LaGO - Branch and Cut for nonconvex MINLPs

Stefan Vigerske

Humboldt-University Berlin, Department of Mathematics

Advances in Global Optimization: Methods and Applications
Myconos, 16.6.2007

Lagrangian Global Optimizer

General purpose solver for sparse, block-separable, nonconvex MINLPs (not rigorous, partially heuristic)

2000	Development started by Ivo Nowak as a solver for nonconvex MIQQPs based on Lagrangian decomposition and semidefinite relaxation
2001-2004	Project funded by German Science Foundation: extension to MINLP solver
2006	start of COIN-OR project, LaGO code becomes public
now	Linear-relaxation based Branch and Cut algorithm version 0.3 (work in progress)

Webpage: `https://projects.coin-or.org/LaGO`

Book: Ivo Nowak, Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming, Birkhäuser 2005

Paper: LaGO - a (heuristic) Branch and Cut algorithm for nonconvex MINLPs, submitted 2006

Overview

Ingredients of LaGOs Branch and Cut Algorithm

- Preprocessing
- Underestimators and Cutting Planes
- Boxreduction

Numerical Experiments

Future Developments

MINLP

We consider mixed-integer nonlinear problems (MINLP) of the form

$$\begin{array}{lll} \text{minimize} & c^T x & \\ \text{such that} & h_i(x) \leq 0, & i \in I, \\ & h_j(x) = 0, & j \in E, \\ & x_k \in \mathbb{Z}, & k \in B, \\ & x \in [\underline{x}, \bar{x}] & \end{array}$$

$-\infty < \underline{x}_i \leq \bar{x}^i < \infty$, $i \in \{1, \ldots, n\}$, $c \in \mathbb{R}^n$

- $h \in C^2\left([\underline{x}, \bar{x}], \mathbb{R}^{|I|+|E|}\right)$ are black-box functions
  LaGO needs
  - methods for the evaluation of values, gradients, and Hessians
  - optional: sparsity of jacobian, interval-arithmetic evaluations
- LaGO interfaces problems via GAMS and AMPL

Overview

Ingredients of LaGOs Branch and Cut Algorithm

- Preprocessing
- Underestimators and Cutting Planes
- Boxreduction

Numerical Experiments

Future Developments

Preprocessing

- Investigation of problem structure:
  - sparsity of Jacobian (given by interface) and Hessian (by sampling)

Preprocessing

- Investigation of problem structure:
  - sparsity of Jacobian (given by interface) and Hessian (by sampling)
  - quadratic terms (Hessian in sample points), block-separability

$$h_i(x) = \text{const} + b^T x + \sum_k x_{Q_k}^T A_k x_{Q_k} + \sum_r g_r(x_{N_r})$$

for “small” disjoint subsets Q_k and N_r of $\{1, \ldots, n\}$

Preprocessing

- Investigation of problem structure:
  - sparsity of Jacobian (given by interface) and Hessian (by sampling)
  - quadratic terms (Hessian in sample points), block-separability

$$h_i(x) = \text{const} + b^T x + \sum_k x_{Q_k}^T A_k x_{Q_k} + \sum_r g_r(x_{N_r})$$

    for "small" disjoint subsets Q_k and N_r of $\{1, \dots, n\}$
  - convexity (evaluate eigenvalues of Hessian in sample points)

Preprocessing

- Investigation of problem structure:
  - sparsity of Jacobian (given by interface) and Hessian (by sampling)
  - quadratic terms (Hessian in sample points), block-separability

$$h_i(x) = \text{const} + b^T x + \sum_k x_{Q_k}^T A_k x_{Q_k} + \sum_r g_r(x_{N_r})$$

  for "small" disjoint subsets Q_k and N_r of $\{1, \ldots, n\}$
  - convexity (evaluate eigenvalues of Hessian in sample points)
- Reduction of box $[\underline{x}, \bar{x}]$:
  - simple constraint propagation (based on interval arithmetic)
  - enclosing a polyhedron defined by linear constraints
  - bounding box for still unbounded variables by "guessing"

Preprocessing

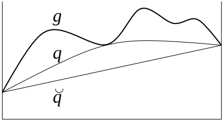
- Investigation of problem structure:
  - sparsity of Jacobian (given by interface) and Hessian (by sampling)
  - quadratic terms (Hessian in sample points), block-separability

$$h_i(x) = \text{const} + b^T x + \sum_k x_{Q_k}^T A_k x_{Q_k} + \sum_r g_r(x_{N_r})$$

  for "small" disjoint subsets Q_k and N_r of $\{1, \ldots, n\}$

  - convexity (evaluate eigenvalues of Hessian in sample points)
- Reduction of box $[\underline{x}, \bar{x}]$:
  - simple constraint propagation (based on interval arithmetic)
  - enclosing a polyhedron defined by linear constraints
  - bounding box for still unbounded variables by "guessing"
- Initialization of relaxations:

  1. quadratic (nonconvex) underestimator q
  2. quadratic convex underestimator $\breve{q}$
  3. linearization, dropping integrality restrictions



Quadratic underestimators

Let $g \in C^2([\underline{x}, \bar{x}], \mathbb{R})$ be nonquadratic and nonconvex.

Compute an underestimator $q(x) = x^T A x + b^T x + c$ by

$$\begin{aligned} \min_{A,b,c} \quad & \sum_{x \in S} g(x) - q(x) \\ \text{such that} \quad & q(x) \leq g(x) \qquad x \in S \\ & q(\hat{x}) = g(\hat{x}) \end{aligned}$$

for a sample set $S \subseteq [\underline{x}, \bar{x}]$ and a reference point $\hat{x}$.

