AN INTERIOR–POINT ALGORITHM FOR NONCONVEX NONLINEAR PROGRAMMING

ROBERT J. VANDERBEI

JOINT WORK WITH D. SHANNO

TECH REPORT SOR 97-21 AVAILABLE AT:
http://www.princeton.edu/∼rvdb/

The Basic Interior-Point Paradigm

Start with an optimization problem—for now, the simplest NLP:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h_i(x) \geq 0, \qquad i = 1, \ldots, m \end{array}$$

Introduce slack variables to make all inequality constraints into nonnegativities:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) - w = 0, \\ & w \geq 0 \end{array}$$

Replace nonnegativity constraints with *logarithmic barrier terms* in the objective:

$$\begin{array}{ll} \text{minimize} & f(x) - \mu \sum_{i=1}^{m} \log(w_i) \\ \text{subject to} & h(x) - w = 0 \end{array}$$

Incorporate the equality constraints into the objective using *Lagrange multipliers*:

$$\text{minimize} \quad f(x) - \mu \sum_{i=1}^{m} \log(w_i) - y^T (h(x) - w)$$

Set all derivatives to zero:

$$\begin{aligned} \nabla f(x) - \nabla h(x)^T y &= 0 \\ -\mu W^{-1} e + y &= 0 \\ h(x) - w &= 0 \end{aligned}$$

Rewrite system:

$$\begin{aligned} \nabla f(x) - \nabla h(x)^T y &= 0 \\ WYe &= \mu e \\ h(x) - w &= 0 \end{aligned}$$

Apply Newton's method to compute *search directions*, Δx, Δw, Δy:

$$\begin{bmatrix} H(x,y) & 0 & -A(x)^T \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A(x)^T y \\ \mu e - WYe \\ -h(x) + w \end{bmatrix}.$$

Here,

$$H(x,y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 h_i(x)$$

and

$$A(x) = \nabla h(x)$$

Use second equation to solve for Δw. Result is the *reduced KKT system*:

$$\begin{bmatrix} -H(x,y) & A^T(x) \\ A(x) & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A^T(x) y \\ -h(x) + \mu Y^{-1} e \end{bmatrix}$$

Iterate:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \alpha^{(k)} \Delta x^{(k)} \\ w^{(k+1)} &= w^{(k)} + \alpha^{(k)} \Delta w^{(k)} \\ y^{(k+1)} &= y^{(k)} + \alpha^{(k)} \Delta y^{(k)} \end{aligned}$$

Modifications for Convex Optimization

For convex nonquadratic optimization, it does not suffice to choose the steplength α simply to maintain positivity of nonnegative variables.

- Consider, e.g., minimizing $f(x) = (1 + x^2)^{1/2}$.

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick *merit function*

$$\Psi_{\beta,\mu}(x, w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2} \|h(x) - w\|_2^2.$$

Define the *dual normal matrix*:

$$N(x, y, w) = H(x, y) + A^T(x) W^{-1} Y A(x).$$

Theorem 1. *Suppose that $N(x, y, w)$ is positive definite.*

(1) *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for the merit function $\Psi_{\beta,\mu}$.*
(2) *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

Modifications for NonConvex Optimization

If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ *might* fail to be positive definite.

In such a case, we lose the descent properties given in previous theorem.

To regain those properties, we perturb the Hessian: $\hat{H}(x, y) = H(x, y) + \lambda I$.

And compute search directions using $\hat{H}$ instead of H.

Notation: let $\hat{N}$ denote the dual normal matrix associated with $\hat{H}$.

Theorem 2. *If $\hat{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

(1) *the primal infeasibility, $\|h(x) - w\|$;*
(2) *the noncomplementarity, $w^T y$.*

Notes:

- *Not necessarily* a descent direction for *dual infeasibility*.
- A line search is performed to find a value of λ within a factor of 2 of the smallest permissible value.

Modifications for General Problem Formulations

Bounds, ranges, and free variables are all treated implicitly as described in *Linear Programming: Foundations and Extensions (LP:F&E)*.

Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

Here, D and E are positive definite diagonal matrices.

Note that D helps reduce frequency of diagonal perturbation.

Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.

Stopping rules, matrix reordering heuristics, etc. are as described in *LP:F&E*.

