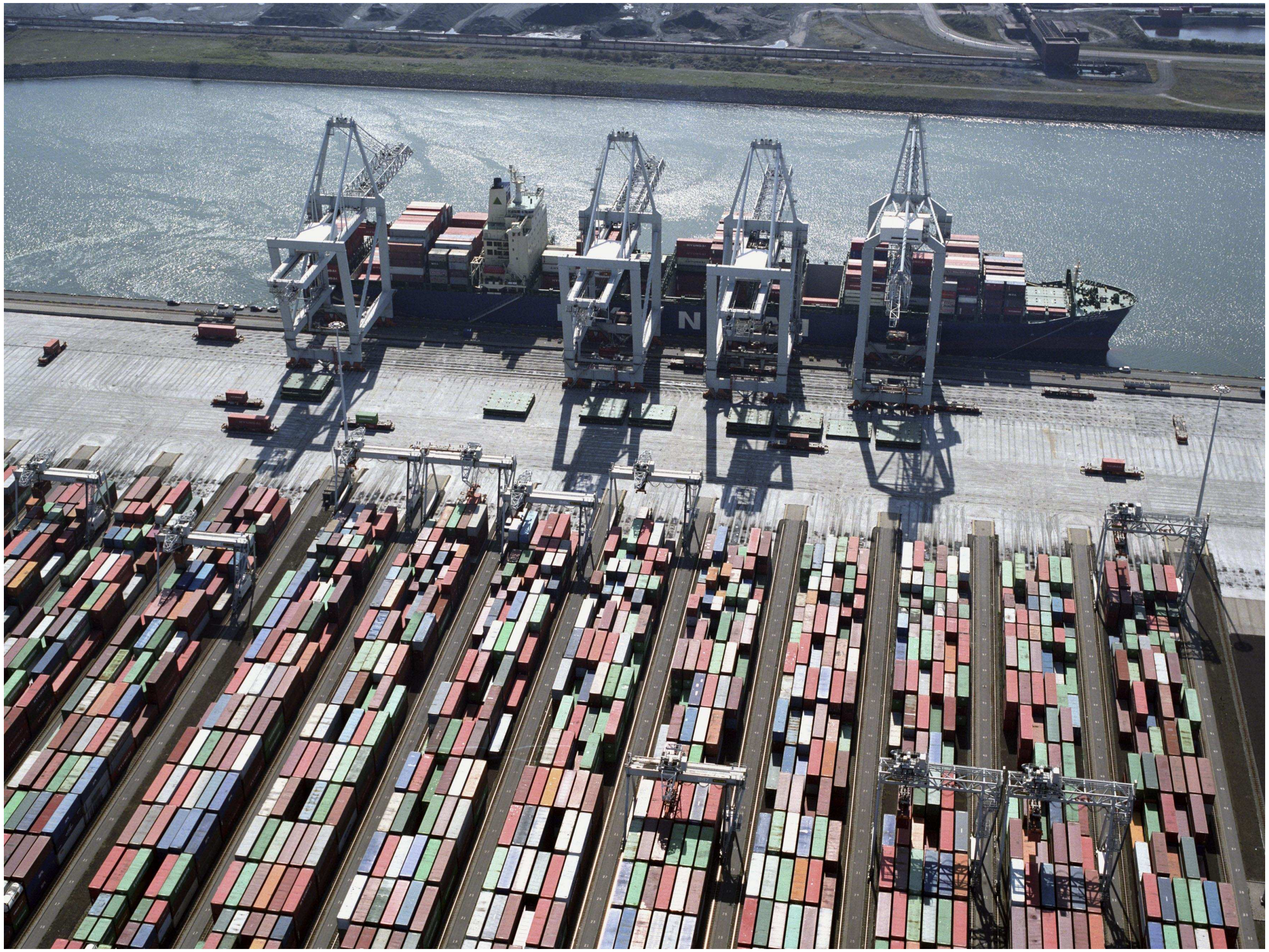
Motivation

Worldwide		2004	2006		2008	
1	Singapore	21’329	24’792	(+16%)	29’918	(+21%)
2	Shanghai	14’557	21’710	(+49%)	27’980	(+29%)
3	Hong Kong	21’984	23’539	(+07%)	24’248	(+03%)

