Recent results for column generation based diving heuristics

Ruslan Sadykov, François Vanderbeck, Artur Pessoa, Eduardo Uchoa, Issam Tahiri

Recent results for column generation based diving heuristics

Ruslan Sadykov

Inria Bordeaux — Sud-Ouest and Bordeaux University, France

informatiques mathématiques

université de BORDEAUX

Joint work with François Clautiaux, Matthieu Gérard, Artur Pessoa, Issam Tahiri, François Vanderbeck, Eduardo Uchoa

Buzios, Brazil, May 24, 2016

Table of contents

Contents

The Branch-and-Price approach

Assume a bounded integer program with decomposable structure:

$$
\begin{array}{rrcl}
[\mathrm{P}] \equiv \min & c\,x & : & \\
 & A\,y & \geq & a \\
 & y & = & \sum_{k \in \mathcal{K}} x^k \\
x^k \in X^k = \{ & B^k\,x^k & \geq & b^k \\
 & x^k & \in & \mathbb{N}^{n(k)} \quad \}, \quad \forall k \in \mathcal{K}
\end{array}
$$

Assume that subproblems

$$[\mathrm{SP}]^k \equiv \min\{c\,x^k : \; x^k \in X^k\} \tag{1}$$

are "relatively easy" to solve compared to problem [P]. Then,

$$
\begin{array}{rcl}
X^k & = & \{z^q\}_{q \in Q(k)} \\
\mathrm{conv}(X^k) & = & \left\{ x^k \in \mathbb{R}_+^{n(k)} : \sum_{q \in Q(k)} z^q \lambda_q, \; \sum_{q \in Q(k)} \lambda_q = 1, \; \lambda_q \geq 0 \; q \in Q(k) \right\}
\end{array}
$$

The Branch-and-Price approach (2)

Reformulation as the master program (Dantzig-Wolfe reformulation):

$$
\begin{aligned}
[\mathrm{M}] \equiv \min \quad \sum_{k\in\mathcal{K}} \sum_{q\in Q(k)} (cz^q)\, \lambda_q^k \quad &: \\
\sum_{k\in\mathcal{K}} \sum_{q\in Q(k)} (Az^q)\lambda_q^k &\geq a \\
\sum_{q\in Q(k)} \lambda_q^k &= 1, \quad \forall k \in \mathcal{K} \\
\lambda_q^k &\in \{0,1\}, \quad k \in \mathcal{K}, q \in Q(k).
\end{aligned}
$$

Aggregation of identical blocks in $\mathcal{K}$:

$$
\begin{aligned}
[\mathrm{AM}] \equiv \min \quad \sum_{q\in Q} (cz^q)\, \lambda_q \quad &: \\
\sum_{q\in Q} (Az^q)\lambda_q &\geq a \\
\sum_{q\in Q} \lambda_q &= K, \\
\lambda_q &\in \mathbb{N}, \quad q \in Q.
\end{aligned}
$$

Contents

Rounding heuristics in λ var space

Rounding a variable $\lambda_q \rightarrow$ new dual variable

- Adding upper bound $\boxed{\lambda_q \leq u_q}$:
  If dual variable is ignored, λ_q might be wrongly regenerated as best.
  If enforced, significant modifications to pricing.
- Adding lower bound $\boxed{\lambda_q \geq l_q}$:
  If ignored, λ_q's reduced cost is overestimated, hence not regenerated
- Adapted to Column Generation: if one only uses $\boldsymbol{\lambda_q \geq l_q}$

Remark

Fixing $\lambda_q \leftarrow \lceil \bar{\lambda}_q \rfloor$ as a **partial solution** is equivalent to setting a lower bound on λ_q

Diving heuristics in λ var space

The residual master problem may become infeasible after rounding, as

- the partial solution may not satisfy the master constraints;
- the partial solution may not be completed with columns generated so far.

