Using Interior-Point Methods within Mixed-Integer Nonlinear Programming

.

Hande Y. Benson

Drexel University

Motivation: Discrete Variables

- Handling discrete variables generally requires a bilevel approach:
  - Upper level: Branch-and-bound, branch-and-cut, outer approximation
  - Lower level: Active-set methods, interior-point methods
- Active-set methods are considered superior at the lower level because
  - They can be warmstarted
  - They can identify infeasible problems
  - They can handle fixed variables naturally
- A textbook interior-point method cannot do any of these.
- But current interior-point implementations outperform active-set implementations on many large problems
- As sizes of mixed-integer nonlinear programming problems grow, there will be a definite need for interior-point methods at the lower level.

Interior-Point Methods

Each NLP relaxation has the form:

$$\begin{array}{ll} \min\limits_{x,y} & f(x,y) \\ \text{s.t.} & h(x,y) \geq 0 \\ & l \leq y \leq u \end{array}$$

Add slack variables:

$$\begin{array}{lllll} \min\limits_{x,y,g,t} & f(x,y) & & & \\ \text{s.t.} & h(x,y) & - & w & = & 0 \\ & y & - & g & = & l \\ & y & + & t & = & u \\ & w, & g, & t & \geq & 0. \end{array}$$

Interior-Point Methods - 2

First-order conditions for the log barrier problem are

$$\begin{aligned} h(x,y) - w &= 0 \\ y - g &= l \\ y + t &= u \end{aligned} \qquad \begin{aligned} \nabla_x f(x,y) - A_x^T \lambda &= 0 \\ \nabla_y f(x,y) - A_y^T \lambda - z + s &= 0 \end{aligned}$$

$$\begin{aligned} W\Lambda e &= \mu e \\ GZe &= \mu e \\ TSe &= \mu e \end{aligned}$$

Use Newton's Method to solve this system. At each iteration, solve the reduced KKT system:

$$\begin{bmatrix} -H_{xx} & -H_{xy} & A_x^T \\ -H_{xy} & -(H_{yy} + D) & A_y^T \\ A_x & A_y & E \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \lambda \end{pmatrix} =$$

$$\begin{pmatrix} \nabla_x f(x,y) - A_x^T \lambda \\ \nabla_y f(x,y) - A_y^T \lambda - z + s - D_g(l - y) - D_t(u - y) - \mu G^{-1} e + \mu T^{-1} e \\ \mu \Lambda^{-1} e - h(x,y) \end{pmatrix}$$

where

$$E = W\Lambda^{-1}, \quad D = D_g + D_t, \quad D_g = G^{-1} Z, \quad D_t = T^{-1} S.$$

Interior-Point Methods - 3

At each iteration:

- choose steplengths to ensure that slacks remain strictly positive and sufficient progress toward optimality and feasibility is attained.
- value of the barrier parameter may also be updated as a function of $(W^{(k+1)}\Lambda^{(k+1)}e, G^{(k+1)}Z^{(k+1)}e, T^{(k+1)}S^{(k+1)}e)$.

Stopping criteria:

$$\begin{array}{lcl} \text{primal infeasibility} & < & \epsilon \\ \text{dual infeasibility} & < & \epsilon \\ \text{average complementarity} & < & \epsilon \end{array}$$

The Failures

Warmstarting: Branch-and-Bound

- Optimal solution at the parent: $(x^*, y^*, g^*, t^*, \lambda^*, z^*, s^*)$.
- Current node: Branch on some variable y_j
- The following must hold:

$$l_j < y_j^* < u_j,$$

$$\begin{aligned} g_j^* &> 0, & z_j^* &= 0 \\ t_j^* &> 0, & s_j^* &= 0. \end{aligned}$$

- WLOG, assume that $l_j < y_j < \lfloor y_j^* \rfloor$
- The only term affected is $D_{tj}(u_j - y_j) = s_j(u_j - y_j)/t_j$.
- However, at the first iteration $s_j^*(u_j - y_j^*)/t_j^* = 0$.
- The algorithm will get stuck at this nonoptimal and, in fact, infeasible solution.