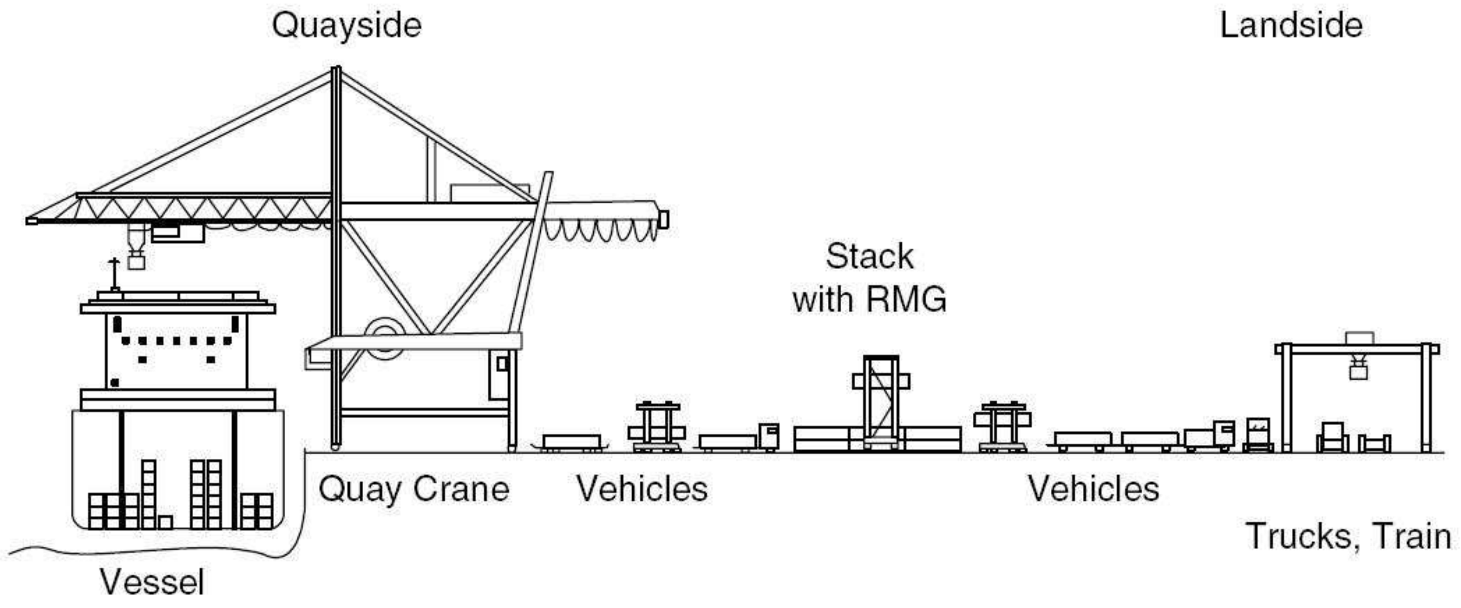
Europe		2004	2006		2008	
1	Rotterdam	8’291	9’655	(+17%)	10’784	(+12%)
2	Hamburg	7’003	8’862	(+27%)	9’737	(+10%)
3	Antwerp	6’064	7’019	(+16%)	8’663	(+23%)

Table 1: Container traffic (in thousands TEUs).

Container terminals



Container terminals



Scheme of a container terminal system (Steenken et al., 2004).

Berth Allocation & Quay Crane Assignment

Berth Allocation Problem (BAP)

to assign and to schedule ships to berths over a time horizon, according to an *expected handling time*, time windows on the arrival time of ships and availability of berths.

Quay-Crane Assignment Problem (QCAP)

to assign quay cranes (QC) to ships scheduled by the given berth allocation plan, over a time horizon, taking into account the *QC capacity constraint* in terms of available quay cranes at the terminal.

Tactical Berth Allocation Problem (TBAP)

