Matheuristics to stabilize column generation: application to a technician routing problem

Nicolas DUPIN, Rémi PARIZE, El-Ghazali TALBI

Univ. Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille
nicolas.dupin@polytechnique.edu

Matheuristics 2018
Tours, the 18th of June 2018
Problem statement and related state-of-the-art

Extended formulation and Column Generation matheuristic

POPMUSIC column generation stabilization

Tabu Search Matheuristic intensification

Computational results
Outline

Problem statement and related state-of-the-art

Extended formulation and Column Generation matheuristic

POPMUSIC column generation stabilization

Tabu Search Matheuristic intensification

Computational results
Minimizing the total length of the routes, having several vehicles to visit customers in defined time-windows.
Meta-heuristic resolution of VRPTW, state-of-the-art

- Large diversity of successful meta-heuristic to solve VRPTW:
  - Tabu search (Gendreau et al 1994, Potvin et al 1996, Cordeau 2001)
  - GRASP (Kontoravdis et al 1995)
  - Simulated Annealing (Chiang et al 1996)
  - Variable Neighborhood Search (Braysy 2003, Polacek et al 2004)
  - Large Neighborhood Search (Christensen et al 2016)

- Difficulties with pure meta-heuristics: highly-constrained optimization problems

- Exact approaches to solve VRPs:
  - Compact MIP formulations inefficient with poor LP relaxations
  - Extended MIP formulations with an exponential number of cuts and Branch&Cut: Kallehauge et al 2007
  - Extended MIP formulations with an exponential number of variables and column generation resolution.

- For rich VRP, emergence of matheuristics to handle constraint feasibilities. Column generation matheuristics often efficient.
Specificities of our problem

- Similarities with the challenge ROADEF 2007: optimizing routes of technicians to make reparations interventions by customers in defined Time Windows
- Multi depot: Technicians start from different places (company places or at home)
- Time windows for technicians: working day constraints
- Skill constraints: jobs require specific skill that technician must have.
- Penalities: it is possible to skip some jobs, which adds penalties in the objective function

Similar problems in the literature:

Implications on the VRP structure

- Time windows for technicians: imply that the routes of technician $i$ are limited to a small number of jobs, denoted $K_i$.

- Skill constraints: limitations of the jobs a vehicle can reach whereas the graph of commutations is usually a clique for classical VRP.

- Combinatorics, lower than the usual clique VRP: helpful for exact methods.

- Make the problem highly-constrained: sometimes difficult to construct solutions reaching the maximal number of jobs.
Notations

$I$ Set of technicians.
$J$ Set of customers, the jobs for the technicians.
$J_i$ Subset of jobs that technician $i$ can complete (skill constraints).
$C_i^j$ $C_i^j = 1$ iff $i$ has the skill to realize job $j$.
$D_j$ Duration of job $j$.
$P_j$ Cost penalization if job $j$ is not planned.
$d_i$ Starting and finishing depot for technician $i$.
$d(j, j')$ Distance from the place of job $j$ to its of $j'$.
$d(i, j)$ Distance from the place of job $j$ to $d_i$.
$t(j, j')$ Transportation time from the place of job $j$ to its of $j'$.
$t(i, j)$ Transportation time from the place of job $j$ to $d_i$.
$[\bar{t}_{\text{start}}^i, \bar{t}_{\text{end}}^i]$ Working time window for technician $i$.
$[\bar{t}_{\text{min}}^j, \bar{t}_{\text{max}}^j]$ Time windows to begin the job $j \in J$. 

[The table above would list all the notations and their definitions in a clear and structured manner, but it is not included here.]
First definition of variables

\[ u^i_{j,j'} \in \{0, 1\} \text{ with } u^i_{j,j'} = 1 \text{ iff technician } i \text{ realizes job } j \text{ and just after job } j'. \]
\[ (u^i_{j,j} = 0). \]

\[ u^i_{d_{i,j}} \quad (\text{resp } u^i_{d_{i,d_i}} \text{ respectively}) \text{ denotes first (last resp) job } j \text{ for the technician } i, \text{ et } u^i_{d_{i,j}} = 0 \text{ si } i \neq i'. \]

