Chebyshev center based column generation

Chungmok Lee and Sungsoo Park

Speaker. Yerin Kim

Systems Optimization Laboratory
Korea Advanced Institute of Science and Technology

January 16, 2019

Outline

1. Introduction
   - Standard Column Generation
   - Column Generation Techniques
   - Motivation
2. Methodology
   - Chebyshev center
   - Chebyshev center based column generation
   - Proximity adjusted Chebyshev center
   - Stabilized Chebyshev center algorithm
3. Computational experiments
   - Computational environment
   - Case1 : Binpacking problem
   - Case2 : vehicle routing problem with time windows (VRPTW)
   - Case3 : generalized assignment problem (GAP)
4. Conclusion

Introduction

Standard Column Generation

- *Danzig-Wolfe Decomposition*
  - An optimization problem (original form) :

$$\begin{aligned} \min\ & c^T x \\ \text{s.t. } & Ax \geq b \\ & x \in conv(X), \end{aligned}$$

  - Dantzig-Wolfe formulation in linear form (restricted version : $X^k \subseteq X$)

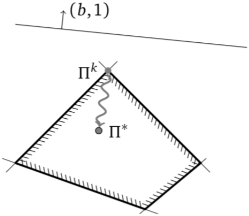
[Restricted Master Problem (RMP)]

$$\begin{aligned} \min\ & \sum_{i \in K} (c^T x^i)\lambda_i, \\ \text{s.t. } & \sum_{i \in K} (Ax^i)\lambda_i \geq b, \\ & \sum_{i \in K} \lambda_i = 1, \\ & \lambda_i \geq 0, \quad \forall i \in K. \end{aligned}$$

[Dual of RMP]

$$\begin{aligned} \max\ & f_D(\pi, \pi_0) = b^T\pi + \pi_0 \\ \text{s.t. } & \sum_{i \in I} (Ax^i)^T \pi + \pi_0 \leq c^T x^i, \quad \forall_{i \in K}, \\ & \pi \geq 0. \end{aligned}$$

- *Danzig-Wolfe Decomposition* (Cont.)
  - In the dual space, dual solutions are enhanced toward optimal!



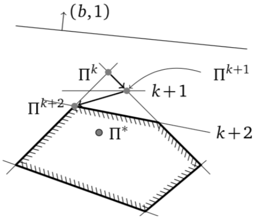
(a) Dual space at the kth iteration.

  - $f_D(\tilde{\pi}, \tilde{\pi}_0) \leq f_D(\pi^*, \pi_0^*) \leq f_D(\pi^k, \pi_0^k)$ (Large is better)

    $\tilde{\Pi} = (\tilde{\pi}, \tilde{\pi}_0)$: a feasible dual solution to the unrelaxed dual problem
    $\Pi^* = (\pi^*, \pi_0^*)$: the optimal solution of the unrelaxed dual problem
    $\Pi^k = (\pi^k, \pi_0^k)$: the dual optimal solution of the dual problem

Introduction
Standard Column Generation

- *Kelley's cutting plane method* (Classical column generation)
  - solve separation problem (oracle) to find a maximally violated inequality at the current solution.
  - In the dual space, zigzag movement is appeared.



(b) Classical column generation (Kelley's method).

- Simplex based column-generation is a successful method for tackling large-size mathematical programming problems.
- However, there are some major drawbacks (F.Vanderbeck,2005):
  1. slow convergence (the *tailing-off effect*);
  2. poor columns in the initial stage (the *head-in effect*);
  3. the optimal value of the restricted master problem remains the same during many iterations (the *plateau effect*);
  4. the dual solution jumps from one extreme point to another (the *bang–bang effect*);
  5. the intermediate Lagrangian dual bounds do not converge monotonically (the *yo-yo effect*).
- Recently, many acceleration schemes for column generation have been proposed.

- *Stabilized column generation* [1]
  - Bundle method : the dual solution is often constrained to a given interval, and any deviation from the interval is penalized by a penalty function.
  - Motivation :

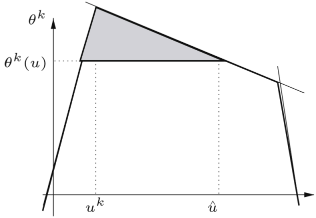


Figure 1: The safeguard polyhedron P for $k = 3$. $\theta^k(u) = \theta(\hat{u})$

- Standard column generation chooses u^k as the highest point in P
- Stabilized column generation : u^k is replaced by some less high but more central point in P

[1] O. Briant, et. al., Comparison of bundle and classical column generation, Mathematical Programming Ser.A 299-344, 113, 2008

- *Stabilized column generation* (Cont.)
  - The penalty function for stabilized column generation : a simple V-shaped function
  - Stabilizing center
  - Slope ϵ determines how much the distance from the best dual solution is penalized.