Solution 1

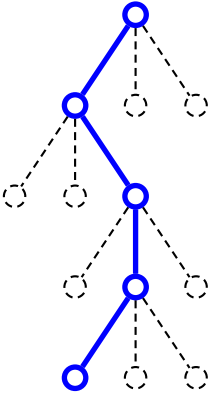
One should work with **proper columns**, i.e. columns that could take a non-zero value in a master integer solution (may be harder to price such columns).

Solution 2

Diving, i.e. further column generation after rounding is a generic way to restore feasibility, i.e. to generate "missing" complementary columns.

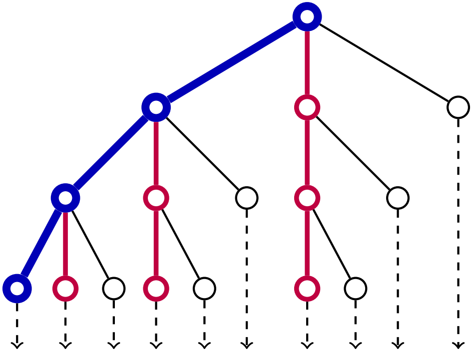
Pure Diving

- use Depth-First Search
- at each node of the tree
  - select a column with its fractional value $\bar{\lambda}_q$ closest to a non-zero integer
  - add $\lceil \bar{\lambda}_q \rfloor$ to the partial solution
  - update right-hand-side of the master constraints
  - apply preprocessing which results in removing non-proper columns
  - solve the updated master LP
- repeat until a complete feasible solution is found or until the master LP is infeasible



Diving with Limited Discrepancy Search

Idea: add some diversification through limited backtracking (Limited Discrepancy Search by [Harvey and Ginsberg, 1995])



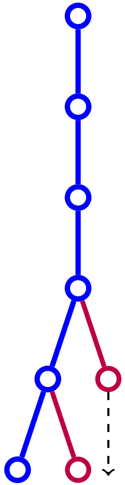
MaxDiscrepancy = 2, **MaxDepth = 3**

At each node, we have a tabu list of columns forbidden to be added to the partial solution.

Variants of Diving with LDS

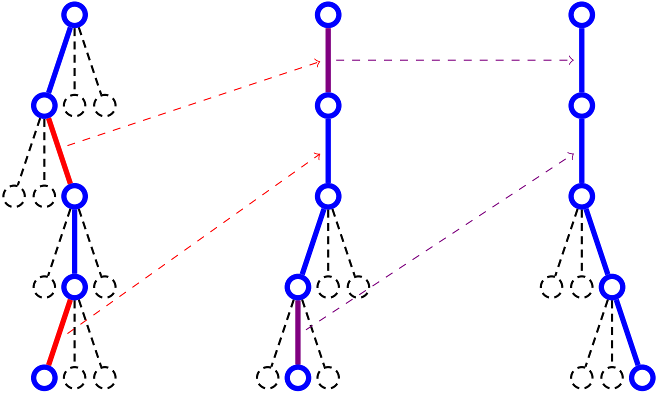
- **Diving for feasibility**
  We are doing backtracking in diving until a feasible solution is found, corresponds to Diving with LDS with parameters **MaxDiscrepancy = 1**, **MaxDepth =** ∞

- **Strong Diving**
  The candidate columns for selection are evaluated (as in strong branching). We choose a candidate which deteriorates the least the column generation bound.



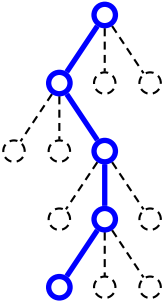
Diving with Restarts

- Keep a fraction of columns participating in the best solution
- Remove other columns from the solution
- Restart diving
- Resembles Relaxation Induced Neighbourhood Search [Danna et al., 2005].

