School of Mathematics

Interior Point Methods and Column Generation

Jacek Gondzio
Email: J.Gondzio@ed.ac.uk
URL: http://www.maths.ed.ac.uk/~gondzio

Outline

- **Part 1: IPMs for Optimization**
  - IPM tricks: log barrier, central path
  - polynomial complexity
  - optimal partition
- **Part 2: Warmstarting IPMs**
- **Part 3: Column Generation with IPM**
  - cutting stock problem
  - vehicle routing problem with time windows
- **Conclusions**

Part 1:

Interior Point Methods for Optimization

Interior Point Methods

- re-born in 1984
- Narendra Karmarkar, AT&T Bell Labs

Shocking mathematical concept:

Take **linear** optimization problem
and add **nonlinear** function to the objective.

A step against common sense and centuries of mathematical practice:

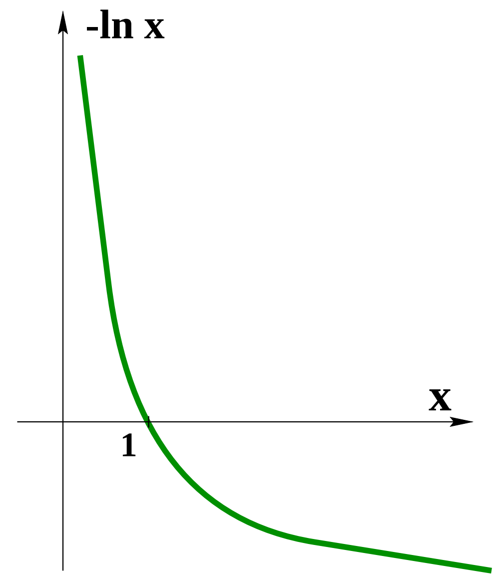
“nonlinearize” linear problem

Logarithmic barrier

$$-\ln x_j$$

"replaces" the inequality

$$x_j \geq 0 \ .$$



Observe that

$$\min \ e^{-\sum_{j=1}^{n} \ln x_j} \quad \Longleftrightarrow \quad \max \prod_{j=1}^{n} x_j$$

The minimization of $-\sum_{j=1}^{n} \ln x_j$ is equivalent to the maximization of the product of distances from all hyperplanes defining the positive orthant: it prevents all x_j from approaching zero.

LP Problem: $\min \; c^T x \;\; \text{s.t.} \;\; Ax = b, \;\; x \geq 0.$

LP Barrier Prob: $\min \; c^T x - \mu \sum_{j=1}^{n} \ln x_j \;\; \text{s.t.} \;\; Ax = b.$

Lagrangian: $$L(x, y, \mu) = c^T x - y^T (Ax - b) - \mu \sum_{j=1}^{n} \ln x_j,$$

Stationarity: $$\nabla_x L(x, y, \mu) = c - A^T y - \mu X^{-1} e = 0$$
$$\nabla_y L(x, y, \mu) = Ax - b = 0.$$

Denote: $s = \mu X^{-1} e, \quad \text{i.e.} \quad XSe = \mu e.$

Complementarity in the Interior Point Method

The first order optimality conditions (FOC)

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= \mu e, \\ x, s &\geq 0, \end{aligned}$$

where $X = diag\{x_j\}$, $S = diag\{s_j\}$ and $e = (1, \cdots, 1) \in \mathcal{R}^n$.

Analytic centre (μ-centre): a (unique) point $(x(\mu), y(\mu), s(\mu))$, $x(\mu) > 0$, $s(\mu) > 0$ that satisfies FOC.

The interior point method gradually reduces the complementarity products

$$x_j \cdot s_j \approx \mu \rightarrow 0 \quad \forall j = 1, 2, ..., n.$$

Interior Point Methods

Theory: convergence in $\mathcal{O}(\sqrt{n})$ or $\mathcal{O}(n)$ iterations
Practice: convergence in $\mathcal{O}(\log n)$ iterations

Expected number of IPM iterations:

Problem Dimension	LP	QP
1,000	5 - 10	5 - 10
10,000	10 - 15	10 - 15
100,000	15 - 20	10 - 15
1,000,000	20 - 25	15 - 20
10,000,000	25 - 30	15 - 20
100,000,000	30 - 35	20 - 25
1000,000,000	35 - 40	20 - 25

... but one iteration may be expensive!