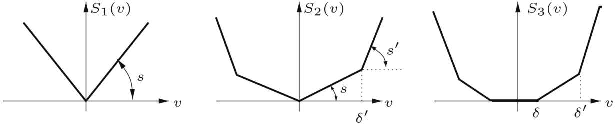


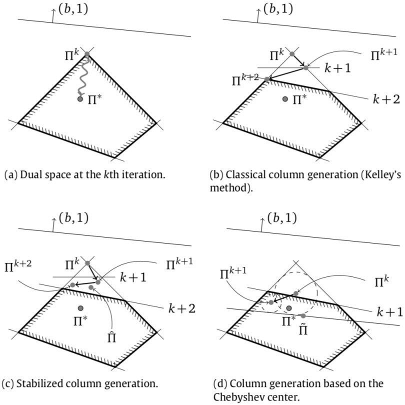
Figure 2: Stabilization by a penalty

  - The performance depends largely on how to manage parameters (stabilizing center, ϵ)

Introduction

Motivation

- ☐ How to improve the convergence?
- ☐ From the literature reviews,



(a) Dual space at the kth iteration.
(b) Classical column generation (Kelley's method).
(c) Stabilized column generation.
(d) Column generation based on the Chebyshev center.

Figure 3: Π^* is the ture optimal solution. Π^k is the optimal solution of the current relaxed dual problem.

Methodology

Chebyshev center

- ☐ *Chebyshev center* : the deepest point inside the set (the farthest from the exterior)
  - Aussumption : bounded, closed, nonempty convex set
  - For a convex set of $a_i^T x \leq b_i, \forall i \in \{1, \ldots, m\}$, the mathematical formulation is

$$\begin{aligned} &\max\ r \\ &\text{s.t. } a_i^T x + \|a_i\|_* r \leq b_i, \forall i \in \{1, \ldots, m\}, \\ &\qquad r \geq 0. \end{aligned}$$

($\|a_i\|_*$ is any norm of vector a.)

- ☐ Accelerated Chebyshev center based cutting plane method(Betro,2004) : concept of centering
- ☐ ACCPM + bundle method(S.Elhedhli and T.G.Moore, 2004) gives good convergence.

Methodology

Chebyshev center based column generation

- Recall the dual of the restricted master problem.

$$\begin{aligned} \max\ & f_D(\pi, \pi_0) = b^T\pi + \pi_0 \\ \text{s.t. } & \sum_{i \in I} (Ax^i)^T \pi + \pi_0 \le c^T x^i, \quad \forall_{i \in K}, \\ & \pi \ge 0. \end{aligned}$$

- *Chebyshev dual master problem* :

$$\begin{aligned} \max\ & r \\ \text{s.t. } & \left(Ax^i\right)^T \pi + \pi_0 + \left\| \left(Ax^i, 1\right) \right\|_* r \le c^T x^i, \quad \forall i \in K, \\ & -\pi_j + r \le 0, \quad \forall j = 1, \dots, m, \\ & \boxed{-b^T\pi - \pi_0 + \|(b,1)\|_* r \le \tilde{Z},} \\ & \pi \ge 0, \\ & r \ge 0. \end{aligned}$$

Methodology

Chebyshev center based column generation

- *Chebyshev primal master problem* :

$$
\begin{aligned}
\max \ & \sum_{i \in K} c^T x^i \lambda_i - \tilde{Z} z \\
\text{s.t. } & \sum_{i \in K} A x^i \lambda_i - y - b^T z \geq 0, \\
& \sum_{i \in K} \lambda_i - z = 0, \\
& \sum_{i \in K} \left\| \left(A x^i, 1 \right) \right\|_* \lambda_i + \sum_{j=1}^{m} y_j + \|(b, 1)\|_* z \geq 1, \\
& \lambda_i \geq 0, \quad \forall i \in K \\
& y_j \geq 0, \quad \forall j = 1, \ldots, m, \\
& z \geq 0.
\end{aligned}
$$