Warmstarting: Outer Approximation

$$\begin{array}{llll} \min_{x,y} & (x-0.25)^2 + y & & \\ \text{s.t.} & -60x^3 & \geq & -y \\ & y & \in & \{0,1\} \end{array}$$

- At each iteration of OA, an NLP subproblem is solved for a fixed value of y.
- The reduced KKT system:

$$\begin{bmatrix} -2-360x\lambda & -180x^2 \\ -180x^2 & \frac{w}{\lambda} \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} 2x-0.5+180x^2\lambda \\ \frac{\mu}{\lambda} - (y-60x^3) \end{pmatrix}$$

- Let $y = 1$ for the first subproblem. Then, $x^* = 0.25$, $w^* = 0.062$, and $\lambda^* = 0$.
- Let $y = 0$ for the next subproblem. In the first iteration, $\Delta x = 0$ and $\Delta y > 0$, but very close to 0. Then, $\Delta w = \frac{\mu}{\lambda} - w - \frac{w}{\lambda}\Delta\lambda = -1$. The steplength is shortened to less than 0.062.
- The algorithm becomes stuck at the old solution.

Infeasibility Identification

- An infeasible interior-point method does not have to start and/or stay feasible.
- Cannot get a certificate of infeasibility - only heuristics available.

Fixed Variables

- Consider the following problem:

$$\begin{array}{ll} \min\limits_{y} & y^2 \\ \text{s.t.} & 1 \leq y \leq 1. \end{array}$$

- The optimality conditions of this problem are:

$$\begin{aligned} y - g &= 1 \\ y + t &= 1 \\ 2y - z + s &= 0 \\ gz &= 0 \\ ts &= 0. \end{aligned}$$

- When $y = 1$, we have both g and t equal 0 and at the optimal solution, the dual variables z and s are free to take on any nonnegative values as long as they satisfy the equality

$$z - s = 2.$$

The Remedy

Previous Work on Warmstarting IPMs

- Approach: Find a suitable starting point
  - Identify an iterate close to the central path of the original problem
  - Modify the iterate so it is well-centered for the new problem
  - Solve the new problem from this point
- Works well in theory and practice: Gondzio (1998), Gondzio and Grothey (2003, 2006), Gondzio and Vial (1999), Yildirim and Wright (2002), John and Yildirim (2006)
- Mostly for LPs and QPs and only certain types of data perturbations

Previous Work on Warmstarting IPMs

- We propose a different approach: Change the problem, not the starting point
- Also investigated by Waltz & Ordonez, and Engau, Anjos, & Vanelli (2008)
- Our approach (Benson & Shanno (2005) and Benson (2007))
  - Corrects the numerical issues in the KKT system at the optimum of the original problem
  - Allows the nonnegative variables to become negative to encourage longer steps
  - Solves the new problem from the optimum of the original problem without modification

Primal-Dual Penalty Model

The primal problem:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & h(x,y) \geq 0 \\ & l \leq y \leq u. \end{aligned}$$

The primal penalty problem:

$$\begin{aligned} \min_{x,y,w,g,t,\xi_w,\xi_g,\xi_t} \quad & f(x,y) + c_w^T \xi_w + c_g^T \xi_g + c_t^T \xi_t \\ \text{s.t.} \quad & h(x,y) - w = 0 \\ & y - g = l \\ & y + t = u \\ & -\xi_w \leq w \leq b_\lambda \\ & -\xi_g \leq g \leq b_z \\ & -\xi_t \leq t \leq b_s \\ & \xi_w, \xi_g, \xi_t \geq 0, \end{aligned}$$

Primal-Dual Penalty Model

The dual problem:

$$\begin{array}{ll} \max\limits_{\lambda,z,s} & \text{dual\_obj}(\lambda, z, s; x, y) \\ \text{s.t.} & \nabla_x f(x,y) - A_x^T \lambda = 0 \\ & \nabla_y f(x,y) - A_y^T \lambda - z + s = 0 \\ & \lambda, z, s \geq 0. \end{array}$$