Continuous variables \( t_j \) to denote the beginning hour of job \( j \):

\[ \forall j \in J, \tilde{t}^\min_j \leq t_j \leq \tilde{t}^\max_j \]
First MIP compact formulation

\[
\begin{align*}
\min & \quad \sum_{j,j'} d(j,j') u^i_{j,j'} + \sum_j P_j \left( 1 - \sum_{i} \sum_{j'} u^i_{j,j'} \right) \\
\forall i, j, & \quad \sum_{j' \in J \cup \{d_i\}} u^i_{j',j} = \sum_{j' \in J \cup \{d_i\}} u^i_{j,j'} \\
\forall i, j, & \quad \sum_{j' \in J \cup \{d_i\}} u^i_{j',j} \leq C^j_i \\
\forall i, & \quad \sum_{j' \in I} u^j_{j',d_i} = \sum_{j' \in I} u^j_{d_i,j'} \leq 1 \\
\forall (j, j'), & \quad t_j + D_i + T(j, j') \leq t_{j'} + \left( 1 - \sum_{i} u^i_{j,j'} \right) \cdot M \\
\forall i, j, & \quad \tilde{t}_i^{\text{start}} + T(d_i, j) \leq t_j + \left( 1 - u^i_{d_i,j} \right) \cdot M \\
\forall i, j, & \quad t_j + D_j + T(j, d_i) \leq \tilde{t}_i^{\text{end}} + \left( 1 - u^i_{j,d_i} \right) \cdot M \\
\forall i, j, j', & \quad u^i_{j,j'} \in \{0, 1\}, \ t_j \in [\tilde{t}_j^{\text{min}}, \tilde{t}_j^{\text{max}}]
\end{align*}
\]

+ cuts to improve the weaknesses of big M constraint formulation
Local optimizations are here defined with variable fixing strategies on binaries 
\[ a_{ij} = \sum_{j' \in J \cup \{d_i\}} u_{ijj'} \in \{0, 1\} \] indicating job affectation:

- Variable fixing strategies? Bottleneck: poor quality of LP relaxation
- Some efficient greedy strategies to have accurately solutions placing the maximal number of jobs.
- Efficient VND scheme with MIP neighborhoods to improve previous solutions
- Benchmark with LocalSolver on specific instances studying impact of constraints in graduated instances with inclusions in feasibility sets and constraint difficulty.
- Column generation (CG) scheme and matheuristics: gives excellent dual bounds and primal heuristics.
- Issue motivating this work: CG convergence too long for large instances.
Outline

Problem statement and related state-of-the-art

Extended formulation and Column Generation matheuristic

POPMUSIC column generation stabilization

Tabu Search Matheuristic intensification

Computational results
We have an extended formulation by enumerating all possible routes $\mathcal{P}_i$ for all technician $i$. The cost of a route $k \in \mathcal{P}_i$ is denoted $c^k_i$. Variables of this formulation are $z_{i,k} \in \{0, 1\}$, such that $z_{i,k} = 1$ if route $k \in \mathcal{P}_i$ is chosen. It leads to following program:

\[
\begin{align*}
\min_{z, y} & \quad \sum_{i \in \mathcal{I}, k \in \mathcal{C}_i} c^k_i z_{i,k} + \sum_{j \in \mathcal{J}} P_j y_j \\
\text{s.t.} & \quad \sum_{i, k : j \in k} z_{i,k} + y_j \geq 1 \quad (\pi) \quad \forall j \in \mathcal{J}, \\
& \quad \sum_{k \in \mathcal{C}_i} z_{i,k} \leq 1 \quad (\sigma) \quad \forall i \in \mathcal{I}, \\
& \quad z_{i,k}, y_j \in \{0, 1\}
\end{align*}
\]

LP relaxation computed with Column Generation (CG) algorithm.
CG subproblems

CG subproblems independent for all technicians $i$ : does it exist a route st $CR_i < 0$?

$$CR_i = \min -\sigma_i + \sum_{j,j'} (d(j,j') - \pi_j) u_{j,j'}$$

s.c : \forall j,

$$\sum_{j' \in \mathcal{J} \cup \{d_i\}} u_{j'j} = \sum_{j' \in \mathcal{J} \cup \{d_i\}} u_{jj'}$$

\forall j,

$$\sum_{j' \in \mathcal{J} \cup \{d_i\}} u_{j'j} \leq C^j_i$$

$$\sum_{j' \in I} u_{j'd_i} = \sum_{j' \in I} u_{d_ij'} \leq 1$$

\forall (j, j') \in \mathcal{J},

$$t_j + D_j + T(j, j') \leq t_{j'} + (1 - u_{j,j'}) \cdot M$$

\forall j,

$$t_{\tilde{t}_{i}^{\text{start}}} + T(d_i, j) \leq t_j + (1 - u_{d_i,j}) \cdot M$$

\forall j,

$$t_j + D_j + T(j, d_i) \leq t_{\tilde{t}_{i}^{\text{end}}} + (1 - u_{j,d_i}) \cdot M$$