Complementarity $x_j \cdot s_j = 0 \quad \forall j = 1, 2, ..., n.$

Simplex Method guesses an optimal partition:

For *basic* variables, $s_B = 0$ and

$$(x_B)_j \cdot (s_B)_j = 0 \quad \forall j \in \mathcal{B}.$$

For *non-basic* variables, $x_N = 0$ hence

$$(x_N)_j \cdot (s_N)_j = 0 \quad \forall j \in \mathcal{N}.$$

Interior Point Method uses ε-mathematics:

Replace $\quad x_j \cdot s_j = 0 \quad \forall j = 1, 2, ..., n$

by $\quad x_j \cdot s_j = \mu \quad \forall j = 1, 2, ..., n.$

Force convergence $\mu \to 0$.

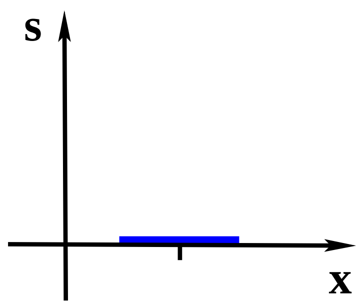
First Order Optimality Conditions

Simplex Method:

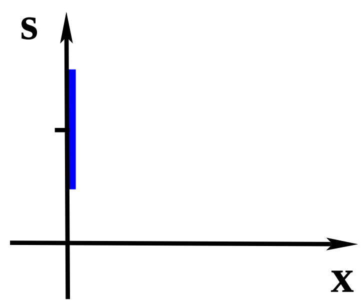
$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ XSe &= 0 \\ x,\, s &\geq 0. \end{aligned}$$

Interior Point Method:

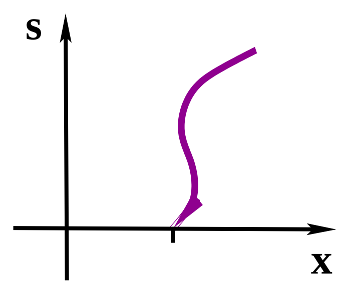
$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ XSe &= \mu e \\ x,\, s &\geq 0. \end{aligned}$$



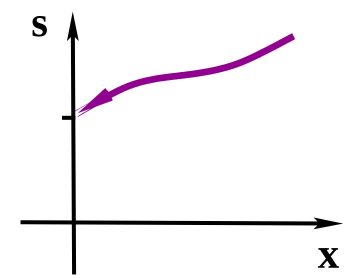
Basic: x > 0, s = 0



Nonbasic: x = 0, s > 0



"Basic": x > 0, s = 0



"Nonbasic": x = 0, s > 0

G, IPMs 25 years later, *EJOR* 218 (2012), 587–601.

Part 2:

Warmstarting IPMs

A need to solve a sequence of similar problems

- **column generation**
- **cutting plane methods**
- subproblems in the block-angular LPs (Dantzig-Wolfe decomp., Benders decomp.)
- **B&B, (and B&Cut, B&Cut&Price, etc)**
- SQP
- any sequence of similar problems
  example: computing efficient frontier in Markowitz portfolio optimization

Warm Starts Which method should be used?