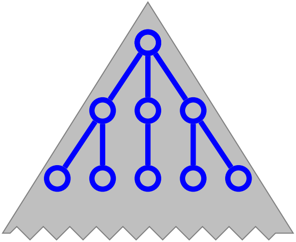


Diving with sub-MIPing

Run Diving



Run Restricted Master Heuristic with all columns generated during diving



A variant with "local branching" [Fischetti and Lodi, 2003]

The following constraint is added to the restricted master:

$$\sum_{q \in Q^{\text{inc}}} \lambda_q \geq r^* - \lceil r^* \cdot \textit{deviationRatio} \rceil, \text{ where } r^* = \sum_{q \in Q^{\text{inc}}} \lambda_q^{\text{inc}}$$

Contents

Test problems and instances

Master is always the set covering formulation

Generalized Assignment

- Pricing : multiple distinct $0-1$ knapsack problems
- Instances of the most difficult in literature type D with (number of tasks, number of machines) in $\{(90, 18), (160, 8)\}$

Bin Packing

- Pricing : multiple identical $0-1$ knapsack problems
- Instances of the most difficult (for heuristics) type AI [Delorme et al., 2016] with number of items in $\{201, 402\}$.

Vertex Coloring

- Pricing : multiple identical weighted stable set problems
- Random instances with number of vertices in $\{50, \dots, 90\}$

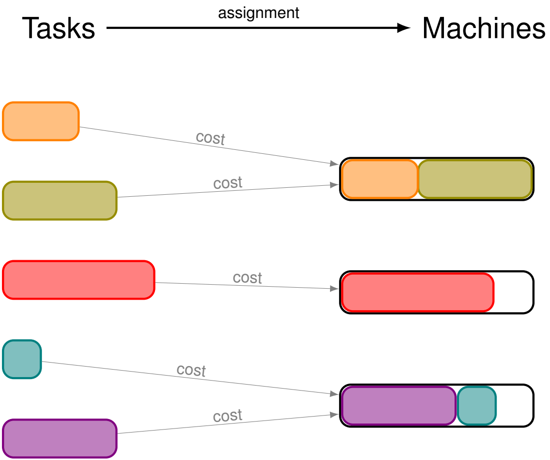
Comparison of heuristics

Average gap is relative for Generalized Assignment and absolute for Bin Packing and Vertex Coloring

	Generalized Assignment			Bin Packing			Vertex Coloring		
Heuristic	Time	Found	Gap	Time	Opt	Gap	Time	Opt	Gap
Restricted Master	26.50	55%	11.00%	224.37	5%	1.22	3.94	49%	0.54
Pure Diving	0.80	70%	0.37%	13.71	46%	0.54	0.94	71%	0.29
Diving for Feasibility	0.81	100%	0.39%	↑ same ↑			↑ same ↑		
Diving + SubMIPing	40.22	100%	0.38%	85.49	53%	0.47	1.93	81%	0.19
Local Branching	1.90	100%	0.38%	44.40	52%	0.48	1.00	74%	0.26
Diving with Restarts	1.52	100%	0.24%	14.83	51%	0.49	1.06	74%	0.26
Diving with LDS	4.21	100%	0.10%	27.44	89%	0.11	1.38	88%	0.12
Strong Diving	33.45	100%	0.05%	67.42	90%	0.10	3.65	94%	0.06

Contents

Generalized Assignment: Description



Pricing oracle: $0-1$ knapsack problem
(solver by [Pisinger, 1997])

Comparison with the best heuristic in the literature

- Classic literature instances
- Critical to use heavy stabilization ([Pessoa et al., 2014])
- Times are “normalised”

