Column Generation Techniques for GAP

immediate

February 26, 2019

1 Literature Review

- Branch-and-Price : Column Generation for Solving Huge Integer Programs (Barnhart, et. al., 1998)
- A Branch-and-Price Algorithm for the Generalized Assignment Problem (Savelsbergh and Martin, 1997)
  - Dantzig-Wolfe decomposition for GAP (construct master and sub problems)
  - Column generation : additional columns of the restricted master problem are generated by solving the pricing problem.
  - Branching strategies : variable dichotomy(sing variable) and GUB dichotomy(set of variables)
- Chebyshev center based column generation (Lee and Park, 2011)
  - The column generation procedure based on the simplex algorithm often shows desperately slow convergence. (zig-zag movement)
  - Chebyshev center based column generation techniques
    - Chebyshev center
    - Proximity adjusted Chebyshev center
    - Chebyshev center + Stabilization
    - Proximity adjusted Chebyshev center + Stabilization
  - Computational experiments on the binpacking, VRP, GAP
  - The proposed algorithm could accelerate the column generation procedure.
- Comparison of bundle and classical column generation (O.Briant, et. al., 2006)
  - Bundle method : the dual solution is often constrained to a given interval, and any deviation from the interval is penalized by a penalty function.
  - The penalty function for stabilized column generation : a simple V-shaped function (stabilizing center, ϵ)
- Stabilized Column Generation (O. Du Merle, et. al., 1997)

2 Problems

2.1 A case study : Generalized Assignment Problem

Dantzig-Wolfe Decomposition [1]

$$
\begin{aligned}
\text{(P)}\ \min\ & \sum_{i\in I}\sum_{k\in K} c_k^i x_k^i, \\
\text{s.t.}\ & \sum_{i\in I}\sum_{k\in K_i} \delta_k^j x_k^i \geq 1, \quad j \in J, \\
& -\sum_{k\in K_i} x_k^i \geq -1, \quad \forall i \in I, \\
& x_k^i \geq 0, \quad \forall k \in K_i, i \in I.
\end{aligned}
$$

$$
\begin{aligned}
\text{(D)}\ \max\ & \sum_{j\in J}\pi_j - \sum_{i\in I}\phi_i, \\
\text{s.t.}\ & \sum_{j\in J} \delta_k^j \pi_j - \phi_i \leq c_k^i, \quad \forall k \in K_i, i \in I, \\
& \pi_j \geq 0, \quad \forall j \in J, \\
& \phi_i \geq 0, \quad \forall i \in I.
\end{aligned}
$$

The GAP oracle finds an assignment pattern while satisfying the knapsack constraints :

$$
\max \sum_{j\in J} (\pi_j - c_{ij})\,\delta_j, \quad \text{s.t.} \sum_{j\in J} a_{ij}\delta_j \leq b_i, \delta_j \in \{0,1\}, \quad \forall j \in J
$$

Stabilization

$$
\begin{aligned}
(\tilde{P})\ \min\ & \sum_{i\in I}\sum_{k\in K} c_k^i x_k^i + \sum_{j\in J}\delta_j(\gamma_j^+ - \gamma_j^-) + \sum_{i\in I}\phi_i(y_i^+ - y_i^-), \\
\text{s.t.}\ & \sum_{i\in I}\sum_{k\in K_i} \delta_k^j x_k^i + \gamma_j^+ - \gamma_j^- \geq 1, \quad \forall j \in J, \\
& -\sum_{k\in K_i} x_k^i + y_i^+ - y_i^- \geq -1, \quad \forall i \in I, \\
& \gamma_j^+ \leq \epsilon,\ \gamma_j^- \leq \epsilon, \quad \forall j \in J, \\
& y_i^+ \leq \epsilon,\ y_i^- \leq \epsilon, \quad \forall i \in I, \\
& x_k^i \geq 0, \quad \forall k \in K_i, i \in I, \\
& y_i^+ \geq 0, \quad \forall k \in K_i, i \in I.
\end{aligned}
$$

[1] The written mathematical formulation are from (Lee and Park, 2011)