The dual penalty problem:

$$\begin{array}{ll} \max\limits_{\lambda,z,s} & \text{dual\_obj}(\lambda, z, s; x, y) - b_\lambda^T \psi_\lambda - b_z^T \psi_z - b_s^T \psi_s \\ \text{s.t.} & \nabla_x f(x,y) - A_x^T \lambda = 0 \\ & \nabla_y f(x,y) - A_y^T \lambda - z + s = 0 \\ & -\psi_\lambda \leq \lambda \leq c_w - \psi_\lambda \\ & -\psi_z \leq z \leq c_g - \psi_z \\ & -\psi_s \leq s \leq c_t - \psi_s \\ & \psi_\lambda, \psi_z, \psi_s \geq 0. \end{array}$$

Solving the Penalty Problem

First-order conditions:

$$
\begin{aligned}
h(x,y) - w &= 0 \\
y - g &= l \\
y + t &= u
\end{aligned}
\qquad
\begin{aligned}
\nabla_x f(x,y) - A_x^T \lambda &= 0 \\
\nabla_y f(x,y) - A_y^T \lambda - z + s &= 0
\end{aligned}
$$

$$
\begin{aligned}
(W + \Xi_w)(\Lambda + \Psi_\lambda)e &= \mu e \\
(G + \Xi_g)(Z + \Psi_z)e &= \mu e \\
(T + \Xi_t)(S + \Psi_s)e &= \mu e
\end{aligned}
\qquad
\begin{aligned}
\Xi_w(C_w - \Lambda - \Psi_\lambda)e &= \mu e \\
\Xi_g(C_g - Z - \Psi_z)e &= \mu e \\
\Xi_t(C_t - S - \Psi_s)e &= \mu e
\end{aligned}
$$

$$
\begin{aligned}
\Psi_\lambda(B_\lambda - W)e &= \mu e \\
\Psi_z(B_z - G)e &= \mu e \\
\Psi_s(B_s - T)e &= \mu e.
\end{aligned}
$$

The reduced KKT system has

$$
\begin{aligned}
E &= \left(\left((\Lambda + \Psi_\lambda)^{-1}(W + \Xi_w) + \Xi_w(C_w - \Lambda - \Psi_\lambda)^{-1} \right)^{-1} + \Psi_\lambda(B_\lambda - W)^{-1} \right)^{-1} \\
D_g &= \left((Z + \Psi_z)^{-1}(G + \Xi_g) + \Xi_g(C_g - Z - \Psi_z)^{-1} \right)^{-1} + \Psi_z(B_z - G)^{-1} \\
D_t &= \left((S + \Psi_s)^{-1}(T + \Xi_t) + \Xi_t(C_t - S - \Psi_s)^{-1} \right)^{-1} + \Psi_s(B_s - T)^{-1}
\end{aligned}
$$

and an appropriately modified rhs.

Exactness of the Penalty Model

Set the penalty parameters so that

$$\begin{array}{lllllll} b_\lambda & > & \bar{w}, & \quad & c_w & > & \bar{\lambda}, \\ b_z & > & \bar{g}, & & c_g & > & \bar{z}, \\ b_s & > & \bar{t}, & & c_t & > & \bar{s}. \end{array}$$

Then, the optimality conditions of the penalty problem reduce to the optimality conditions of the original problem.

Computational Issues: Initialization

How do we initialize the relaxation variables and the penalty parameters in order to reach the new optimum quickly after a warmstart?
Relaxation variables:

$$\begin{aligned} \xi_w &= \max(h(x,y) - w, 0) + \beta & \psi_\lambda &= \beta \\ \xi_g &= \max(x - g - l, 0) + \beta & \psi_z &= \beta \\ \xi_t &= \max(x + t - u, 0) + \beta & \psi_s &= \beta \end{aligned}$$

where β is a small parameter, currently set to $10^{-5}M$, where M is the greater of 1 and the largest primal or dual slack value. For discrete variables, $\beta = 1$.
Penalty parameters:

$$\begin{aligned} b_\lambda &= 2(w + \kappa) & c_w &= 2(\lambda + \psi_\lambda + \kappa) \\ b_z &= 2(g + \kappa) & c_g &= 2(z + \psi_z + \kappa) \\ b_s &= 2(t + \kappa) & c_t &= 2(s + \psi_s + \kappa), \end{aligned}$$

where κ is a constant with a default value of 1.