	[Yagiura et al., 2006]				Diving heuristic with LDS			
Group	Time	Opt	RelGap	Gap	Time	Opt	RelGap	Gap
Type C	145.1	53%	0.010%	0.9	30.0	47%	0.015%	0.7
Type D	145.1	7%	0.103%	21.1	69.5	7%	0.047%	8.5
Type E	145.1	33%	0.013%	6.7	38.1	47%	0.014%	3.2
$n = 100$	9.4	67%	0.073%	4.8	1.4	44%	0.045%	3.4
$n = 200$	18.8	44%	0.045%	5.3	6.1	11%	0.054%	6.0
$n = 400$	187.5	33%	0.051%	12.8	40.2	44%	0.017%	4.1
$n = 900$	625.0	0%	0.029%	14.7	291.1	33%	0.006%	3.0
$n = 1600$	3125.0	11%	0.011%	10.2	1500.7	33%	0.006%	4.1
high n/m	145.1	47%	0.006%	2.5	19.1	33%	0.023%	3.3
med n/m	145.1	27%	0.031%	7.1	46.7	27%	0.025%	3.7
low n/m	145.1	33%	0.089%	19.1	88.7	40%	0.029%	5.3
All	145.1	31%	0.042%	9.6	43.0	33%	0.026%	4.1

GAP: Results for large open instances

- Best known bounds and solutions are from [Posta et al., 2012]
- Seven runs with different col. gen. parameters
- 3 hours time limit

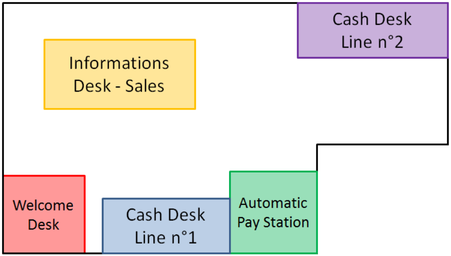
	Best known		Best run			Average	
Instance	Bound	Solution	Solution	Time	Red. gap	Time	Red. gap
D-20-200	12235	12244	**12238**	$<$1m	66%	$<$1m	3%
D-20-400	24563	24585	**24567**	1m	82%	1m	56%
D-40-400	24350	24417	**24356**	2m	89%	2m	72%
D-15-900	55404	55414	**54404**	1m	100%	3m	43%
D-30-900	54834	54868	**54838**	9m	88%	8m	61%
D-60-900	54551	54606	**54554**	24m	95%	25m	83%
D-20-1600	97824	97837	**97825**	12m	92%	11m	69%
D-40-1600	97105	97113	**97105**	53m	100%	2h03m	38%
D-80-1600	97034	97052	**97035**	3h00m	94%	3h00m	-48%
C-80-1600	16284	16289	**16285**	36m	80%	43m	80%

Contents

Tour Scheduling: Description

A tour scheduling problem

1. **Needs:** to perform at best a limited list of activities (workload) during a planning horizon (a week).
2. **Human Ressources:** list of employees with skills, individualised contract and personal preferences/obligations.



Main objective

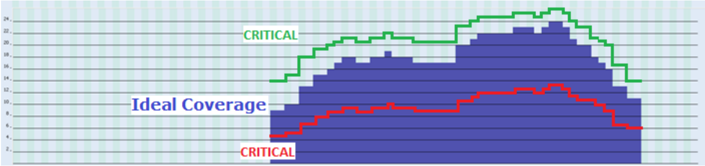
[Chan, 2002]

To design a JuSTE planning: Juridical, Social, Technical, Economical.

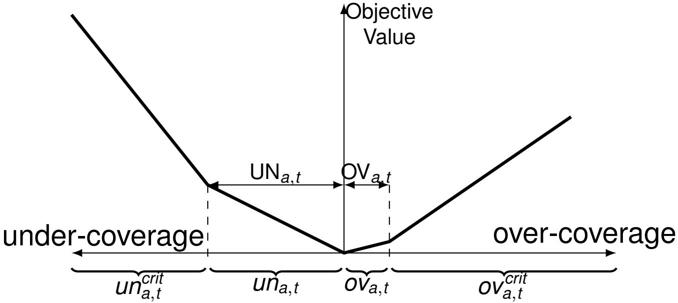
$\rightarrow$ feasibility and optimisation problem.