\forall j, j'\quad u_{j'j} \in \{0, 1\}, t_j \in [\tilde{t}_{j}^{\text{min}}, \tilde{t}_{j}^{\text{max}}]$$

(10)

For one subproblem, resolution is similar to an ERCPSP (Elementary Shortest Patch with Resource Constraints), dynamic programming based approach
Algorithm 1: Standard column generation algorithm

Input: 
\( C \) set of initial columns.

do :
    solve RMP (9) with columns defined in \( C \)
    store dual variables \( \sigma \) and \( \pi \) and optimal cost from (9)
    for each technician \( i \in I \):
        solve (10) to optimality with last \((\sigma, \pi)\) values
        if \( CR_i^* < 0 \) then add the optimal column to \( C \)
    end for
while : columns are added in \( C \)
return the last cost of the RMP (9)
Solving such sub-problems

- Optimality proof: considering ESPPRC or B&B.
- Branch&Bound resolutions allow to generate a pool of negative reduced cost solutions.
- MIP compact matheuristic apply to generate negative reduced cost solutions.
- Aggressive generation of columns with greedy algorithms/matheuristics and local search.
- Note: aggressive column generation require also a procedure to remove columns that are not used in the RMP for several iterations to avoid memory errors.
CG matheuristics

- Compute integer RMP? (usually not a good strategy, here it worked very well, especially fit the highly constrained instances)
- Branch&Price diving heuristics. (here, very good relaxation DW, sometimes optimal, no branching in this case)
- Lagrangian heuristics: compute $v_{i,j} = \sum_{i,k} \mathbb{I}_{j\in k} z_{i,k}$ and repair the continuous assignments tech/job to have a feasible solution.
- RINS tailored heuristic: compute the continuous and the integer RMP. Fix assignments with a common value in continuous and integer RMP and optimize (heuristically).
- Note for Lagrangian, RINS and integral RMP heuristics: can be processed at each iteration of CG algorithm, no need for optimality of RMP.
CG stabilization issues

- General difficulty: an erratic convergence of dual variables
- Bad dual variables generate bad columns, having earlier good dual signals decreases the number of iteration for the CG convergence.
- Stabilization methods: reducing the number of CG iterations providing better dual variables.
- Mainly mathematical techniques to smooth dual variables
- (heuristic) Generation of several columns with a $< 0$ reduced cost may improve the convergence.


⇒ Stabilization has positive impact on exact CG solving and CG matheuristics
Outline

Problem statement and related state-of-the-art

Extended formulation and Column Generation matheuristic

**POPMUSIC** column generation stabilization

Tabu Search Matheuristic intensification

Computational results
Motivations for POPMUSIC column generation

- Dual variables induce to generate columns with jobs with the highest values $\pi_j$ in (10).
- Independent computations of (10) are likely to use the same jobs, with the highest values of $\pi_j$.
- These columns are likely to be redundant in the recombination induced by the next RMP computation.
- Worst case: technicians with identical (or close) characteristics technicians will generate the same column.
- Idea: impose a diversification by considering several technicians $\mathcal{I}_0$ and forbidding to do twice a job among technicians in $\mathcal{I}_0$. 
Conjoint optimization for technicians in $\mathcal{I}_0$

$$RC^*_0 = \max_u \sum_{i \in \mathcal{I}_0} C_i$$

s.t: $\forall j \in \mathcal{J}$

$\forall i \in \mathcal{I}_0,$

$\forall i \in \mathcal{I}_0, \forall j \in \mathcal{J},$

$\forall i \in \mathcal{I}_0,$

$\forall i \in \mathcal{I}_0, \forall (j, j') \in \mathcal{J},$

$\forall i \in \mathcal{I}_0, j \in \mathcal{J},$

$\forall i \in \mathcal{I}_0, j \in \mathcal{J},$

$\forall i \in \mathcal{I}_0,$

$\forall j, j'$

$$\sum_{i \in \mathcal{I}_0, j' \in \mathcal{J}} (u^i_{j,j'} + u^i_{j,d_i}) \leq 1$$

$$-C_i \leq +\sigma_i + \sum_{j,j'} (D(j, j') - \pi_j) u^i_{j,j'}$$

$$\sum_{j' \in \mathcal{J} \cup \{d_i\}} u^i_{j'j} = \sum_{j' \in \mathcal{J} \cup \{d_i\}} u^i_{jj'}$$