Computational Issues: Updates

If necessary, how do we update the penalty parameters?

- Static updates
- Dynamic updates

If	$w_i^{(k+1)} > 0.9 b_{\lambda i}^{(k)}$,	then	$b_{\lambda i}^{(k+1)} = 10 b_{\lambda i}^{(k)}$,	$i = 1, \ldots, m.$
If	$g_j^{(k+1)} > 0.9 b_{zj}^{(k)}$,	then	$b_{zj}^{(k+1)} = 10 b_{zj}^{(k)}$,	$j = 1, \ldots, p.$
If	$t_j^{(k+1)} > 0.9 b_{sj}^{(k)}$,	then	$b_{sj}^{(k+1)} = 10 b_{sj}^{(k)}$,	$j = 1, \ldots, p.$
If	$\lambda_i^{(k+1)} + \psi_{\lambda i}^{(k)} > 0.9 c_{wi} w^{(k)}$,	then	$c_{wi}^{(k+1)} = 10 c_{wi}^{(k)}$,	$i = 1, \ldots, m.$
If	$z_j^{(k+1)} + \psi_{zj}^{(k)} > 0.9 c_{gj}^{(k)}$,	then	$c_{gj}^{(k+1)} = 10 c_{gj}^{(k)}$,	$j = 1, \ldots, p.$
If	$s_j^{(k+1)} + \psi_{sj}^{(k)} > 0.9 c_{tj}^{(k)}$,	then	$c_{tj}^{(k+1)} = 10 c_{tj}^{(k)}$,	$j = 1, \ldots, p.$

Benefits of the Primal-Dual Penalty Model

- Warmstarting
- Primal and dual infeasibility/unboundedness detection
- Handling of fixed variables
- Bounded sets of optimal primal and dual solutions
- Handling of complementarity conditions
- Detection of nonKKT optima
- Relieving of the jamming phenomenon

One last improvement...

- A binary variable y can also be expressed as

$$0 \leq y \perp 1 - y \geq 0.$$

- Not guaranteed to give the optimal solution, but it can provide integer feasible solutions.
- Computational effort is not significantly more than solving one subproblem:

$$D = D_g + D_t - 2\tilde{\Lambda} + (2Y - I)\tilde{\Lambda}\tilde{W}^{-1}(2Y - I)$$

with an appropriately modified rhs.

The Results

Binary variables

Mixed-integer nonlinear programming problems from MINLPLib:

NAME	#MPEC	#node	f^*_{MPEC}	f^*	MPEC opt?
alan	6	20	2.925000E+000	2.925000E+000	Y
batch	38	38	4.128580E+005	2.855065E+005	N
batchdes	8	8	1.812017E+005	1.674277E+005	N
ex1223	10	20	4.579582E+000	4.579582E+000	Y
ex1223a	4	10	4.579582E+000	4.579582E+000	Y
ex1223b	6	14	4.579582E+000	4.579582E+000	Y
ex1225	16	16	4.800000E+001	3.100000E+001	N
ex3	14	26	8.237255E+001	6.800974E+001	N
ex4	158	158	2.891999E+005	-8.064136E+000	N
fac3	70	98	3.643282E+007	3.198231E+007	N
fuel	4	6	8.566119E+003	8.566119E+003	Y
gbd	2	4	2.200000E+000	2.200000E+000	Y
gkocis	8	8	-1.411002E+000	-1.923099E+000	N
johnall	2	305	-2.247302E+002	-2.247302E+002	Y
meanvarx	8	31	1.440406E+001	1.436923E+001	N
nous1	4	4	1.567072E+000	1.567072E+000	Y

Binary variables - 2

Mixed-integer nonlinear programming problems from MINLPLib:

NAME	#MPEC	#node	f^*_{MPEC}	f^*	MPEC opt?
oaer	2	2	-1.923098E+000	-1.923099E+000	Y
procsel	4	6	-7.273556E+000	-7.273556E+000	Y
ravem	198	198	4.071004E+005	2.695902E+005	N
st_e14	6	14	4.579582E+000	4.579582E+000	Y
st_miqp1	14	14	3.805000E+002	2.810000E+002	N
st_miqp4	2	2	-4.574000E+003	-4.574000E+003	Y
st_test1	14	14	3.000000E+000	4.367746E-012	N
st_test5	28	28	-1.100000E+002	-1.100000E+002	Y
st_test6	24	24	6.610000E+002	4.710000E+002	N
synthes1	4	4	7.092732E+000	6.009759E+000	N
synthes2	8	12	7.303531E+001	7.303531E+001	Y
synthes3	14	22	8.237255E+001	6.800974E+001	N

Warmstarting

Mixed-integer nonlinear programming problems from MINLPLib:

Problem	WarmIters	ColdIters	%Impr	#Nodes	#Inf	$f(x*)$
alan	8.40	9.20	8.70	6	1	2.93E+00
batch*	22.06	36.06	38.83	38	4	2.86E+05
batchdes	24.83	30.67	19.02	8	2	1.67E+05
du-opt	25.69	34.68	25.91	131	0	3.56E+00
du-opt5	14.54	24.31	40.19	109	18	8.07E+00
eg_all_s	14.61	24.71	40.88	192	7	7.92E+00
eg_disc_s	14.56	19.21	24.21	8	0	5.76E+00
eg_disc2_s	15	20.86	28.08	14	0	5.64E+00
eg_int_s	13.9	31.8	56.29	10	0	7.46E+00
ex1223	15.20	15.90	4.40	10	0	4.58E+00
ex1223a	14.80	14.60	-1.37	4	0	4.58E+00
ex1223b	12.67	12.83	1.30	6	0	4.58E+00
ex1225	14.92	12.58	-18.54	16	4	3.10E+01
ex3	16.21	16.93	4.22	14	0	6.80E+01
ex4	28.44	23.99	-18.55	158	0	-8.06E+00
fac3	24.66	25.38	2.83	70	18	3.20E+07

Warmstarting - 2

Mixed-integer nonlinear programming problems from MINLPLib:

Problem	WarmIters	ColdIters	%Impr	#Nodes	#Inf	$f(x*)$
fuel*	8.75	63.67	86.26	4	0	8.57E+03
gbd	8.00	7.00	-14.29	2	1	2.20E+00
gear	15.24	17.24	11.60	25	0	8.20E-05
gkocis	11.88	11.75	-1.06	8	0	-1.92E+00
johnall	2.00	13.00	84.26	2	0	-2.25E+02
meanvarx	9.50	14.13	32.74	8	0	1.44E+01
nous1	15.00	21.00	28.57	4	2	1.57E+00
nvs03	14	16.8	16.67	8	3	1.60E+01
nvs04	20.25	53.25	61.97	4	0	7.20E-01
nvs06	10.25	12.25	16.33	4	0	1.77E+00
nvs08	11.5	13.67	15.85	6	0	2.34E+01
nvs10	10	14.75	32.20	4	0	-3.11E+02
nvs11	9.38	14.25	34.21	8	0	-4.31E+02
nvs12	9.25	14.58	36.57	12	0	-4.81E+02
nvs13	10.25	15.5	33.87	20	0	-5.85E+02
nvs14	9.97	24.85	59.89	66	6	-4.04E+04

Warmstarting - 3

Mixed-integer nonlinear programming problems from MINLPLib:

Problem	WarmIters	ColdIters	%Impr	#Nodes	#Inf	$f(x*)$
nvs15	13	18	27.78	8	0	1.00E+00
nvs17	9.69	15.89	39.04	64	0	-1.10E+03
nvs18	9.88	15.02	34.26	48	0	-7.78E+02
nvs19	9.7	16.27	40.35	101	0	-1.10E+03
nvs20	12.13	12.5	3.00	24	0	2.31E+02
nvs23	10.63	27.49	61.34	218	2	-1.13E+03
nvs24	10.3	28.61	64.00	310	4	-1.03E+03
oaer	12.00	15.00	20.00	2	0	-1.92E+00
prob02	14	14	0.00	2	0	1.12E+03
prob03	9	7.75	-16.13	4	0	1.00E+01
procsel	8.00	8.50	5.88	4	0	-1.92E+00
ravem	28.03	38.67	27.53	198	6	2.70E+05
st_e14	12.67	12.83	1.30	6	0	4.58E+00
st_miqp1	8.30	7.90	-5.06	14	5	2.81E+02
st_miqp2	9	9.43	4.55	10	3	2.00E+00
st_miqp4	11.50	15.50	25.81	2	0	-4.57E+03

Warmstarting - 4

Mixed-integer nonlinear programming problems from MINLPLib:

Problem	WarmIters	ColdIters	%Impr	#Nodes	#Inf	$f(x*)$
st_test1	3.21	6.14	47.67	14	0	4.37E-12
st_test3	9.62	8.92	-7.76	13	0	-7.00E+00
st_test4	8.63	14.38	40.00	8	0	-7.00E+00
st_test5	10.57	15.14	30.19	28	14	-1.10E+02
st_test6	10.38	12.19	14.87	24	8	4.71E+02
st_test8	9	15.5	41.94	2	0	-2.96E+04
st_testgr1	11.19	16.74	33.13	94	6	-1.28E+01
st_testgr3	12.67	25.86	51.03	1196	198	-2.06E+01
st_testph4	8.25	9	8.33	4	0	-8.05E+01
synthes1	12.00	10.50	-14.29	4	0	6.01E+00
synthes2	14.13	16.88	16.30	8	0	7.30E+01
synthes3	16.31	16.54	1.40	14	0	6.80E+01
tloss	47.34	119.31	60.33	101	15	1.63E+01
OVERALL	15.60	27.67	43.65	3490	327	

MILPs

Mixed-integer linear programming problems solved using branch-and-bound:

Problem	WarmIters	ColdIters	△
Diet-2	6	11	1
Diet-3	6	10	1
Diet-4	6	10	1
Diet-5	6	10	1
Diet-6	6	11	1
Diet-7	6	11	1
Diet-8	6	10	2
Diet-9	6	11	1
Diet-10	7	11	1
Diet-11	6	10	1
HL415-2	7	12	1
HL415-3	8	11	2
HL415-4	(inf)	(inf)	(inf)
HL415-5	9	10	1
HL415-6	7	11	1
HL415-7	7	11	1

Problem	WarmIters	ColdIters	△
Synthes3-2	11	15	5
Synthes3-3	9	14	4
Synthes3-4	11	15	1
Synthes3-5	9	14	2
Synthes3-6	9	13	3
Synthes3-7	10	13	1
Synthes3-8	10	15	2
Synthes3-9	9	13	3
Synthes3-10	11	14	1
Synthes3-11	10	16	4
Synthes3-12	9	15	4
Synthes3-13	10	15	1
Synthes3-14	9	14	2
Synthes3-15	10	15	2

Cutting Stock

Master problems of the cutting stock model:

Problem	WarmIters	ColdIters	Δ
Master-2	6	11	5
Master-3	6	11	5
Master-4	6	11	5

Footnote

Mixed-Integer SOCP

- Interesting application areas, e.g. portfolio optimization with cardinality constraints, facility location problems with fixed costs.
- Interior-point methods have good convergence properties and computational performance.
- Homogeneous self-dual approach allows for a similar type of warmstart capability.
  - A naive implementation of SeDuMi + Outer Approximation
  - Looking to implement a re-centering scheme
- The primal-dual penalty approach also extends naturally to SOCPs.

Conclusion

Conclusion

- Interior-point methods can be warmstarted when regularization is used.
- Primal-dual regularization allows for warmstarts after any change to the problem.

- Moral of the story: We’re working on it!