Integration of BAP and QCAP

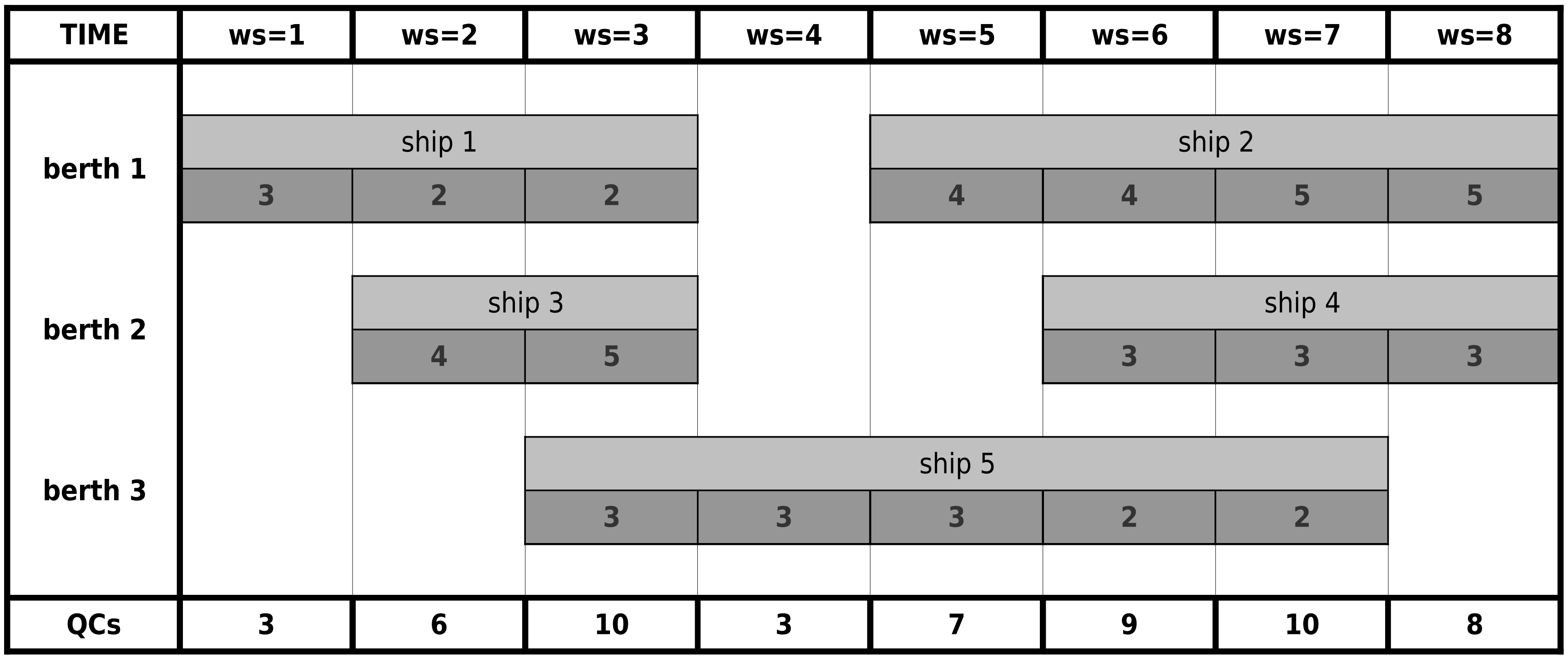
- *tactical decision level*: we analyze the problem from the terminal point of view, in order to provide decision support in the context of the negotiation between the terminal and shipping lines.
- *quay-crane profiles and handling time*: the handling time becomes a decision variable, dependent on the assigned quay crane profile (i.e. number of cranes per shift, ex. 332). Feasible profiles can vary in length (number of shifts dedicated to the ship) and in size (number of QCs dedicated to the ship in each active shift).

Housekeeping Yard Costs

in the context of a *transshipment container terminal*, we take into account the cost generated by the exchange of containers between ships in terms of traveled distance quay-yard-quay.

More details in (Giallombardo et al., 2010).

The concept of QC assignment profile



TIME	ws=1	ws=2	ws=3	ws=4	ws=5	ws=6	ws=7	ws=8
berth 1	ship 1: 3	ship 1: 2	ship 1: 2		ship 2: 4	ship 2: 4	ship 2: 5	ship 2: 5
berth 2		ship 3: 4	ship 3: 5			ship 4: 3	ship 4: 3	ship 4: 3
berth 3			ship 5: 3	ship 5: 3	ship 5: 3	ship 5: 2	ship 5: 2	
QCs	3	6	10	3	7	9	10	8

Problem definition

Find

- a berth allocation;
- a schedule;
- a quay crane assignment;

Given

- time windows on availability of berths;
- time windows on arrival of ships;
- *handling times dependent on QC profiles*;
- values of QC profiles;

Objective

- maximize total value of QC assignment;
- minimize housekeeping costs of transshipment flows between ships.