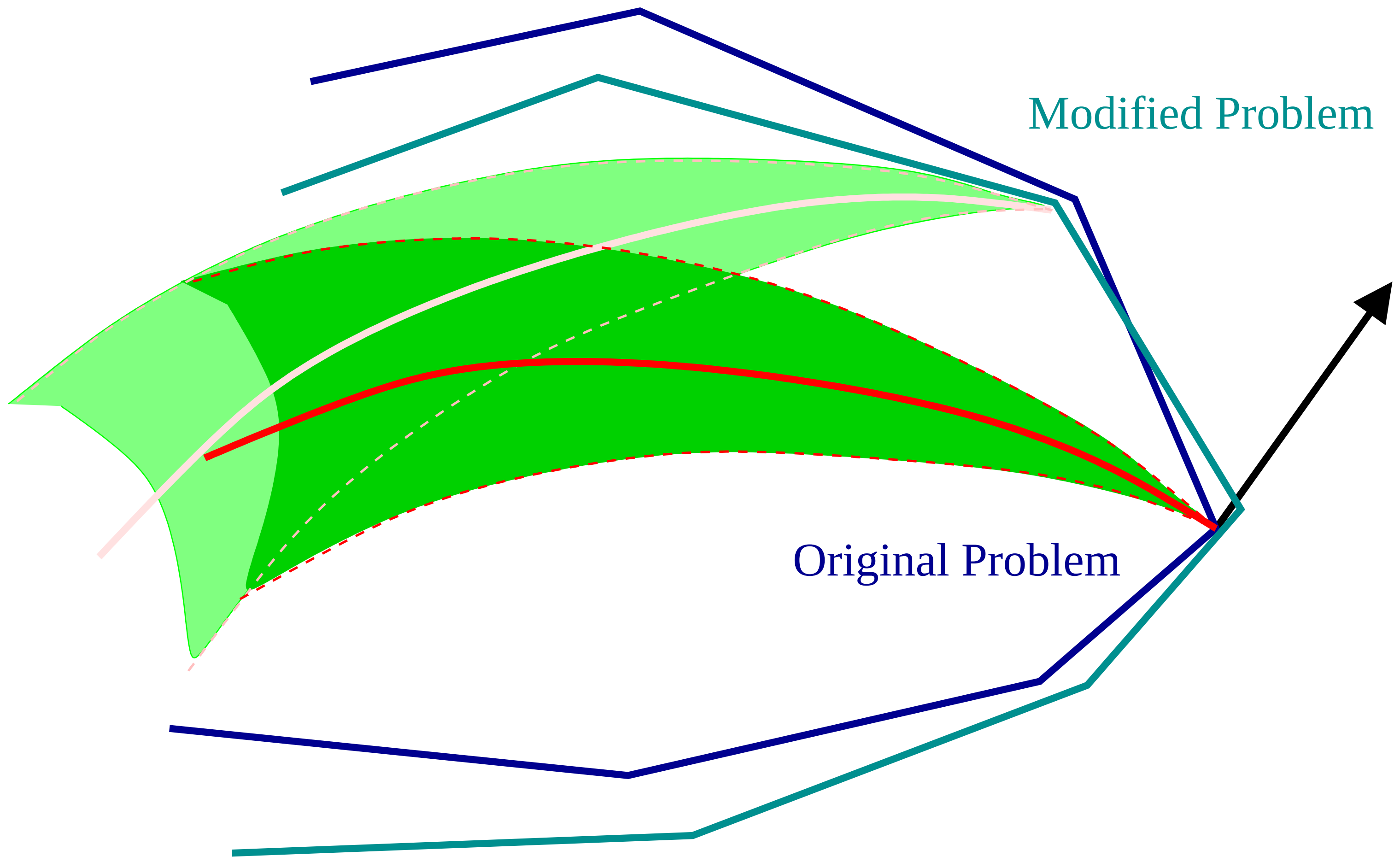
- Simplex Method, or
- Interior Point Method.

When is the Simplex Method better?
→ **few** indices change optimal partition
B & B, adding *one* cut in CPM, etc.

When is the Interior Point Method better?
→ **many** indices change optimal partition
adding *many* cuts in CPM,
dealing with a general change of problem data, etc

Conjecture:
The more changes in the (large) problem
the more attractive IPM-based warm starts are.

Difficulty of IPM Warm Starts



Warm Starting in 1990+

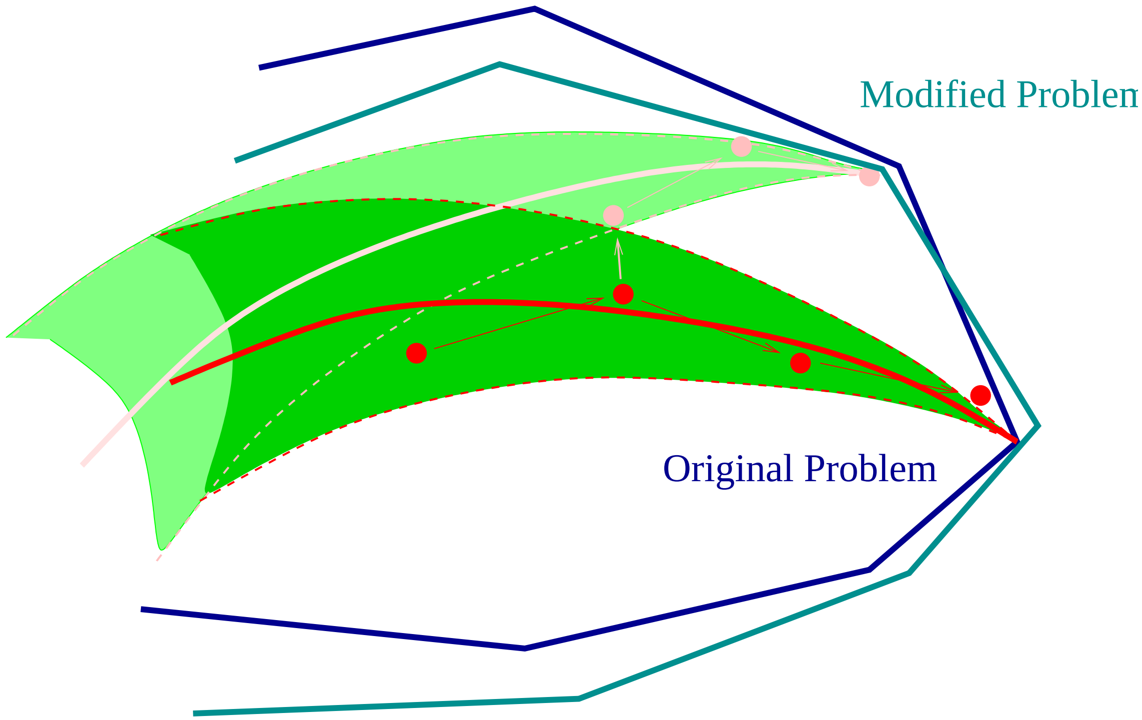
Mitchell, PhD Thesis, Cornell Univ. 1988
Goffin & Vial et al., development of ACCPM 1990+
G. & Sarkissian, development of PDCGM in 1995
G., *Math. Prog.* 83 (1998) 125–143
G. & Vial, *COAP* 14 (1999) 17–36

