Recent results for column generation based diving heuristics

Ruslan Sadykov

Inria Bordeaux — Sud-Ouest and Bordeaux University, France

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

Buzios, Brazil, May 24, 2016
Table of contents

Introduction

Various heuristics based on diving in $\lambda$ variables space

Computational comparison of heuristics

Computational results for Generalized Assignment

Computational results for Multi-Activity Tour Scheduling

Conclusions
Contents

Introduction

Various heuristics based on diving in $\lambda$ variables space

Computational comparison of heuristics

Computational results for Generalized Assignment

Computational results for Multi-Activity Tour Scheduling

Conclusions
The Branch-and-Price approach

Assume a bounded integer program with decomposable structure:

\[
[P] \equiv \min c x : \\
A y \geq a \\
y = \sum_{k \in K} x^k
\]

\[
x^k \in X^k = \{ B^k x^k \geq b^k \\
x^k \in \mathbb{N}^{n(k)} \ \}, \ \forall k \in K
\]

Assume that subproblems

\[
[SP]^k \equiv \min \{ c x^k : x^k \in X^k \}
\]

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

\[
X^k = \{ z^q \}_{q \in Q(k)}
\]

\[
\text{conv}(X^k) = \left\{ x^k \in \mathbb{R}^{n(k)}_+ : \sum_{q \in Q(k)} z^q \lambda_q, \sum_{q \in Q(k)} \lambda_q = 1, \lambda_q \geq 0 \ q \in Q(k) \right\}
\]
The Branch-and-Price approach (2)

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

\[ [M] \equiv \min \sum_{k \in \mathcal{K}} \sum_{q \in Q(k)} (cz^q) \lambda_k^q : \]
\[ \sum_{k \in \mathcal{K}} \sum_{q \in Q(k)} (Az^q) \lambda_k^q \geq a \]
\[ \sum_{q \in Q(k)} \lambda_k^q = 1, \quad \forall k \in \mathcal{K} \]
\[ \lambda_k^q \in \{0, 1\}, \quad k \in \mathcal{K}, \quad q \in Q(k). \]

Aggregation of identical blocks in \( \mathcal{K} \):

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

Introduction

Various heuristics based on diving in \( \lambda \) variables space

Computational comparison of heuristics

Computational results for Generalized Assignment

Computational results for Multi-Activity Tour Scheduling

Conclusions
Rounding heuristics in $\lambda$ var space

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

- Adding upper bound $\lambda_q \leq u_q$:
  If dual variable is ignored, $\lambda_q$ might be wrongly regenerated as best.
  If enforced, significant modifications to pricing.

- Adding lower bound $\lambda_q \geq l_q$:
  If ignored, $\lambda_q$’s reduced cost is overestimated, hence not regenerated

- Adapted to Column Generation: if one only uses $\lambda_q \geq l_q$

Remark
Fixing $\lambda_q \leftarrow \lfloor \bar{\lambda}_q \rfloor$ as a partial solution is equivalent to setting a lower bound on $\lambda_q$
Diving heuristics in $\lambda$ 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 $\lambda_q$ closest to a non-zero integer
  - add $\lceil \lambda_q \rceil$ 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
  \( \text{MaxDiscrepancy} = 1, \text{MaxDepth} = \infty \)

- **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:

\[
\sum_{q \in Q^{inc}} \lambda_q \geq r^* - \lfloor r^* \cdot \text{deviationRatio} \rfloor, \quad \text{where} \quad r^* = \sum_{q \in Q^{inc}} \lambda_q^{inc}
\]
Contents

Introduction

Various heuristics based on diving in $\lambda$ variables space

Computational comparison of heuristics

Computational results for Generalized Assignment

Computational results for Multi-Activity Tour Scheduling

Conclusions
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 Al [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, . . . , 90}
Comparison of heuristics

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

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Generalized Assignment</th>
<th>Bin Packing</th>
<th>Vertex Coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restricted Master</td>
<td>26.50</td>
<td>55%</td>
<td>11.00%</td>
</tr>
<tr>
<td>Pure Diving</td>
<td>0.80</td>
<td>70%</td>
<td>0.37%</td>
</tr>
<tr>
<td>Diving for Feasibility</td>
<td>0.81</td>
<td>100%</td>
<td>0.39%</td>
</tr>
<tr>
<td>Diving + SubMIPing</td>
<td>40.22</td>
<td>100%</td>
<td>0.38%</td>
</tr>
<tr>
<td>Local Branching</td>
<td>1.90</td>
<td>100%</td>
<td>0.38%</td>
</tr>
<tr>
<td>Diving with Restarts</td>
<td>1.52</td>
<td>100%</td>
<td>0.24%</td>
</tr>
<tr>
<td>Diving with LDS</td>
<td>4.21</td>
<td>100%</td>
<td>0.10%</td>
</tr>
<tr>
<td>Strong Diving</td>
<td>33.45</td>
<td>100%</td>
<td>0.05%</td>
</tr>
</tbody>
</table>
Contents

Introduction

Various heuristics based on diving in $\lambda$ variables space

Computational comparison of heuristics

Computational results for Generalized Assignment

Computational results for Multi-Activity Tour Scheduling

Conclusions
Generalized Assignment: Description

Tasks → assignment → Machines

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”