3 New Column Generation Approach

Consider $J = J_1 \cup J_2$ and the dual solutions π_j for all $j \in J_2$ are fixed to $\Pi' = \bigcup_{j \in J_2} \pi'_j$ (π' is a feasible solution). Then, the reformulation of the dual problem and its primal are as follows :

$$
\begin{aligned}
\text{(D)}\ \ \max\ & \sum_{j \in J} \pi_j - \sum_{i \in I} \phi_i, \\
\text{s.t.}\ & \sum_{j \in J} \delta_k^j \pi_j - \phi_i \le c_k^i, \quad \forall k \in K_i, i \in I, \\
& \pi_j \le \pi'_j \quad \forall j \in J_2 \\
& \pi_j \ge 0, \quad \forall j \in J, \\
& \phi_i \ge 0, \quad \forall i \in I.
\end{aligned}
$$

$$
\begin{aligned}
\text{(P)}\ \ \min\ & \sum_{i \in I} \sum_{k \in K} c_k^i x_k^i + \sum_{j \in J_2} \pi'_j y_j, \\
\text{s.t.}\ & \sum_{i \in I} \sum_{k \in K_i} \delta_k^j x_k^i \ge 1, \quad j \in J_1, \\
& \sum_{i \in I} \sum_{k \in K_i} \delta_k^j x_k^i + y_j \ge 1, \quad j \in J_2, \\
& -\sum_{k \in K_i} x_k^i \ge -1, \quad \forall i \in I, \\
& x_k^i \ge 0, \quad \forall k \in K_i, i \in I, \\
& y_j \ge 0, \quad \forall j \in J.
\end{aligned}
$$

Stabilization

$$
\begin{aligned}
(\tilde{P})\ \ \min\ & \sum_{i \in I} \sum_{k \in K} c_k^i x_k^i + \sum_{j \in J_2} \pi'_j y_j + \sum_{j \in J} \delta_j (\gamma_j^+ - \gamma_j^-) + \sum_{i \in I} \phi_i (y_i^+ - y_i^-), \\
\text{s.t.}\ & \sum_{i \in I} \sum_{k \in K_i} \delta_k^j x_k^i + \gamma_j^+ - \gamma_j^- \ge 1, \quad j \in J_1, \\
& \sum_{i \in I} \sum_{k \in K_i} \delta_k^j x_k^i + y_j + \gamma_j^+ - \gamma_j^- \ge 1, \quad j \in J_2, \\
& -\sum_{k \in K_i} x_k^i + y_i^+ - y_i^- \ge -1, \quad \forall i \in I, \\
& \gamma_j^+ \le \epsilon,\ \gamma_j^- \le \epsilon, \quad \forall j \in J, \\
& y_i^+ \le \epsilon,\ y_i^- \le \epsilon, \quad \forall i \in I, \\
& x_k^i \ge 0, \quad \forall k \in K_i, i \in I, \\
& y_j \ge 0, \quad \forall j \in J.
\end{aligned}
$$

Algorithm 1 Column generation for GAP

tolerance $\leftarrow$ 0.000001
$\Pi_2 \leftarrow \mathbf{0}$
iteration $\leftarrow$ 0
criteria $\leftarrow$ True
while any(criteria) **do**
    iteration += 1
    if iteration % 2 == 0 **then**
        Solve M1 with fixed Π_1
        $\Pi_2 \leftarrow$ optimal dual solution
        Solve S
        if Reduced cost $<$ tolerance **then**
            Add the column to M1 ad M2
        else
            criteria $\leftarrow$ False
        end if

Π_1 is fixed.

    else
        Solve M2 with fixed Π_2
        $\Pi_1 \leftarrow$ optimal dual solution
        Solve S
        if Reduced cost $<$ tolerance **then**
            Add the column to M1 ad M2
        else
            criteria $\leftarrow$ False
        end if

Π_2 is fixed.

    end if
end while

4 Preliminary Tests

Github page : `https://github.com/mody3062/CG`

Testing algorithms

- Classical column generation (Kelly's cutting plane)
- Stabilized column generation (O. Du Merle, et. al., 1997)
- Separation + Classical column generation
- Separation + Stabilized column generation

Algorithmic parameters RMP was constructed with a single decision variable which is dummy. The coefficient of the dummy variable on the objective function was set to a sufficiently large value, which is the sum of listed values such that `np.sum(c,axis=1)`. For stabilized column generation algorithm, I changed the parameter ϵ from 0.01 to 0.0001 for every 100 trials. (I am not sure whether I could understand the criteria for changing the parameter value(ϵ) well.)