ACCPM Analytic Centre Cutting Plane Method

PDCGM Primal-Dual Column Generation Method

Warmstarting Heuristic

Idea: Start close to the (new) central path, not close to the (old) solution



G., *Mathematical Programming* 83 (1998) 125–143

Warm Start with μ-centres

Old Problem:

$$\begin{array}{ll} \min & c_0^T x + \frac{1}{2} x^T Q_0 \, x \\ \text{s.t.} & A_0 x = b_0, \\ & x \geq 0, \end{array}$$

New Problem:

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q \, x \\ \text{s.t.} & A x = b, \\ & x \geq 0, \end{array}$$

We assume:
$c \approx c_0, \;\; Q \approx Q_0, \;\; A \approx A_0, \;\; b \approx b_0.$

Warm Starting in 2000+

Yildirim & Wright, *SIOPT* 12 (2002) 782–810
G. & Grothey, *SIOPT* 13 (2003) 842–864
Fliege, *Maths of OR* 31 (2006) 825–845
Benson & Shanno, *COAP* 38 (2007) 371–399
Benson & Shanno, *COAP* 40 (2008) 143–189
G. & Grothey, *SIOPT* 19 (2008) 1184–1210
John & Yildirim, *COAP* 41 (2008) 151–183
Colombo, G. & Grothey, *MP* 127 (2011) 371-397
Colombo & Grothey, follow-up reports in 09,10
Engau, Anjos & Vannelli, *SIOPT* 20 (2010) 1828
Benson & Mahanta, report in 2009
Ordonez & Waltz, report in 2009

IPM Warmstarts: Theoretical Results

Yildirim & Wright, *SIOPT* 12 (2002) 782–810
G. & Grothey, *SIOPT* 13 (2003) 842–864

Lemma. Let $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$ for problem (LP) then the full Newton step $(\Delta x, \Delta y, \Delta s)$ in the perturbed problem ($\tilde{\text{LP}}$) is feasible and

$$(x + \Delta x, y + \Delta y, s + \Delta s) \in \tilde{\mathcal{N}}_{-\infty}(\gamma)$$

provided that

$$\delta_{bc} = \|\xi_c\|_2 + \|A^T(AA^T)^{-1}\xi_b\|_2 \leq \|P\|_\infty \frac{\gamma_0}{1 + 1/\gamma}\mu,$$

where

$$P = I - S^{-1}A^T(AXS^{-1}A^T)^{-1}AX, \;\; \xi_b = \tilde{b} - Ax, \;\; \xi_c = \tilde{c} - A^T y - s.$$

LOQO vs OOPS warmstarting NETLIB problems

Benson & Shanno, *COAP* 38 (2007) 371–399

G. & Grothey, *SIOPT* 19 (2008) 1184–1210
→ *Unblocking technique ...*

Average savings:

- LOQO (B&S, 2007) 20–30 %
- OOPS (G&G, 2008) 50–70 %

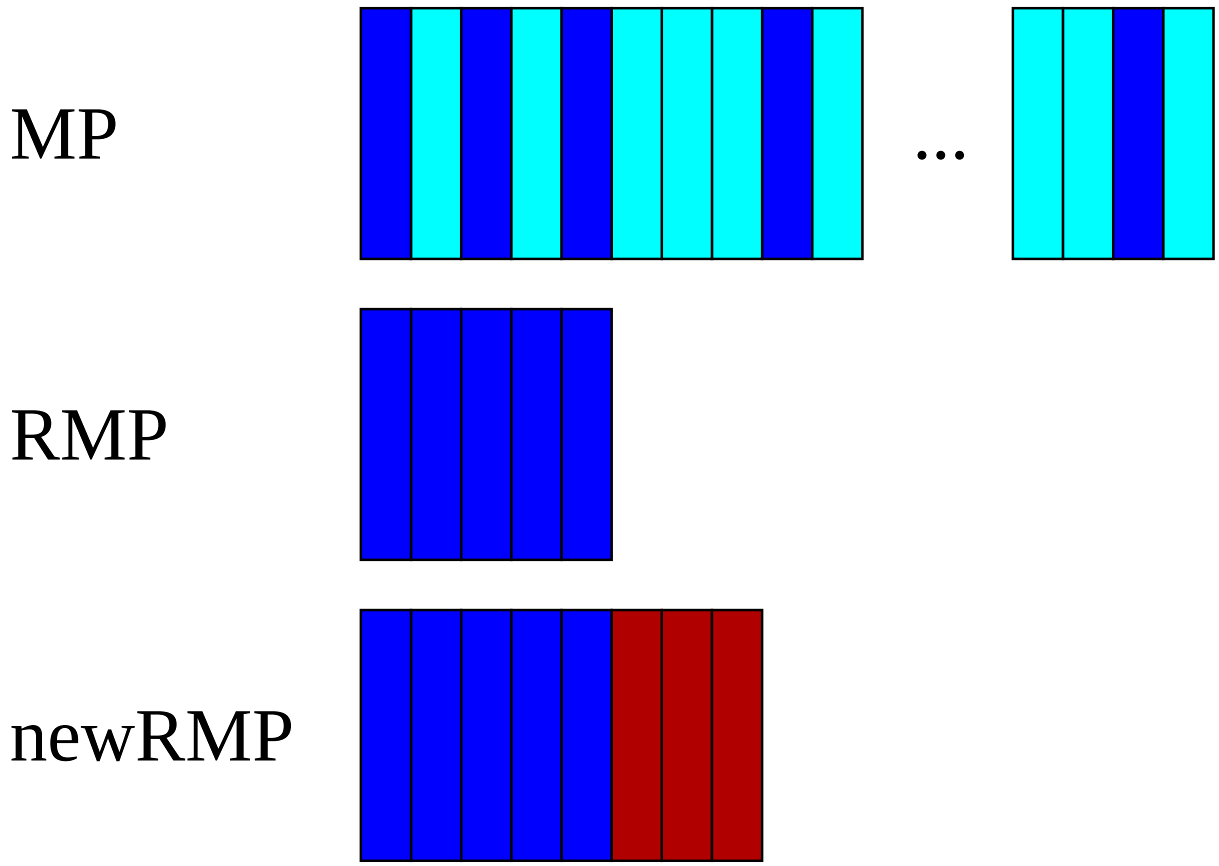
Part 3:

Primal-Dual Column Generation Method

Joint work with two PhD students:

Pablo Gonzalez-Brevis and Pedro Munari

Column Generation (CG)



Column Generation (CG)

Consider an LP, called the master problem (MP):

$$\begin{aligned} z^\star := \min \quad & \sum_{j \in N} c_j \lambda_j, \\ \text{s.t.} \quad & \sum_{j \in N} a_j \lambda_j = b, \\ & \lambda_j \geq 0, \qquad \forall j \in N. \end{aligned}$$

- N is too big;
- The columns a_j are implicit elements of $\mathcal{A}$;
- We know how to generate them!

CG: Restricted master problem (RMP): $\overline{N} \subset N$

$$
\begin{aligned}
z_{RMP} := \min \quad & \sum_{j \in \overline{N}} c_j \lambda_j, \\
\text{s.t.} \quad & \sum_{j \in \overline{N}} a_j \lambda_j = b, \\
& \lambda_j \geq 0, \qquad \forall j \in \overline{N}.
\end{aligned}
$$

- Optimal $\bar{\lambda}$ for the RMP $\Rightarrow$ feasible $\hat{\lambda}$ for the MP;
- $\hat{\lambda}_j = \bar{\lambda}_j$, $\forall j \in \overline{N}$, and $\hat{\lambda}_j = 0$ otherwise;
- Hence, $z^\star \leq z_{RMP} = UB$ (Upper Bound).
- How to know it is optimal?
  - Call the oracle!