Notation & data

N	set of **vessels**;
M	set of **berths**;
H	set of **time steps**;
P_i	set of **quay crane profiles** for the vessel $i \in N$;
t_i^p	**handling time** of ship $i \in N$ using QC profile $p \in P_i$;
v_i^p	monetary **value** associated to qc profile $p \in P_i, \ i \in N$;
q_i^{pu}	number of **quay cranes** used by profile $p \in P_i, \ i \in N$ at time position u;
Q^h	maximum number of quay cranes available at the time step $h \in H$;
f_{ij}	**flow of containers** exchanged between vessels $i, j \in N$;
g_{ij}	binary parameter equal to 1 if $f_{ij} > 0$ and 0 otherwise;
d_{kw}	**unit housekeeping cost** between yard slots corresponding to berths $k, w \in M$.

Column generation for TBAP

- We propose a Dantzig-Wolfe (DW) reformulation of the MILP by Giallombardo et al. (2010) and we solve it using column generation.
- A *column* represents the sequence of ships calling at a given berth.
- A quay crane profile is assigned to every ship in the sequence.
- The *master problem* selects sequences in order to provide a min-cost solution.
- Profitable columns are generated by the *pricing subproblem*.

Master problem

Additional notation

Ω^k	set of all feasible sequences for berth $k \in M$;
α_r^i	coefficient equal to 1 if ship i is operated in sequence r, 0 otherwise;
β_r^{ip}	coefficient equal to 1 if ship i is operated in sequence r with profile p, 0 otherwise;
q_r^h	number of quay cranes used by sequence r at time step h;
v_r	total value of sequence $r \in \Omega^k$ defined as $v_r = \sum_{i \in N} \sum_{p \in P_i} \beta_r^{ip} v_i^p$.

Decision variables

s_r	equal to 1 if sequence $r \in \Omega^k$ is chosen, 0 otherwise;
z_{ij}^{kw}	equal to 1 if ship $i \in N$ is assigned to berth $k \in M$ and ship $j \in N$ to berth $w \in M$, 0 otherwise.

Master problem

Objective function

$$\min \sum_{i\in N}\sum_{j\in N}\sum_{k\in M}\sum_{w\in M} f_{ij} d_{kw} z_{ij}^{kw} - \sum_{k\in M}\sum_{r\in\Omega^k} v_r s_r \tag{1}$$

Ship covering & berth assignment

$$\sum_{k\in M}\sum_{r\in\Omega^k} \alpha_r^i s_r = 1 \quad \forall i \in N, \tag{2}$$

$$\sum_{r\in\Omega^k} s_r \leq 1 \quad \forall k \in M, \tag{3}$$

Quay-cranes capacity

$$\sum_{k\in M}\sum_{r\in\Omega^k} q_r^h s_r \leq Q^h \quad \forall h \in H, \tag{4}$$

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Master problem

Linearization constraints

$$\begin{aligned}
\sum_{k\in M}\sum_{w\in M} z_{ij}^{kw} &= g_{ij} && \forall i\in N, j\in N, && (5)\\
\sum_{r\in\Omega^k} a_r^i s_r - z_{ij}^{kw} &\geq 0 && \forall i\in N, j\in N, k\in M, w\in M, && (6)\\
\sum_{r\in\Omega^w} a_r^j s_r - z_{ij}^{kw} &\geq 0 && \forall i\in N, j\in N, k\in M, w\in M, && (7)
\end{aligned}$$

Variables' domain

$$\begin{aligned}
z_{ij}^{kw} &\geq 0 && \forall i\in N, j\in N, k\in M, w\in M, && (8)\\
s_r &\geq 0 && \forall r\in \Omega^k, k\in M. && (9)
\end{aligned}$$

Pricing subproblem

Let $[\pi, \mu, \pi_0, \theta, \eta]$ be the dual vector associated to constraints (2), (3), (4), (6) and (7).

Reduced cost of sequence $r_k \in \Omega^k$

$$\tilde{c}_{r_k} = -v_{r_k} - \pi_0^k - \sum_{i \in N} \pi^i \alpha_r^i - \sum_{h \in H} \mu^h q_r^h - \sum_{i,j \in N} \sum_{w \in M} \theta_{ij}^{kw} a_r^i - \sum_{i,j \in N} \sum_{w \in M} \eta_{ij}^{kw} a_r^j$$

Multiple pricing

- at each iteration, we have $|M|$ subproblems, one for every berth;
- the subproblem identifies the column r_k^* with the minimum reduced cost.