Computational Results

Compared:

- MINOS version 5.4 (19940910)
- LANCELOT version 20/03/1997
- LOQO version 3.10 (19971027)
- SNOPT version 5.3-2 (May 1998), driver 19980605

Input/output interface: AMPL with Hessian info (recent enhancement due to hard work of David Gay).

Hock and Schittkowski Problems

Name	Time in Seconds Minos	Lancelot	Loqo	Name	Time in Seconds Minos	Lancelot	Loqo	Name	Time in Seconds Minos	Lancelot	Loqo
hs001	0.02	0.11	0.06	hs040	0.01	0.04	0.03	hs080	0.04	0.06	0.04
hs002	0.00	0.04	0.04	hs041	0.00	0.04	0.04	hs081	0.05	0.07	0.07
hs003	0.00	0.05	0.03	hs042	0.01	0.04	0.03	hs083	0.01	0.07	0.05
hs004	0.00	0.05	0.03	hs043	0.05	0.08	0.03	hs084	0.03	(3)	0.06
hs005	0.01	0.04	0.02	hs044	0.00	0.05	0.03	hs085	0.49	(7)	1.19
hs006	0.08	0.11	0.05	hs045	0.00	0.01	0.09	hs086	0.01	0.14	0.06
hs007	0.08	0.06	0.03	hs046	0.09	0.08	0.05	hs087	0.05	0.23	0.08
hs008	0.01	0.04	0.03	hs047	0.10	0.08	0.10	hs088	0.50	(3)	1.06
hs009	0.00	0.06	0.03	hs048	0.01	0.02	0.04	hs089	1.02	(3)	2.88
hs010	0.04	0.06	0.03	hs049	0.02	0.08	0.06	hs090	(6)	(3)	1.34
hs011	0.03	0.05	0.03	hs050	0.01	0.04	0.04	hs091	(4)	10.36	1.98
hs012	0.03	0.07	0.03	hs051	0.00	0.04	0.06	hs092	9.74	23.84	1.93
hs013	0.03	0.11	(4)	hs052	0.00	0.03	0.03	hs093	0.06	(1)	0.04
hs014	0.01	0.05	0.03	hs053	0.00	0.03	0.03	hs095	0.01	0.16	0.06
hs015	0.01	0.11	0.06	hs054	0.01	0.04	0.06	hs096	0.01	0.17	0.07
hs016	0.01	0.08	0.04	hs055	0.00	0.03	0.04	hs097	0.09	0.13	0.06
hs017	0.01	0.07	0.08	hs056	0.05	0.04	0.04	hs098	0.04	0.13	0.06
hs018	0.13	0.39	0.04	hs057	0.04	0.08	0.09	hs099	0.05	(4)	0.14
hs019	0.04	0.19	0.06	hs059	0.09	0.83	0.04	hs100	0.11	0.25	0.03
hs020	0.01	0.08	0.05	hs060	0.09	0.05	0.05	hs101	2.03	(4)	0.61
hs021	0.00	0.05	0.04	hs061	0.04	0.05	0.03	hs102	2.04	(4)	0.26
hs022	0.02	0.04	0.02	hs062	0.01	0.09	0.03	hs103	4.48	(4)	0.24
hs023	0.04	0.13	0.04	hs063	0.12	0.05	0.03	hs104	0.32	(1)	0.05
hs024	0.00	0.05	0.04	hs064	0.09	0.09	0.05	hs105	2.59	(4)	6.78
hs025	0.01	0.06	0.24	hs065	0.06	0.17	0.04	hs106	0.17	(4)	0.11
hs026	0.07	0.09	0.04	hs066	0.02	0.04	0.04	hs107	0.04	(4)	0.25
hs027	0.09	0.07	0.20	hs067	(5)	(5)	(5)	hs108	0.13	0.33	0.12
hs028	0.00	0.03	0.03	hs068	(2)	(2)	0.12	hs109	0.30	(4)	0.35
hs029	0.04	0.05	0.03	hs069	(2)	(2)	0.04	hs110	0.02	0.06	0.05
hs030	0.03	0.05	0.03	hs070	0.17	4.73	0.42	hs111	0.41	0.34	0.16
hs031	0.02	0.04	0.03	hs071	0.04	0.06	0.03	hs112	0.05	0.29	0.06
hs032	0.01	0.03	0.06	hs072	0.07	(3)	0.05	hs113	0.11	0.91	0.05
hs033	0.01	0.04	0.05	hs073	0.02	0.07	0.05	hs114	0.09	3.43	0.15
hs034	0.05	0.05	0.03	hs074	0.03	0.07	0.04	hs116	0.37	(4)	0.18
hs035	0.00	0.04	0.03	hs075	0.02	(3)	0.04	hs117	0.13	0.55	0.10
hs036	0.00	0.04	0.03	hs076	0.01	0.05	0.03	hs118	0.03	0.23	0.07
hs037	0.01	0.04	0.03	hs077	0.10	0.08	0.04	hs119	0.04	0.29	0.30
hs038	0.03	0.12	0.09	hs078	0.04	0.05	0.03				
hs039	0.06	0.05	0.04	hs079	0.04	0.06	0.03				