$\left(\|(a)\| \text{ is } \sqrt{\sum_{i \in N} |a_i|^2} \text{ : Euclidean norm} \right)$

- Pricing subproblem : $\min \left(c^T - \pi^T A \right) x - \pi_0'$, s.t. $x \in X$

$(\because z - \left\| \left(A x^i, 1 \right) \right\|_* r \leq z')$

Methodology

Chebyshev center based column generation

- The detailed algoirthm of the column generation based on the Chebyshev center

Algorithm 1 Column generation based on the Chebyshev center

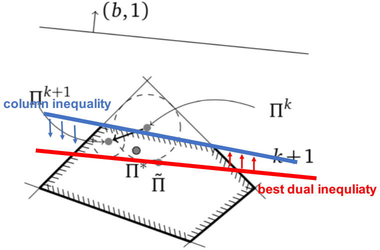
1: **procedure** CHEBYSHEVCENTERCOLGEN
2: $\quad \tilde{Z} \leftarrow 0$ ▷ Or any valid dual bound
3: $\quad$ **repeat**
4: $\quad\quad$ Solve the restricted Chebyshev primal master problem
5: $\quad\quad$ (λ', y', z') and $(\pi', \pi_0') \leftarrow$ the optimal primal and dual solutions
6: $\quad\quad$ **if** the optimal value $\sum_{i \in K} c^T x^i \lambda_i' - \tilde{Z} z'$ is greater than $\varepsilon z'$ **then** ▷ Equivalently $r > \varepsilon z'$
7: $\quad\quad\quad$ Solve the column generation subproblem using (π', π_0') as dual solution
8: $\quad\quad\quad$ **if** new column is identified with x' **then** ▷ Reduced cost is negative
9: $\quad\quad\quad\quad$ Add new column $(Ax', 1, \|(Ax', 1)\|)$ to the problem
10: $\quad\quad\quad$ **else**
11: $\quad\quad\quad\quad$ $\tilde{Z} \leftarrow b^T \pi' + \pi_0'$ ▷ Update the best dual bound
12: $\quad\quad\quad$ **end if**
13: $\quad\quad$ **end if**
14: $\quad$ **until** $r \leq \varepsilon z'$
15: **end procedure**

- *line 8-9* : $z' < 0$ ($(\pi', \pi_0')$is infeasible to daul) $\rightarrow$ add the new column.
- *line 10-11* : $z' \geq 0$ ($(\pi', \pi_0')$is feasible to dual) $\rightarrow$ update the best dual bound $\tilde{Z}$.
- *line 14* : set termination criterion to εz instead of 0.

Methodology

Proximity adjusted Chebyshev center

- The column inequalities attempt to *push* the next Chebyshev center down, while the best dual inequality tries to *lift* it up.

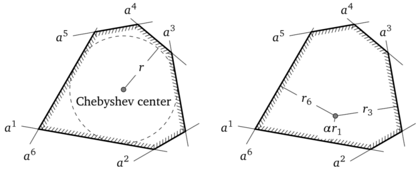


- By introducing the proximity parameter α, the equation is obtained :

$$\sum_{i \in K} \left\| \left(Ax^i, 1 \right) \right\|_* \lambda_i + \sum_{j=1}^{m} y_j + \boxed{\alpha \|(b,1)\|_* z} \geq 1$$

Methodology

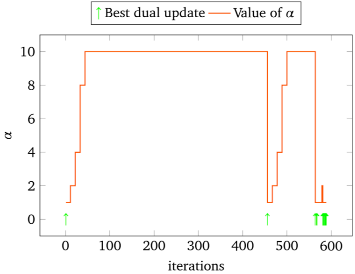
Proximity adjusted Chebyshev center



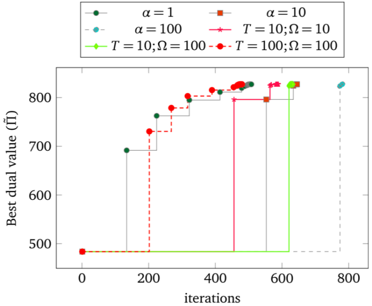
- α determines how far the next Chebyshev center point is located from the dual bound inequality.
  - small $\alpha \rightarrow$ update dual bound quickly
  - large $\alpha \rightarrow$ more relevant new columns ($\because$ Chebyshev center $\approx$ true optimal solution(extreme point))
- Two more parameters : $T \in \mathbb{R}^+$ and $\Omega \in \mathbb{R}^+$
  - α is increased gradually up to Ω when the best dual value is not updated during the last T iterations.