<table>
<thead>
<tr>
<th>Group</th>
<th>[Yagiura et al., 2006]</th>
<th>Diving heuristic with LDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Opt</td>
</tr>
<tr>
<td>Type C</td>
<td>145.1</td>
<td>53%</td>
</tr>
<tr>
<td>Type D</td>
<td>145.1</td>
<td>7%</td>
</tr>
<tr>
<td>Type E</td>
<td>145.1</td>
<td>33%</td>
</tr>
<tr>
<td>n = 100</td>
<td>9.4</td>
<td>67%</td>
</tr>
<tr>
<td>n = 200</td>
<td>18.8</td>
<td>44%</td>
</tr>
<tr>
<td>n = 400</td>
<td>187.5</td>
<td>33%</td>
</tr>
<tr>
<td>n = 900</td>
<td>625.0</td>
<td>0%</td>
</tr>
<tr>
<td>n = 1600</td>
<td>3125.0</td>
<td>11%</td>
</tr>
<tr>
<td>high n/m</td>
<td>145.1</td>
<td>47%</td>
</tr>
<tr>
<td>med n/m</td>
<td>145.1</td>
<td>27%</td>
</tr>
<tr>
<td>low n/m</td>
<td>145.1</td>
<td>33%</td>
</tr>
<tr>
<td>All</td>
<td>145.1</td>
<td>31%</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Instance</th>
<th>Best known</th>
<th>Best run</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bound</td>
<td>Solution</td>
<td>Solution</td>
</tr>
<tr>
<td>D-20-200</td>
<td>12235</td>
<td>12244</td>
<td>12238</td>
</tr>
<tr>
<td>D-20-400</td>
<td>24563</td>
<td>24585</td>
<td>24567</td>
</tr>
<tr>
<td>D-40-400</td>
<td>24350</td>
<td>24417</td>
<td>24356</td>
</tr>
<tr>
<td>D-15-900</td>
<td>55404</td>
<td>55414</td>
<td>54404</td>
</tr>
<tr>
<td>D-30-900</td>
<td>54834</td>
<td>54868</td>
<td>54838</td>
</tr>
<tr>
<td>D-60-900</td>
<td>54551</td>
<td>54606</td>
<td>54554</td>
</tr>
<tr>
<td>D-20-1600</td>
<td>97824</td>
<td>97837</td>
<td>97825</td>
</tr>
<tr>
<td>D-40-1600</td>
<td>97105</td>
<td>97113</td>
<td>97105</td>
</tr>
<tr>
<td>D-80-1600</td>
<td>97034</td>
<td>97052</td>
<td>97035</td>
</tr>
<tr>
<td>C-80-1600</td>
<td>16284</td>
<td>16289</td>
<td>16285</td>
</tr>
</tbody>
</table>
Contents

Introduction

Various heuristics based on diving in $\lambda$ variables space

Computational comparison of heuristics

Computational results for Generalized Assignment

Computational results for Multi-Activity Tour Scheduling

Conclusions
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 Resources:** list of employees with skills, individualised contract and personal preferences/obligations.

**Main objective**

[Chan, 2002]

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

→ 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 \( e \in \mathcal{E} \)

\[
\min \sum_{a \in \mathcal{A}} \sum_{t \in \mathcal{T}} \text{CO}_a \cdot \text{ov}_{a,t} + \text{CU}_a \cdot \text{un}_{a,t}
\]

s.t. \[
\sum_{e \in \mathcal{E}} \sum_{c \in \mathcal{C}(e)} x_{c,a,t} \lambda_c - \text{ov}_{a,t} + \text{un}_{a,t} = \text{DE}_{a,t} \quad \forall t \in \mathcal{T}, \forall a \in \mathcal{A}
\]

\[
\sum_{c \in \mathcal{C}(e)} \lambda_c = 1 \quad \forall e \in \mathcal{E}
\]

\[
\lambda_c \in \{0, 1\} \quad \forall e \in \mathcal{E}, \forall c \in \mathcal{C}(e)
\]

\[
\text{un}_{a,t}, \text{ov}_{a,t} \in \mathbb{R}^+ \quad \forall t \in \mathcal{T}, \forall a \in \mathcal{A}
\]

Pricing problem for employee \( e \in \mathcal{E} \)

Construct a feasible individual planning with objective

\[
\sum_{a \in \mathcal{A}} \sum_{t \in \mathcal{T}} \pi_{a,t} x_{a,t}
\]

where binary variable \( x_{a,t} \) determines whether activity \( a \) is performed at time period \( t \), and \( \pi \) 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

<table>
<thead>
<tr>
<th>Instance</th>
<th>Gap with “easily computable” lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>Average</td>
<td>35.8%</td>
</tr>
</tbody>
</table>

$^1$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 $|\mathcal{E}| = 15$, $|\mathcal{A}| = 2$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Greedy solution gap reduction with time limit of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 min</td>
</tr>
<tr>
<td>Diving with ColGen</td>
<td>3%</td>
</tr>
<tr>
<td>Diving with SubGrad</td>
<td>10%</td>
</tr>
</tbody>
</table>
Contents

Introduction

Various heuristics based on diving in $\lambda$ variables space

Computational comparison of heuristics

Computational results for Generalized Assignment

Computational results for Multi-Activity Tour Scheduling

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:
\[ \sum_k \sum_{q \in Q(k)} \lambda_q = K \text{ with } K \gg 1 \]
2. Column generation (Lagrangian) bound is tight
3. Most of the combinatorial difficulty is in the subproblem
This presentation is based on

Primal heuristics for Branch-and-Price.
Technical Report hal-01237204, HAL Inria.

Column generation based approaches for a tour scheduling problem with a multi-skill heterogeneous workforce.
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.

Exploring relaxation induced neighborhoods to improve MIP solutions.

Bin packing and cutting stock problems: Mathematical models and exact algorithms.
European Journal of Operational Research, accepted:–.

Local branching.