Tour Scheduling: Objective Function

Daily workload for a production activity (time period = 15 min)



Piecewise linear cost function for each period - production activity



Tour Scheduling: Formulation

- $\mathcal{T}$ — set of time periods, $\mathcal{A}$ — set of activities
- $\mathcal{C}(e)$ — set of feasible individual plannings for employee $e \in \mathcal{E}$

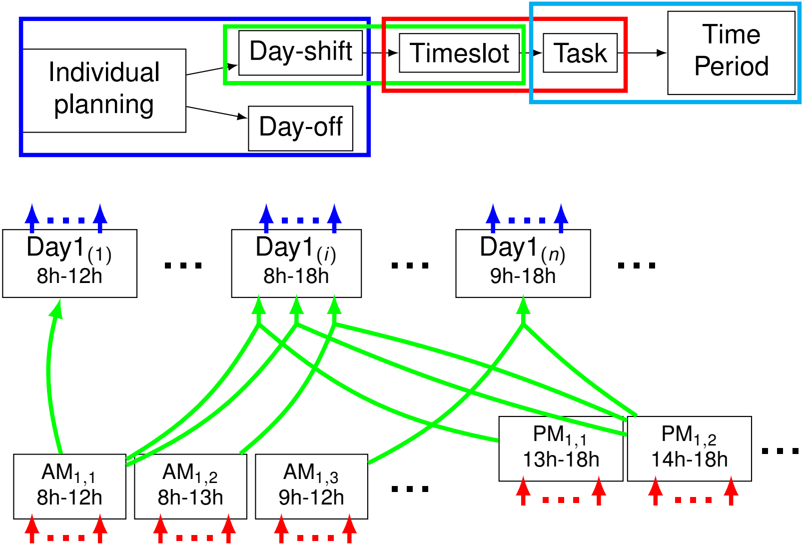
$$
\begin{aligned}
\min \sum_{a\in\mathcal{A}} \sum_{t\in\mathcal{T}} \mathrm{CO}_a \cdot ov_{a,t} + \mathrm{CU}_a \cdot un_{a,t} & \\
\text{s.t.} \sum_{e\in\mathcal{E}} \sum_{c\in\mathcal{C}(e)} x_{c,a,t}\lambda_c - ov_{a,t} + un_{a,t} = \mathrm{DE}_{a,t} & \qquad \forall t \in \mathcal{T}, \forall a \in \mathcal{A} \\
\sum_{c\in\mathcal{C}(e)} \lambda_c = 1 & \qquad \forall e \in \mathcal{E} \\
\lambda_c \in \{0,1\} & \qquad \forall e \in \mathcal{E}, \forall c \in \mathcal{C}(e) \\
un_{a,t}, ov_{a,t} \in \mathbb{R}_+ & \qquad \forall t \in \mathcal{T}, \forall a \in \mathcal{A}
\end{aligned}
$$

Pricing problem for employee $e \in \mathcal{E}$

Construct a feasible individual planning with objective $\sum_{a\in\mathcal{A}, a\in\mathcal{T}} \pi_{a,t} x_{a,t}$, where binary variable $x_{a,t}$ determines whether activity a is performed at time period t, and π are reduced costs

Tour Scheduling: Pricing Oracle

4 Segmentations in our nested dynamic program (5 levels)



Tour Scheduling: results for customer instances

- Greedy heuristic time is from 0.3 to 2.3 seconds
- Diving times are 2, 10, and 30 minutes
- The best solutions were obtained by a heuristic Branch and Price with 24 hours time limit