Legend: (1) Could not find a feasible soluton. (2) Erf() not available. (3) Step got too small. (4) Too many iterations. (5) Could not code model in AMPL. (6) Unbounded or badly scaled. (7) Core dump.

Hock and Schittkowski Problems—Iteration Counts

Name	Iters	Name	Iters	Name	Iters	Name	Iters	Name	Iters
hs001	32	hs025	15	hs048	15	hs073	20	hs098	19
hs002	19	hs026	15	hs049	24	hs074	13	hs099	20
hs003	11	hs027	55	hs050	16	hs075	15	hs100	11
hs004	10	hs028	11	hs051	18	hs076	11	hs101	40
hs005	10	hs029	10	hs052	10	hs077	13	hs102	24
hs006	17	hs030	11	hs053	13	hs078	9	hs103	24
hs007	10	hs031	9	hs054	18	hs079	9	hs104	14
hs008	9	hs032	25	hs055	13	hs080	11	hs105	21
hs009	10	hs033	17	hs056	12	hs081	19	hs106	33
hs010	15	hs034	14	hs057	19	hs083	15	hs107	56
hs011	12	hs035	11	hs059	17	hs084	18	hs108	23
hs012	10	hs036	14	hs060	18	hs085	48	hs109	49
hs014	11	hs037	11	hs061	11	hs086	15	hs110	12
hs015	33	hs038	44	hs062	14	hs087	25	hs111	17
hs016	20	hs039	15	hs063	10	hs088	25	hs112	17
hs017	36	hs040	9	hs064	28	hs089	34	hs113	16
hs018	17	hs041	17	hs065	14	hs090	25	hs114	31
hs019	31	hs042	9	hs066	13	hs091	29	hs116	33
hs020	24	hs043	11	hs068	41	hs092	27	hs117	22
hs021	16	hs044	11	hs069	15	hs093	10	hs118	17
hs022	9	hs045	38	hs070	19	hs095	18	hs119	33
hs023	16	hs046	20	hs071	12	hs096	22		
hs024	16	hs047	37	hs072	21	hs097	18		

Comments:

- MINOS won 72 times, LOQO 44 times, and LANCELOT 4 times.
- LOQO found local optima that were worse than reported values on 8 problems (002, 016, 020, 025, 059, 070, 097, and 098).
- LOQO found local optima that were **better** than reported values on 7 problems (047, 088–092, 109).

Larger Real-World Problems

Name	m	n	nonz(A)	nonz(H)	LOQO	MINOS	LANCELOT	SNOPT
antenna	167	49	8014	2303	1m1s	*2m34s	*19m41s	**
fekete2	50	150	150	22500	1m18s	24s	*6m 1s	28s
markowitz2	201	1200	201200	200	4m43s	1m56s	13m35s	**
minsurf	172	1849	172	12601	21s	*13m37s	2m7s	*2m42s
polygon2	195	42	766	880	0.7s	$>$60m0s	1m1s	3.1s
sawpath	198	5	784	25	1.7s	4.9s	*8s	340s
structure4	720	1536	5724	20356	2m40s	*43m7s	**	*1h33m
trafequil2	628	1194	5512	76	5.1s	5.3s	2m39s	3.8s

Legend:

(*) either infeasible or suboptimal.
(**) not enough memory.

Final Comments

Bounded vs. free variables.

Bound pushing.

Presolve.

Artificial cusps (e.g. $(\cdots)^{1/4}$).

Scaling.

Singular Hessian can be a point of attraction.

- Consider, e.g., $x(x-1)(x+1)+6=0$.