Methodology

Proximity adjusted Chebyshev center



(a) Change of α for the proximity adjusted Chebyshev algorithm with $T = 10$ and $\Omega = 10$.



(b) Updates of dual bound for the different algorithmic parameters.

- Test problem : vehicle routing problem (VRP) C101
- Small α may lead to frequent updates of the dual bound.

Methodology

Stabilized Chebyshev center algorithm

- Stabilized Chebyshev primal master problem

$$
\begin{aligned}
\min\ & \sum_{i\in K} c^T x^i \lambda_i - \tilde{Z} z + \sum_{j=1,\dots,m} \tilde{\Pi}_j \left(\delta_j^+ - \delta_j^-\right) + \tilde{\Pi}_0 \left(\delta_0^+ - \delta_0^-\right) \\
\text{s.t.}\ & \sum_{i\in K} A x^i \lambda_i - y - b^T z + \delta^+ - \delta^- \geq 0, \\
& \sum_{i\in K} \lambda_i - z + \delta_0^+ - \delta_0^- = 0, \\
& \sum_{i\in K} \left\| \left(A x^i, 1\right) \right\|_* \lambda_i + \sum_{j=1}^{m} y_j + \|(b,1)\|_* z \geq 1, \\
& \lambda_i \geq 0, \quad \forall i \in K, \\
& y_j \geq 0, \quad \forall j = 1, \dots, m, \\
& z \geq 0, \\
& \delta_j^+ \leq \epsilon, \quad \delta_j^- \leq \epsilon, \quad \forall j = 1, \dots, m, \\
& \delta_0^+ \leq \epsilon, \quad \delta_0^- \leq \epsilon.
\end{aligned}
$$

  - reduce to Chebyshev primal master problem iff $\epsilon = 0$.

Computational experiments

Computational environment

- Test problems

Case 1. binpacking problem
Case 2. vehicle routing problem with time windows (VRPTW)
Case 3. generalized assignment problem (GAP)

- Algorithms & Parameters
  - `Chebyshev` with $\alpha = 1$
  - `PA Chebyshev` with $T = 10$ and $\Omega = 100$
  - `Stabilization`
  - `Kelley`
  - `Chebyshev+Sta.`
  - `PA Chebyshev+Sta.` with $T = 10$ and $\Omega = 100$

  * The penalty coefficient ε was initially set to 0.1, and then sequentially updated to 0.01, 0.001, 0.0001, and 0.

- Experiment Setting
  - AMD X2 2.9 GHz PC with 4 GB RAM
  - Optimization solver : CPLEX 10.1

Computational experiments

Case1 : Binpacking problem

- LP relaxation of the standard covering type Dantzig–Wolfe decomposition

[Binpacking Primal]

$$
\begin{aligned}
\min \ & \sum_{p \in P} x_p \\
\text{s.t. } & \sum_{p \in P} a_{ip} x_p \geq 1, \quad \forall i = 1, \ldots, I \\
& x_p \geq 0, \quad \forall p \in P.
\end{aligned}
$$

[Binpacking Dual]

$$
\begin{aligned}
\max \ & \sum_{i=1}^{I} \pi_i \\
\text{s.t. } & \sum_{i=1}^{I} a_{ip} \pi_i \leq 1, \quad \forall p \in P, \\
& \pi_i \geq 0, \quad \forall i = 1, \ldots, I.
\end{aligned}
$$

[Binpacking Chebyshev Primal]

$$
\begin{aligned}
\min \ & \sum_{p \in P'} x_p - \sum_{i=1}^{I} \tilde{\pi}_i z \\
\text{s.t. } & \sum_{p \in P'} a_{ip} x_p - y_i - z \geq 0, \quad \forall i = 1, \ldots, I \\
& \sum_{p \in P'} \|a_p\| \, x_p + \sum_{i=1} y_i + \alpha \|\vec{1}\| z \geq 1 \\
& x_p \geq 0, \quad \forall p \in P', \quad y_i \geq 0, \quad \forall i = 1, \ldots, I, \quad z \geq 0.
\end{aligned}
$$

Computational experiments

Case1 : Binpacking problem

☐ Computational Results

Table 1
Binpacking problem.