Column generation

- if $\tilde{c}_{r_k^*} < 0$ for some k, we add column r_k^* and we iterate;
- otherwise the current master problem solution is proven to be optimal.

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Pricing subproblem

- The pricing subproblem is an Elementary Shortest Path Problem with Resource Constraints (ESPP-RC).
- Generated sequences satisfy:
  - flow and precedence constraints (scheduling);
  - time windows constraints;
  - profile assignment constraints.
- The pricing problem is solved via dynamic programming.

Quality of bounds

Instance	Best solution	GAP wrt Lin.Rel.	GAP wrt Col.Gen.	Improvement
10Ap10	786’439	1.80%	1.42%	27.07%
10Ap20	785’637	1.94%	1.56%	24.69%
10Ap30	785’888	1.91%	1.56%	22.76%
10Ep10	732’101	1.21%	0.88%	37.38%
10Ep20	729’472	1.65%	1.33%	23.64%
10Ep30	729’173	1.69%	1.40%	20.56%
10Bp10	515’902	0.66%	0.47%	40.50%
10Bp20	515’991	0.65%	0.53%	21.90%
10Bp30	513’731	1.10%	0.98%	11.96%
10Cp10	564’831	0.59%	0.38%	54.83%
10Cp20	561’504	1.19%	0.98%	21.35%
10Cp30	559’389	1.58%	1.39%	13.89%

Quality of bounds

- the bound provided by the linear relaxation is relatively good;
- however, it is never improved by CPLEX even using cuts and/or the MIP solver;
- the proposed Dantzig-Wolfe reformulation improves the bound, and thus the gap, already at the root node;
- the further implementation of a *problem-specific branching scheme* is likely to close the gap faster than the generic branching scheme implemented in CPLEX;
- ongoing work: branch-and-price-and-cut algorithm for TBAP.

Pricing subproblem

Issues

- the underlying network has one node for every ship $i \in N$, for every quay crane profile $p \in P_i$ and for every time step $h \in H$;
- with a "standard" implementation, only small-size instances are solved in a reasonable time.

Accelerating strategies

- bi-directional dynamic programming;
- heuristic pricing;
- for every ship i, definition of the list of non-dominated (h, p) pairs;
- recursive column generation.

Recursive column generation

- We refer to the MILP by Giallombardo et al. (2010) as *compact formulation*.
- The decision variables of the TBAP compact formulation are:

$y_i^k \in \{0,1\}$	berth assignment;
$\lambda_i^p \in \{0,1\}$	qc profile assignment;
$x_{ij}^k \in \{0,1\}\,,\ \ T_i^k \geq 0$	ship scheduling.

- We focus on variables λ_i^p since the number of profiles has impact on the size of the network in the pricing problem.

Recursive column generation

Basic idea

- start solving the problem only with a meaningful subset of qc profiles $\hat{P} \subset P$;
- *dynamically* add the profitable profiles $p \in P \setminus \hat{P}$ that are missing.

Procedure: Recursive column generation

```
    Input : a meaningful subset of qc profiles $\hat{P} \subset P$.
    repeat
        repeat
CG1         add columns $s_r \in \hat{\Omega}$
        until optimal "partial" master problem ;
CG2     add compact-formulation variables $\lambda_i^p$ such that $p \in P \setminus \hat{P}$
    until optimal master problem ;
```

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Recursive column generation

Advantages

- the pricing problem is easier to solve;
- possibly many sub-optimal compact variables are left out from the formulation.

Remarks

- CG1 is standard column generation: the dual optimal vector is known at every iteration and reduced costs $\tilde{c}_{r_k}$ can be computed exactly;
- CG2 does not have any *direct* information available on the reduced cost of compact-formulation variables λ_i^p;
- We need the solution of the full pricing (or a valid bound to it) in order to compute a valid LB for the problem.

Main ingredients

Sub-optimal variable detection (Nemhauser and Wolsey, 1988)

If the reduced cost of a non-negative integer variable exceeds a given optimality gap, the variable must be zero in any optimal integer solution.

LB to reduced cost of compact-formulation variables (Irnich et al., 2010)

If the minimum reduced cost of all path variables of a DW master problem containing arc (i, j) exceeds a given optimality gap, no path that contains arc (i, j) can be used in an optimal solution. Hence, the arc (i, j) can be eliminated.

Other methods to compute reduced cost of compact-formulation variables:

- Walker (1969): the pricing problem must be a pure linear program.
- Poggi de Aragão and Uchoa (2003): coupling constraints in the master problem.