Instance		Gap with "easily computable" lower bound				
$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{A}\rvert$	Greedy	Diving 2	Diving 10	Diving 30	Best sol.
5	1	21.1%	4.0%	4.0%	4.0%	0.9%[1]
5	1	41.5%	0.0%	0.0%	0.0%	0.0%[1]
10	1	34.4%	0.0%	0.0%	0.0%	0.0%[1]
10	1	44.2%	0.9%	0.9%	0.9%	0.9%[1]
25	3	31.3%	1.5%	1.7%	1.5%	0.5%
25	3	24.2%	3.5%	0.9%	3.8%	0.3%
30	3	37.3%	3.6%	2.8%	1.9%	1.9%
30	3	90.8%	21.3%	12.9%	11.3%	8.4%
45	5	13.9%	1.2%	0.4%	0.0%	0.0%[1]
45	5	18.9%	1.4%	1.1%	1.9%	0.3%
Average		35.8%	3.7%	2.5%	2.5%	1.3%

[1] Optimum

Tour Scheduling: variant with 4 weeks horizon

Column generation

Restricted master becomes too heavy for the LP solver.

Solution

Use sub-gradient instead!

- Find good Lagrangian multipliers (within time limit)
- Generate a pricing problem solution with these multipliers
- Fix this partial solution and iterate

Results for a hard instance with $|\mathcal{E}| = 15$, $|\mathcal{A}| = 2$

	Greedy solution gap reduction with time limit of		
Algorithm	8 min	40 min	2 hours
Diving with ColGen	3%	5%	17%
Diving with SubGrad	10%	74%	79%

Contents

Conclusions

- **Seven variants** of generic diving heuristics were tested on **three different problems**
- All this variants are **significantly better than** the most used in the literature **Restricted Master Heuristic**
- Such **generic** primal heuristics may **outperform** ad-hoc heuristics of the literature.
- Rounding/Diving based on fixing master var. **works when**
  1. Sufficiently many columns in the solution:
     $$\sum_k \sum_{q \in Q(k)} \lambda_q = K \quad \text{with} \quad K \gg 1$$
  2. Column generation (Lagrangian) bound is tight
  3. Most of the combinatorial difficulty is in the subproblem

This presentation is based on

Sadykov, R., Vanderbeck, F., Pessoa, A., Tahiri, I., and Uchoa, E. (2015).
Primal heuristics for Branch-and-Price.
Technical Report hal-01237204, HAL Inria.

Gérard, M., Clautiaux, F., and Sadykov, R. (2016).
Column generation based approaches for a tour scheduling problem with a multi-skill heterogeneous workforce.
European Journal of Operational Research, 252(3):1019 – 1030.

References I

Chan, Y.-C. P. (2002).
La planification du personnel: acteurs, actions et termes multiples pour une planification opérationnelle des personnes.
PhD thesis, Université Joseph-Fourier-Grenoble I.

Danna, E., Rothberg, E., and Pape, C. L. (2005).
Exploring relaxation induced neighborhoods to improve MIP solutions.
Mathematical Programming, 102(1, Ser. A):71–90.

Delorme, M., Iori, M., and Martello, S. (2016).
Bin packing and cutting stock problems: Mathematical models and exact algorithms.
European Journal of Operational Research, accepted:–.

Fischetti, M. and Lodi, A. (2003).
Local branching.
Mathematical Programming, 98(1-3, Ser. B):23–47.

References II

Harvey, W. D. and Ginsberg, M. L. (1995).
Limited discrepancy search.
In *Proc. IJCAI-95, Montreal, Quebec*, pages 607–613. Morgan Kaufmann.

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2014).
Automation and combination of linear-programming based stabilization techniques in column generation.
Technical Report hal-01077984, HAL Inria.

Pisinger, D. (1997).
A minimal algorithm for the 0-1 knapsack problem.
Operations Research, 45(5):758–767.

Posta, M., Ferland, J. A., and Michelon, P. (2012).
An exact method with variable fixing for solving the generalized assignment problem.
Computational Optimization and Applications, 52:629–644.

References III

Yagiura, M., Ibaraki, T., and Glover, F. (2006).

A path relinking approach with ejection chains for the generalized assignment problem.

European Journal of Operational Research, 169(2):548 – 569.