Prob	Chebyshev		PA Chebyshev		Chebyshev+Sta.		PA Chebyshev+Sta.		Stabilization		Kelley	
	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)
u120	373.1	0.4 (35.9%)	360.5	0.4 (32.4%)	**201.8**	0.3 (34.4%)	244.4	0.3 (35.7%)	329.8	0.4 (33.3%)	403.2	0.3 (31.0%)
u250	760.6	2.2 (31.5%)	727.1	2.4 (27.6%)	**398.8**	1.8 (31.8%)	576.6	1.9 (28.6%)	669.9	2.2 (33.9%)	834.6	1.7 (30.2%)
u500	1437.6	21.1 (52.0%)	1388.3	20.0 (51.9%)	**797.1**	14.8 (66.6%)	1154.1	17.0 (52.8%)	1222.5	15.7 (42.5%)	1584.0	10.0 (18.0%)
u1000	2792	1274.7 (92.6%)	2721	1271.2 (92.8%)	**1614**	824.4 (96.9%)	2303	873.7 (91.8%)	2346	175.0 (59.1%)	3073	82.6 (9.6%)
t60	268.6	0.3 (66.7%)	250.3	0.3 (66.7%)	100.9	0.1 (44.4%)	**94.1**	0.1 (42.9%)	99.8	0.1 (50.0%)	213.3	0.1 (50.0%)
t120	483.8	3.2 (87.9%)	445.9	3.2 (89.1%)	**177.5**	0.4 (52.3%)	200.7	0.5 (68.8%)	225.2	0.7 (58.0%)	405.0	0.5 (51.9%)
t249	819.8	15.2 (86.9%)	741.7	14.9 (85.6%)	**370.6**	2.3 (48.9%)	494.3	3.5 (63.4%)	487.5	11.2 (84.8%)	809.8	3.2 (52.4%)
t501	1564.9	20.4 (30.1%)	1392.0	16.6 (31.0%)	**757.6**	14.6 (66.8%)	964.3	17.9 (61.7%)	1010.8	393.6 (98.0%)	1594.0	18.0 (39.6%)

- `Chebyshev+Sta.` gives the best performance in terms of the iteration number.
- However, `Kelley` is the fastest among tested algorithms.
- In my opinion, it takes pretty long time for each iteration. (Chebyshev master problem is relatively hard to solve as it is.)

☐ Computational Results (Cont.)

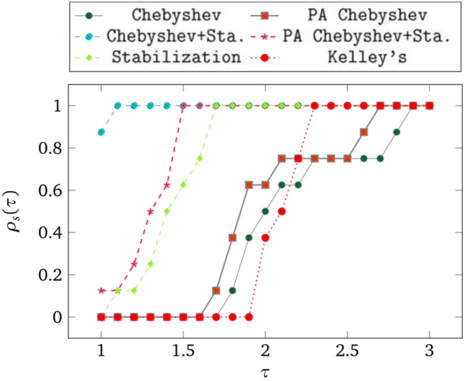


Fig. 4. Performance profile graph for the binpacking problems.

(Large is better.)

- *performance ratio*

  : $r_{p,s} = \frac{t_{p,s}}{\min_{s \in S}\{t_{p,s}\}}, \forall p \in P$

  (S : a set of algorithms , P : a set of problems)

- *performance measure*

  : $\rho_s(\tau) = \frac{|\{p \in P | r_{p,s} \leq \tau\}|}{|P|}$

  ($\rho_s(1)$ means the probability that the algorithm s will not be outperformed by the rest of the algorithms.)

Computational experiments

Case2 : vehicle routing problem with time windows (VRPTW)

☐ Computational Results

Table 2
Vehicle routing problem.