CG:

- Oracle: check the feasibility of the dual $\overline{u}$;
- Reduced costs: $s_j = c_j - \overline{u}^T a_j, \; \forall j \in N$;
- But the columns are not explicit and, hence,

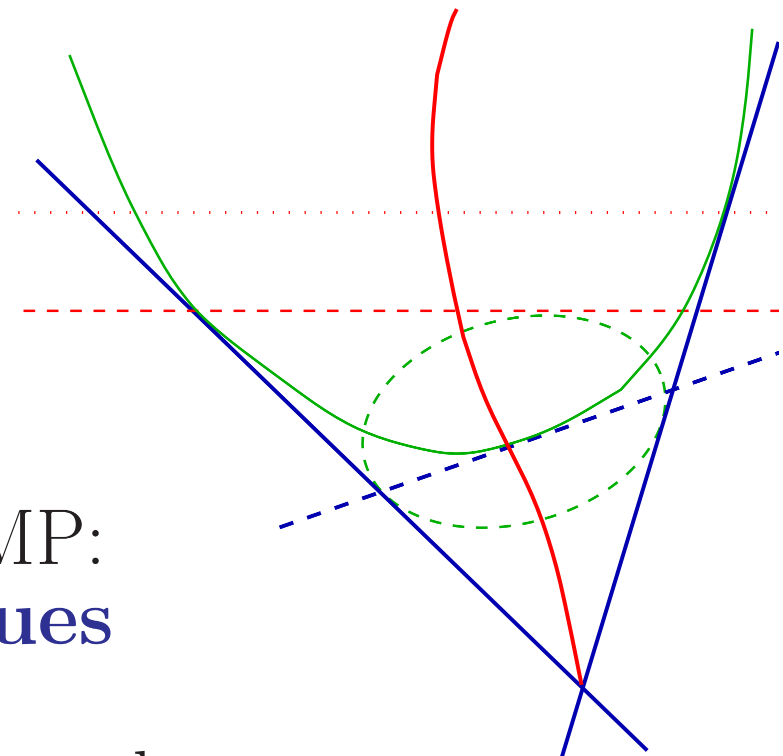
$$z_{SP} := \min\{c_j - \overline{u}^T a_j | a_j \in \mathcal{A}\}.$$

- (we reset $z_{SP} := 0$, if $z_{SP} > 0$);
- Lower Bound: $LB = z_{RMP} + \kappa z_{SP} \leq z^\star$, where

$$\kappa \geq \sum_{i \in N} \lambda_i^\star,$$

- If $z_{SP} < 0$, then new columns are generated;
- Otherwise, an optimal solution of the MP was found!

Appealing features of IPMs



- Use IPM to solve the RMP:
  $\rightarrow$ **no degeneracy issues**

- Terminate RMP solution early:
  $\rightarrow$ **get stable dual solution** $\bar{u}$

PDCGM Algorithm

Parameters: $\varepsilon_{\max}$, D, δ, κ

1. set $\text{LB} = -\infty$, $\text{UB} = \infty$, $\text{gap} = \infty$, $\varepsilon = 0.5$;
2. while $(\text{gap} > \delta)$ do
3. find a well-centred ε-optimal $(\tilde{\lambda}, \tilde{u})$ of the RMP;
4. $\text{UB} = \tilde{z}_{RMP}$;
5. call the oracle with the query point $\tilde{u}$;
6. $\text{LB} = \kappa \tilde{z}_{SP} + b^T \tilde{u}$;
7. $\text{gap} = (\text{UB} - \text{LB})/(1 + |\text{UB}|)$;
8. $\varepsilon = \min\{\varepsilon_{\max},\ \text{gap}/D\}$;
9. if $(\tilde{z}_{SP} < 0)$ then add new columns into the RMP;
10. end(while)

CSP: Column Generation Formulation

Gilmore and Gomory (1961) formulation:

$$\begin{aligned} \min \quad & \sum_{p \in P} \lambda_p, \\ \text{s.t.} \quad & \sum_{p \in P} a_p \lambda_p \geq d, \\ & \lambda_p \geq 0 \text{ and integer}, \quad \forall p \in P. \end{aligned}$$