$$\sum_{j' \in \mathcal{I}} u^i_{j'd_i} = \sum_{j' \in \mathcal{I}} u^i_{d_i j'} \leq 1$$

$$t_j + D_j + T(j, j') \leq t_{j'} + (1 - u^i_{j,j'}) \cdot M^3_{i,j}$$

$$\tilde{t}^\text{start} + T(i, j) \leq t_j + (1 - u^i_{d_i j}) \cdot M^2_{i,j}$$

$$t_j + D_j + T(j, i) \leq \tilde{t}^\text{end} + (1 - u^i_{j,d_i}) \cdot M^1_{j,j'}$$

$$C_i \geq 0$$

$$u^i_{j,j'} \in \{0, 1\}, t_j \in [\tilde{t}^\text{min}_j, \tilde{t}^\text{max}_j]$$

(11)
Algorithm 2 : POPMUSIC column generation algorithm

**Input :**
- $C$ set of initial columns.

**do :**
- solve RMP with columns defined in $C$
- store dual variables $\sigma$ and $\pi$ and optimal cost the last RMP
- compute $\mathcal{P}_I$ a partition of $\mathcal{I}$ in small subsets
  **for each** subset $\mathcal{I}_0 \in \mathcal{P}_I$ :
  - solve (11) with a matheuristic with last $(\sigma, \pi)$ values
  **for each** column $c$ with a negative reduced cost
  - add the column to $C$
  **end for**
**end for**
**while :** columns are added in $C$
**return** the last cost of the RMP
Solving subproblems of CG schemes

- ESPPRC not valid to solve conjoint optimization with several technicians.
- Matheuristics, frontal
- Decomposition scheme (Algorithm 3): initial solution iterating single technicians computations removing the jobs previously assigned.
- Algorithm 3: computes for all technician their best column as previously and the best complementary columns for them
- VND apply after initial solution. After Algorithm 3, allows to have well balanced solutions
- A question: well balanced reduced costs or the extreme solutions of Algorithm 3?
Decomposition scheme

Algorithm 3 : Diversification of subproblem solutions

Input :
- $I_0$ a subset of technician.
- $s$ a cyclic permutation of $I_0$ with order($s$) = $|I_0|$.
- $\sigma, \pi$ the dual variables of the last RMP computation.

Initialization : $C = \emptyset$, the columns to add in the RMP

for each technician $i \in I_0$ :
    Let $i' = i_0$, $J_0 = J$

for $k = 1$ to $|I_0|$ :
    solve (10) for technician $i'$ with $(\sigma, \pi)$ values and the remaining jobs in $J_0$
    if the solution induces a column with a negative reduced cost
        add the column in $C$
        remove the jobs of the column in $J_0$
        $i' = s(i_0)$
    end for
end for
return $C$
Outline

Problem statement and related state-of-the-art

Extended formulation and Column Generation matheuristic

POPMUSIC column generation stabilization

Tabu Search Matheuristic intensification

Computational results
Motivations for Tabu Search Matheuristic intensification

Generation of several columns with a negative reduced costs only for the 5 first iterations

⇒ Generating several columns per subproblems with a negative reduced cost even with a B&B solving stabilize the CG convergence

⇒ Having a quick procedure to generate several solutions with a negative reduced cost
We denote the binaries \( v_{i,j} = \sum_{j'} u_{i,j'}^{j} \) indicating if technician \( i \in I_0 \) realizes job \( j \). Having \( N \) feasible solutions previously calculated, we denote with \( \tilde{v}_{i,j}^{N} \) the value of these variables. To forbid already generated columns, we add the following “no-good-cuts”:

\[
\forall n \in [1, N], \quad \sum_{i,j : \tilde{v}_{i,j}^{n} = 1} (1 - v_{i,j}^{n}) + \sum_{i,j : \tilde{v}_{i,j}^{n} = 0} v_{i,j}^{n} \geq 1
\]

(12)

To search around the last solution \( \tilde{v}_{i,j}^{N} \), allowing \( k \) modifications from the \( N \)-th solution, it can also be written as a linear constraints:

\[
\sum_{i,j : \tilde{v}_{i,j}^{N} = 1} (1 - v_{i,j}^{N}) + \sum_{i,j : \tilde{v}_{i,j}^{N} = 0} v_{i,j}^{N} \leq k
\]

(13)

Similarly from "pseudo-cuts" from:
Algorithm 4: Tabu search intensification

Input:
- $\mathcal{I}_0$, a subset of technicians
- the current value in the RMP of dual variables $(\sigma, \pi)$
- a set of initial columns $c \in \prod_{i \in \mathcal{I}_0} \mathcal{P}_i$ with a negative reduced cost in (10)
- an integer $N \in \mathbb{N}$, a maximal number of TS iterations
- an integer $k \in \mathbb{N}$, a number of maximal modifications