Prob	Chebyshev		PA Chebyshev		Chebyshev+Sta.		PA Chebyshev+Sta.		Stabilization		Kelley	
	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)
C101	509	12.3 (96.2%)	634	21.5 (96.0%)	**137**	2.7 (97.0%)	762	23.3 (96.5%)	273	8.5 (97.7%)	825	35.0 (97.9%)
C102	935	256.6 (99.7%)	1219	459.1 (99.3%)	**220**	62.6 (99.8%)	902	323.6 (99.3%)	526	201.6 (99.7%)	1084	447.8 (99.7%)
C105	685	25.8 (97.9%)	814	46.8 (97.1%)	**140**	4.8 (98.3%)	1174	73.0 (96.6%)	315	17.5 (98.5%)	1653	107.8 (97.8%)
C106	578	74.5 (99.4%)	814	126.2 (99.0%)	**176**	9.7 (98.9%)	856	148.3 (98.9%)	443	81.4 (99.6%)	1062	199.9 (99.5%)
C107	799	41.7 (98.4%)	1210	99.2 (97.1%)	**173**	6.7 (98.5%)	1147	97.2 (97.6%)	376	36.3 (99.1%)	1036	142.1 (99.2%)
C108	599	351.7 (99.9%)	703	584.4 (99.8%)	**325**	165.2 (99.9%)	695	497.3 (99.8%)	536	390.4 (99.9%)	669	678.4 (99.9%)
C109	563	1070.4 (100.0%)	661	2252.4 (99.9%)	**374**	553.5 (99.9%)	625	1641.8 (99.9%)	547	1578.1 (100.0%)	666	2113.5 (100.0%)
R101	428	4.3 (93.0%)	486	5.9 (88.3%)	**284**	2.9 (93.1%)	451	5.3 (88.9%)	401	4.7 (93.0%)	466	5.3 (94.0%)
R102	578	24.6 (98.0%)	553	30.3 (97.1%)	**369**	16.8 (98.4%)	509	26.5 (97.1%)	489	27.4 (98.3%)	573	29.8 (98.6%)
R103	1075	174.1 (99.2%)	594	97.1 (98.7%)	**454**	61.9 (99.2%)	657	107.3 (98.3%)	543	99.5 (99.3%)	617	115.4 (99.5%)
R105	553	19.2 (97.0%)	573	27.4 (94.1%)	**381**	13.8 (97.1%)	580	27.8 (94.2%)	492	21.3 (97.0%)	592	27.0 (97.7%)
R106	1215	138.0 (98.7%)	618	101.8 (98.2%)	**409**	57.5 (99.2%)	623	101.4 (98.2%)	566	104.0 (99.3%)	642	124.5 (99.4%)
R107	1060	361.2 (99.6%)	668	351.2 (99.2%)	**397**	152.3 (99.7%)	647	309.5 (99.2%)	568	337.2 (99.8%)	693	370.1 (99.8%)
R109	514	57.3 (99.1%)	536	97.7 (98.8%)	**360**	45.1 (99.1%)	546	101.1 (98.6%)	487	84.4 (99.3%)	580	110.1 (99.5%)
R110	562	188.4 (99.7%)	597	311.2 (99.5%)	**412**	149.3 (99.7%)	572	272.0 (99.4%)	546	291.3 (99.8%)	592	325.1 (99.8%)
R111	739	194.2 (99.6%)	632	242.2 (99.2%)	**383**	88.8 (99.5%)	653	242.0 (99.1%)	548	231.8 (99.7%)	660	274.4 (99.7%)
RC101	457	10.9 (96.6%)	503	14.3 (94.0%)	**365**	9.8 (96.6%)	499	14.0 (93.6%)	448	12.6 (96.4%)	495	13.6 (97.1%)
RC102	598	62.7 (99.1%)	564	84.6 (98.7%)	**394**	58.5 (99.4%)	557	91.5 (98.7%)	492	92.7 (99.4%)	571	102.7 (99.5%)
RC103	677	274.3 (99.7%)	618	343.6 (99.5%)	**410**	268.9 (99.8%)	583	319.6 (99.6%)	510	339.1 (99.8%)	580	354.1 (99.8%)
RC105	519	30.8 (98.6%)	519	38.5 (97.8%)	**354**	24.0 (98.6%)	529	40.3 (97.5%)	454	38.2 (98.8%)	547	41.0 (98.9%)
RC106	470	64.6 (99.3%)	520	97.5 (99.0%)	**377**	58.8 (99.3%)	523	98.3 (99.1%)	475	89.7 (99.5%)	538	98.0 (99.5%)
RC107	523	382.3 (99.9%)	537	485.6 (99.8%)	**373**	271.8 (99.9%)	535	558.8 (99.8%)	478	498.1 (99.9%)	553	532.1 (99.9%)

- `Chebyshev+Sta.` gives the best performance in terms of the iteration number and time.

Computational experiments

Case2 : vehicle routing problem with time windows (VRPTW)

☐ Computational Results (Cont.)