- Columns are cutting patterns;
- We do not need to enumerate all of them;
- They can be dynamically generated *knapsack problem.*

VRPTW: Column Generation Formulation

Desrochers et al. (1992):

$$\begin{aligned} \min \quad & \sum_{p \in P} c_p \lambda_p \\ \text{s.t.} \quad & \sum_{p \in P} a_p \lambda_p = \mathbf{1}, \\ & \lambda_p \in \{0, 1\}, \qquad \forall p \in P. \end{aligned}$$

- Columns are possible vehicle paths;
- The columns can be dynamically generated *shortest path problem with resource constraints.*

Computational experiments

Solving LP relaxations

Comparison of:

- Standard column generation (**SCG**):
  - simplex-type methods of IBM/CPLEX v.12.1.
- Primal dual column generation (**PDCGM**):
  - interior point solver HOPDM.
- Analytic centre cutting plane (**ACCPM**):
  - open-source solver OBOE/COIN-OR.

Cutting stock problem

		SCG		PDCGM		ACCPM	
Cuts	Class	iters	time	iters	time	iters	time
10	Small	150	**1.2**	102	2.1	253	26.1
	Large	251	77.0	158	**18.3**	368	148.7
50	Small	71	**2.1**	63	3.8	277	106.3
	Large	134	58.2	97	**23.1**	400	277.6
100	Small	54	**4.2**	54	7.3	308	221.8
	Large	101	67.8	82	**31.5**	449	525.2

262 instances:

178 small ($m \leq 199$), 84 large ($m \geq 200$)

`http://www.tu-dresden.de/~capad/`

CSP: Larger Instances: BPP-U09??? family

		SCG		PDCGM		ACCPM	
Instance	m	iters	time	iters	time	iters	time
U09498	1005	548	12947	293	**5678**	762	21254
U09513	975	518	9904	267	**4277**	779	19362
U09528	945	541	9173	276	**4924**	740	15920
U09543	915	506	7798	263	**3724**	723	13449
U09558	885	482	5585	265	**2730**	683	10861
U09573	855	473	4771	230	**2054**	672	9794
U09588	825	467	4950	247	**1649**	658	9376
U09603	795	465	3962	237	**1668**	627	7504

Vehicle routing problem with time windows

		SCG		PDCGM		ACCPM	
Cuts	Class	iters	time	iters	time	iters	time
10	Small	26	0.3	22	**0.2**	94	0.5
	Medium	67	6.2	38	**2.6**	122	5.7
	Large	188	114.1	73	**41.6**	171	92.1
100	Small	12	0.2	17	0.2	92	0.6
	Medium	26	3.0	23	**1.7**	120	5.8
	Large	65	42.4	38	**21.5**	166	87.5

87 instances:

29 small ($n = 25$), 29 med ($n = 50$), 29 large ($n = 100$)

`http://www2.imm.dtu.dk/~jla/solomon.html`

VRPTW: Larger Instances

		SCG		PDCGM		ACCPM	
Instance	n	iters	time	iters	time	iters	time
R_1_2_1	200	57	43	45	**34**	423	202
C_1_2_1	200	85	41	29	**15**	169	82
RC_1_2_1	200	67	110	57	**88**	385	607
R_1_4_1	400	131	865	84	**641**	636	3076
C_1_4_1	400	137	552	53	**186**	272	909
RC_1_4_1	400	189	2789	113	**1436**	521	6649
R_1_6_1	600	222	7558	118	**4260**	897	25870
C_1_6_1	600	183	2335	48	**510**	482	5173
RC_1_6_1	600	258	18972	150	**8844**	923	56683

Integer optimization

Integer VRPTW solved to optimality

Branch-Price-and-Cut, Pedro Munari's PhD

	DLH08			I-PDCGM		
Problem	cuts	nodes	time	cuts	nodes	time
C103	0	1	28	0	1	**8**
C104	0	1	86	0	1	**17**
RC103	262	5	541	162	5	**429**
RC104	437	21	11773	251	7	**3436**
R103	53	1	20	15	1	**9**
R104	391	11	3103	216	7	**949**

DLH08: **Desaulniers, Lessard & Hadjar**, *Transportation Science* 42 (2008) 387-404.

Solomon, *Operations Research* 35 (1987) 254–265.
Homberger&Gehring, *EJOR* 162 (2005) 220-238.