Lower bound to reduced cost of qc profiles

For a given ship $i \in N$ and a given profile $p \in P_i$, a lower bound to the reduced cost of the compact-formulation variable λ_i^p is provided by the minimum over all sequences $r \in \Omega$ that use the qc profile p (i.e., $\lambda_i^p = 1$). In other words:

$$LB(rc(\lambda_i^p)) = \min_{r \in \mathcal{F}_p} \tilde{c}_r \tag{10}$$

where $\mathcal{F}_p \subset \Omega$ represents the set of all feasible sequences that use profile p.

- This bound can be computed using dynamic programming, as proposed by Irnich et al. (2010).
- However, any lower bound to the minimum reduced cost $\tilde{c}_r$ of sequences $r \in \mathcal{F}_p$ can be used (although it may be weak).

Recursive column generation

Summing up

- if $LB(rc(\lambda_i^p)) > GAP$ then profile p is sub-optimal and can be eliminated;
- if $LB(rc(\lambda_i^p)) < 0$ then profile p is added to the formulation;
- if $0 \leq LB(rc(\lambda_i^p)) \leq GAP$ then profile p cannot be classified (eventually it is added to the formulation too).

Recursive column generation vs variable elimination

- we compare our method to the variable elimination of Irnich et al. (2010);
- in their approach, variable elimination is applied at the end of the root node, in order to speed up the branch-and-price algorithm;
- on the contrary, we dynamically add compact-formulation variables.

Preliminary experimental results

Rich VRP with $|N| = 25$ and compact-formulation variable similar to the TBAP profiles. Results for the root node.

		Irnich et al. 2010			Recursive C.G.					
Inst.	$\|P\|$	% sub	t(s)	# col	% sub	% act	% na	# it	t(s)	# col
A	125	40.80	0.46	409	33.60	20.00	46.40	9	3.03	424
B	125	37.60	0.62	681	36.00	20.80	43.20	8	1.41	**446**
C	125	31.20	1.53	906	26.40	20.00	53.60	9	3.76	**662**
D	125	0.00	1.09	872	0.00	20.00	80.00	11	13.91	**853**
E	175	44.00	1.30	628	36.57	14.29	49.14	8	5.52	**403**
F	175	40.00	1.60	941	30.29	14.86	54.86	11	8.30	**610**
G	175	34.86	4.06	1298	32.57	14.29	53.14	11	10.50	**747**
H	175	0.00	4.48	1542	0.00	14.29	85.71	14	105.00	**918**

Preliminary experimental results

- Validation of the proposed methodology: instances with sub-optimal compact-formulation variables are correctly detected.
- Interestingly, the number of columns generated to prove optimality is reduced.
- A lot of variables λ_i^p do not participate to prove optimality of the root node.
- Even with no eliminated variables, the total number of generated columns is smaller.
- We need to work on the computational time: in particular, # it CG2 should be at most 2 or 3 (since every CG2 iteration currently requires to solve a full pricing).
- The method is sensitive to initialization: in particular, we observed that only the last 2 iterations of CG2 produced suboptimal variables (when GAP < 6%).

Conclusion

- The proposed methodology is applicable to column generation itself (and not necessarily to a branch-and-price algorithm).
- The size of the pricing underlying network is increased at every iteration, but (almost) never reaches the "full" size.
- It need a faster implementation and it is worth for very complex problems where already the root node is difficult to solve.

Thanks for your attention!

References

Giallombardo, G., Moccia, L., Salani, M. and Vacca, I. (2010). Modeling and solving the tactical berth allocation problem, *Transportation Research Part B: Methodological* **44(2)**: 232–245.

Irnich, S., Desaulniers, G., Desrosiers, J. and Hadjar, A. (2010). Path reduced costs for eliminating arcs, *Journal on Computing* **22**(2): 297–313.

Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, N. Y.

Poggi de Aragão, M. and Uchoa, E. (2003). Integer program reformulation for robust branch-and-price algorithms, *Proceedings of Mathematical Programming in Rio: A conference in honour of Nelson Maculan*, pp. 56–61.

Steenken, D., Voss, S. and Stahlbock, R. (2004). Container terminal operation and operations research - a classification and literature review, *OR Spectrum* **26**: 3–49.

Walker, W. E. (1969). A method for obtaining the optimal dual solution to a linear program using the Dantzig-Wolfe decomposition, *Operations Research* **17**: 368–370.