$\text{TSintensification}(\mathcal{I}_0, k, N)$

1. MIP, a MIP formulation for (10) related to technician $i$
2. $p = c$ initial columns
3. Taboo list of columns $l = \{c\}$
4. an integer $n = 0$ to denote iterations

Do: //Loop to generate columns with negative reduced costs

- Add constraint (13) in MIP with columns of $p$
- Add constraint (12) in MIP with columns of $p$
- solve MIP
- remove constraint (13) in MIP with columns of $p$
- update $p$, the optimal columns in the last MIP
- update $l$ adding the columns of $p$ with a negative reduced cost in $l$

While $n < N$ and ReducedCost($p$) < 0

Return $l$ // the list of column to add in the next RMP
TS with buckets of jobs

- TS iteration requires a computation with $|I_0|$ technicians and $|J|$ jobs.
- Constraints (13) and (12) are helpful for the B&B search.
- However, the B&B can become too difficult for $|J|$ increasing.
- Only subsets of jobs can be considered to have fixed size computations: the jobs of the current solution and the new one can be inserted using different bucket decompositions.
Outline

- Problem statement and related state-of-the-art
- Extended formulation and Column Generation matheuristic
- POPMUSIC column generation stabilization
- Tabu Search Matheuristic intensification

Computational results
Protocol to compare CG convergences

- The CG stabilization can be combined.

- Analyses considering single CG schemes and optimal computations of subproblems to compare the CG schemes.

- Using several matheuristic to solve CG sub-problems have an impact, not detailed here.

- Measures of gaps from the dual variables at an iteration to the optimal dual values: stabilization makes the duals converge faster.

- Computes the convergence of the RMP, measures the number of iterations to converge. (illustrated in the next slides)

- Specific instances built with feasibility graduations (removing/adding technicians, modifying commutation speed, aggregating skills)

- Instances of the literature: Kovacs et al (2012) (work in progress)
**Protocol to compare CG convergences**

To understand the specific contributions of the different CG schemes, we compare several schemes solving exactly subproblems and a proven termination of the CG algorithm:

- **CG1**: it is the classic CG scheme, as written in the Algorithm 1.
- **CG2**: The classical CG scheme is deployed with a TS intensification to solve single technician subproblems. It corresponds to the Algorithm 5 with only partitions into singletons, with parameters $k = 3$ and $N = 5$.
- **CG3**: it is the POPMUSIC CG scheme without TS intensification, where the subproblems (11) are solved only using the Algorithm 3 and exact computations.
- **CG4**: it is the POPMUSIC CG scheme without TS intensification, where the subproblems (11) are solved twice: Algorithm 3 gives first solutions and a second phase optimize the summed reduced cost (and thus balancing the reduced costs) with a VND similar to [?], generating all the columns with a negative reduced cost got from these two phases.
- **CG5**: it corresponds to the strategy CG4 with TS intensification activated with $k = 3$ and $N = 5$. 
⇒ High impact of the matheuristic stabilization for the very first iterations with unstable dual variables
Impact of the matheuristic stabilization for the final iterations
POPMUSIC-CG is interesting for the very first iterations, TS is better for stabilized dual variables.
pure TS stabilization is not significantly improved combining POPMUSIC-CG and TS-GC.
CONCLUSIONS AND PERSPECTIVES
Matheuristics and Column Generation

Several relations and efficient hybridizations:

- CG truncated to guide heuristic search
- Matheuristics to solve CG subproblems, to accelerate the computations CG subproblem.
- Matheuristic within tailored CG schemes to have a more stable CG convergence.

Results:

- Accelerate the computation of primal heuristics relying on CG relaxation.
- Makes easier the computations of proven dual bounds (or allow to compute dual bounds for larger instances than the pure CG scheme).
Specific conclusions for the problem

- Different CG schemes/matheuristics for subproblems that can be combined.
- GC2: Tabu Search alone is very efficient (explain mostly the final convergence)
- GC4: POPMUSIC CG interesting for the first iterations
- GC4 vs GC3: it is better to have balanced reduced costs in the POPMUSIC CG

⇒ Design of the CG scheme to implement more efficiently
Perspectives

- Comparison/combination with exact stabilization techniques

- Instances of Kovacs et al (2012): where are they in our graduation of few to highly-constrained?

- Results of the matheuristics in the instances of Kovacs et al (2012)