Conclusions

A completely new perspective is needed to exploit the insight offered by IPMs in a number of **combinatorial optimization** applications:

- column generation
- cutting plane methods
- B & B, (and B & Cut, B & Cut & Price, etc)

Warmstarting works well in the CG context:
problems are re-optimized in **3-5** IPM iterations

References

- **G., Gonzalez-Brevis, Munari**, *New developments in the primal-dual column generation technique*, ERGO Tech Rep, Edinburgh, 2011.
- **Munari, G.**, *Using the primal-dual interior point algorithm within the branch-price-and-cut method*, ERGO Tech Rep, Edinburgh, 2012.
- **G., Gonzalez-Brevis**, *A new warmstarting strategy for the primal-dual column generation method*, ERGO Tech Rep, Edinburgh, 2012.

Example: Cutting Stock Problem (CSP)

A set N of large pieces of wood of length W is given.
We need to cut them into smaller pieces.
We need d_j units of small piece $j \in M$ of length w_j.

Minimize the number of units of large pieces of wood.

Define binary variable y_i which takes value 1 if i-th large piece of wood is cut and 0 if it is not used.

Define integer variable x_{ij} which determines the number of units of small piece of wood $j \in M$ obtained by cutting the large piece $i \in N$.

Cutting Stock Problem (CSP)

Kantorovich's formulation:

$$
\begin{array}{lrl}
\min & \sum_{i \in N} y_i & \\
\text{s.t.} & \sum_{i \in N} x_{ij} \geq d_j & \forall j \in M, \\
 & \sum_{j \in M} x_{ij} w_j \leq W y_i & \forall i \in N, \\
 & y_i \in \{0, 1\} & \forall i \in N, \\
 & x_{ij} \geq 0 \text{ and integer} & \forall j \in M, \forall i \in N.
\end{array}
$$

LP relaxation gives very weak bound.

Vehicle Routing Problem with Time Windows

A company delivers goods to customers $i \in C$.
The company has vehicles $k \in V$ and each of them starts at a depot, travels to several customers and returns to the depot. The visit of vehicle k to customer i needs to take place in a specific time window: $a_i \leq s_{ik} \leq b_i$, where s_{ik} is the time when vehicle k reaches customer i.

Objective: Minimize the total cost of delivery.

Define binary variable x_{ijk} which takes value 1 if vehicle k travels from customer i to customer j $(k \in V,\ i, j \in C)$ and takes value zero otherwise.

Vehicle Routing Problem with Time Windows

Constraints:

Exactly one vehicle leaves customer i:

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in C$$

Vehicle capacity constraint:

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \le q, \quad \forall k \in V$$

Each vehicle leaves the depot and returns to it:

$$\sum_{j \in N} x_{0jk} = 1 \quad \text{and} \quad \sum_{j \in N} x_{i(n+1)k} = 1, \quad \forall k \in V$$

VRPTW: Constraints (continued)

Time-window constraint

$$s_{ik} + t_{ij} - M(1 - x_{ijk}) \leq s_{jk}, \quad \forall i, j \in N, \forall k \in V.$$

Since x_{ijk} is binary the above constraint has the following meaning: If $x_{ijk} = 1$ (vehicle k travels from customer i to customer j) then

$$s_{ik} + t_{ij} \leq s_{jk}$$

that is, the arrival time of vehicle k to customer j is greater than or equal the sum of time when vehicle k arrives to customer i and the time t_{ij} it takes to travel from i to j.

Otherwise (if $x_{ijk} = 0$) the presence of “big” M guarantees that the constraint is always inactive.

VRPTW

$$
\begin{array}{llr}
\min & \sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk} & \\
\text{s.t.} & \sum_{k\in V}\sum_{j\in N} x_{ijk} = 1, & \forall i \in C, \\
& \sum_{i\in C} d_i \sum_{j\in N} x_{ijk} \le q, & \forall k \in V, \\
& \sum_{j\in N} x_{0jk} = 1, \ \sum_{i\in N} x_{i(n+1)k} = 1, & \forall k \in V, \\
& \sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{jhk} = 0, & \forall h \in C, \forall k \in V, \\
& s_{ik} + t_{ij} - M(1 - x_{ijk}) \le s_{jk}, & \forall i,j \in N, \forall k \in V, \\
& a_i \le s_{ik} \le b_i, & \forall i \in N, \forall k \in V, \\
& x_{ijk} \in \{0,1\}, & \forall i,j \in N, \forall k \in V.
\end{array}
$$