	Kelly			Stab.			Sep.			Sep.+Stab.		
	iteration	total(s)	M(%)	iteration	total(s)	M(%)	iteration	total(s)	M(%)	iteration	total(s)	M(%)
d05100	2485	26.13	25%	2216	33.27	32%	2296	44.57	61%	2321	49.8	66%
d10100	1223	7.83	25%	1068	11.49	42%	949	6.38	32%	858	7.25	40%
d10200	4485	133.04	54%	4423	199.37	61%	3253	132.49	67%	3703	193.21	73%
d20100	852	3.67	26%	782	6.63	49%	673	3.24	29%	629	4.15	41%
d20200	2513	38.59	41%	2549	65.75	55%	1862	32.86	51%	2288	52.38	59%
e05100	2577	15.03	42%	2356	27.73	51%	3487	74.37	86%	2761	37.61	76%
e10100	1345	6.83	36%	1405	12.45	52%	1160	6.64	54%	1261	8.93	58%
e10200	6273	153.91	77%	6580	271.2	80%	4328	186.97	87%	4455	218.22	88%
e20100	942	3.13	32%	979	7.4	52%	711	2.81	36%	644	3.5	46%
e20200	3273	37.77	57%	3374	76.73	69%	2390	44.17	75%	2713	61.78	78%

$$
\begin{aligned}
\text{(P)}\ \ \min\ \ & x^T Q x + c^T x \\
\text{s.t.}\ \ & Ax \geq b, \\
& x \in \mathbb{R}^n_+
\end{aligned}
$$

$$
\begin{aligned}
\text{(P)} \quad \min\ & \boldsymbol{d}^T \Omega \boldsymbol{d} - \lambda \boldsymbol{d}^T \boldsymbol{\alpha} && (1) \\
\text{s.t.}\ & d_i = w_i - w_i^{bench}, \quad \forall i \in N, && (2) \\
& w_i \geq 0, \quad \forall i \in N, && (3) \\
& \sum_{i \in N} w_i = 1, && (4) \\
& \underline{\mathrm{A}}_i \leq d_i \leq \overline{\mathrm{A}}_i, \quad \forall i \in N, && (5) \\
& \underline{\mathrm{SE}}_s \leq \sum_{i \in N_s} d_i \leq \overline{\mathrm{SE}}_s, \quad \forall s \in S, && (6) \\
& \underline{\mathrm{MC}}_k \leq \sum_{i \in N_k} d_i \leq \overline{\mathrm{MC}}_k, \quad \forall k \in K, && (7) \\
& \underline{\mathrm{B}} \leq \sum_{i \in N} \beta_i d_i \leq \overline{\mathrm{B}}, && (8) \\
& \underline{\mathrm{K}} \leq card(w_i \neq 0) \leq \overline{\mathrm{K}}, && (9) \\
& \underline{\mathrm{AS}} \leq 1 - \sum_{i \in N} \min\{w_i, w_i^{bench}\} \leq \overline{\mathrm{AS}}, && (10) \\
& \underline{\mathrm{TE}} \leq \sqrt{\boldsymbol{d}^T \Omega \boldsymbol{d}} \leq \overline{\mathrm{TE}}, && (11)
\end{aligned}
$$

Proposition 1. *Constraint* (9) *can be replaced by the following set of constraints:*

$$
\begin{aligned}
& y_i \geq w_i - 0.001, \quad \forall i \in N, \\
& y_i \leq w_i + 0.999, \quad \forall i \in N, \\
& \underline{K} \leq \sum_{i \in N} y_i \leq \overline{K}, \\
& y_i \in \{0, 1\}, \quad \forall i \in N.
\end{aligned}
$$

Proposition 2. *Constraint* (10) *can be replaced by the following set of constraints:*

$$
\begin{aligned}
& \underline{AS} \leq 1 - \sum_{i \in N} v_i \leq \overline{AS}, \\
& w_i - z_i \leq v_i \leq w_i, \quad \forall i \in N, \\
& w_i^{bench} - 1 + z_i \leq v_i \leq w_i^{bench}, \quad \forall i \in N, \\
& z_i \in \{0, 1\}, \quad \forall i \in N.
\end{aligned}
$$