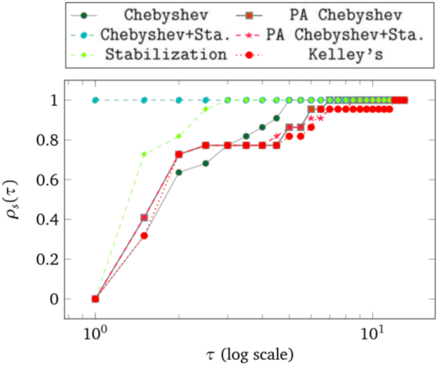


Fig. 5. Performance profile graph for the VRP problems.

Computational experiments

Case3 : generalized assignment problem (GAP)

- Computational Results

Table 3
Generalized assignment problem.

Prob	Chebyshev		PA Chebyshev		Chebyshev+Sta.		PA Chebyshev+Sta.		Stabilization		Kelley's	
	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)
d05100	985	34.6 (57.9%)	861	38.4 (56.3%)	729	23.9 (54.7%)	791	34.7 (60.6%)	**712**	16.1 (58.9%)	825	26.4 (58.8%)
d10100	233	3.7 (50.9%)	**203**	3.6 (53.3%)	216	2.4 (39.1%)	229	4.2 (46.0%)	302	4.3 (49.1%)	267	3.8 (50.3%)
d10200	1104	350.6 (49.9%)	1022	518.9 (37.7%)	**683**	67.5 (33.5%)	1013	483.5 (23.3%)	916	130.5 (46.2%)	1114	225.1 (45.8%)
d20100	126	1.4 (44.6%)	**124**	1.5 (43.5%)	154	1.6 (39.9%)	136	2.0 (47.3%)	176	2.0 (57.4%)	162	1.8 (54.9%)
d20200	328	34.7 (39.6%)	**263**	35.5 (43.0%)	396	38.5 (24.8%)	282	35.6 (38.9%)	445	44.2 (33.1%)	411	42.9 (33.6%)
e05100	709	19.8 (55.3%)	675	26.8 (49.8%)	692	15.3 (46.5%)	681	22.8 (57.3%)	**617**	12.7 (59.0%)	710	21.2 (57.1%)
e10100	276	4.1 (49.9%)	**236**	5.2 (41.8%)	347	4.6 (44.5%)	252	4.9 (49.0%)	319	4.5 (48.5%)	271	3.5 (48.1%)
e10200	1338	469.7 (39.0%)	1173	734.0 (28.3%)	1260	407.4 (30.2%)	1149	671.8 (24.9%)	**950**	132.8 (55.0%)	1267	247.9 (47.3%)
e20100	151	1.6 (52.1%)	**126**	1.8 (52.0%)	227	2.7 (43.5%)	151	2.6 (51.9%)	189	2.2 (53.9%)	168	1.7 (56.1%)
e20200	377	68.1 (61.1%)	**292**	66.6 (63.3%)	507	65.7 (39.1%)	299	51.9 (48.4%)	495	45.8 (35.6%)	439	37.7 (32.5%)

- `PA Chebyshev` outperforms the other five algorithms. (The performance gap is not that apparent.)

Computational experiments

Case3 : generalized assignment problem (GAP)

☐ Computational Results (Cont.)

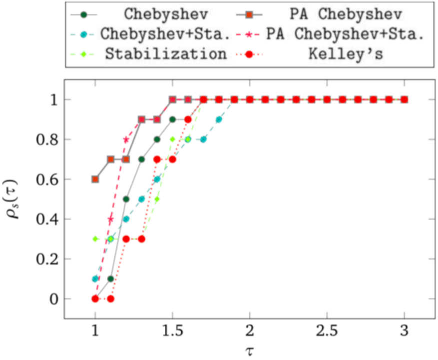


Fig. 6. Performance profile graph for the GAP problems.

Conclusion

- The column generation procedure based on the simplex algorithm often shows desperately slow convergence. (zig-zag movement)
- Chebyshev center based column generation
  - 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.

Discussion

- Chebyshev doesn't show good performance on the binpacking and GAP.
  - Both have low percentage of solving the sub-problems.
  - The degree of degeneracy on the master problem seems to be big.
- The proposed algorithms also largely depend on the algorithmic parameters.
- Github page : https://github.com/mody3062/CCG