Quadratic underestimators

Let $g \in C^2\left([\underline{x}, \bar{x}], \mathbb{R}\right)$ be nonquadratic and nonconvex.

Compute an underestimator $q(x) = x^T A x + b^T x + c$ by

$$\begin{array}{lll} \min\limits_{A,b,c} & \sum\limits_{x \in S} g(x) - q(x) & \\ \text{such that} & q(x) \leq g(x) & x \in S \\ & q(\hat{x}) = g(\hat{x}) & \end{array}$$

for a sample set $S \subseteq [\underline{x}, \bar{x}]$ and a reference point $\hat{x}$.

- Sparsity of A and b determined by $g(x)$.
- Quality of $q(x)$ depends strongly on the choice of the sample set S

Quadratic underestimators

Let $g \in C^2([\underline{x}, \bar{x}], \mathbb{R})$ be nonquadratic and nonconvex.

Compute an underestimator $q(x) = x^T A x + b^T x + c$ by

$$\begin{aligned} \min_{A,b,c} \quad & \sum_{x \in S} g(x) - q(x) \\ \text{such that} \quad & q(x) \leq g(x) \qquad x \in S \\ & q(\hat{x}) = g(\hat{x}) \end{aligned}$$

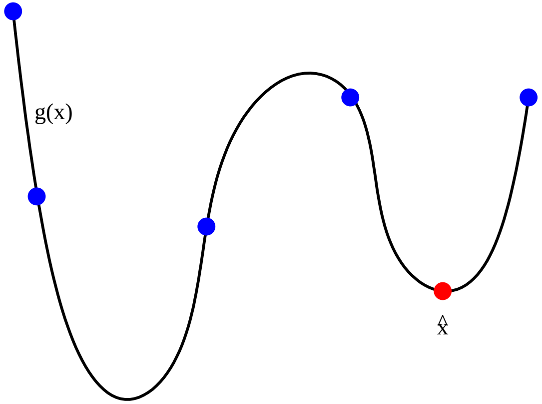
for a sample set $S \subseteq [\underline{x}, \bar{x}]$ and a reference point $\hat{x}$.

- Sparsity of A and b determined by $g(x)$.
- Quality of $q(x)$ depends strongly on the choice of the sample set S

$\Rightarrow$ A. Neumaier 2006: adaptive choice of S

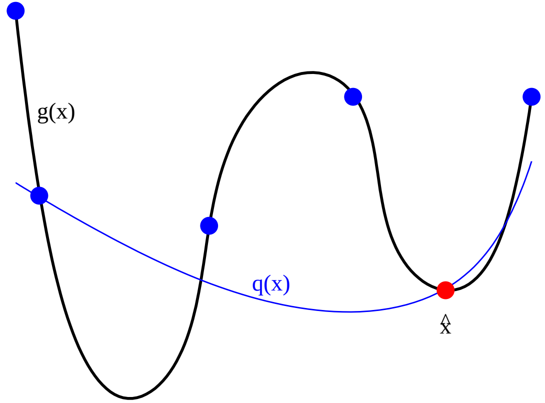
Nonconvex quadratic underestimator (cont.)

- initial choice: $S = \text{vert}([\underline{x}, \bar{x}]) \cup \left\{x_{\min}, \frac{1}{2}(\underline{x} + \bar{x})\right\} \cup M$ with $\hat{x} := x_{\min}$ a local minimum of $g(x)$ and M a set of random points



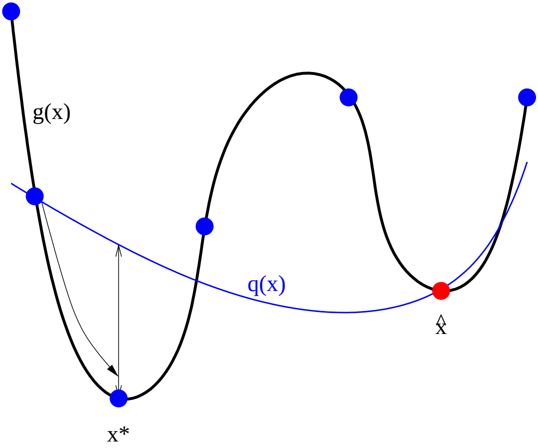
Nonconvex quadratic underestimator (cont.)

- initial choice: $S = \text{vert}([\underline{x}, \bar{x}]) \cup \left\{x_{\min}, \frac{1}{2}(\underline{x} + \bar{x})\right\} \cup M$ with $\hat{x} := x_{\min}$ a local minimum of $g(x)$ and M a set of random points



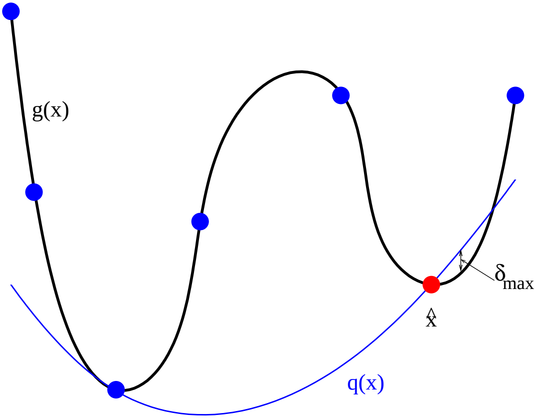
Nonconvex quadratic underestimator (cont.)

- for $x \in S$ with $g(x) = q(x)$, maximize the error $q(x) - g(x) \Rightarrow x^*$
- if $q(x^*) - g(x^*) > \delta_{\text{tol}}$, add x^* to S and recompute $q(x)$



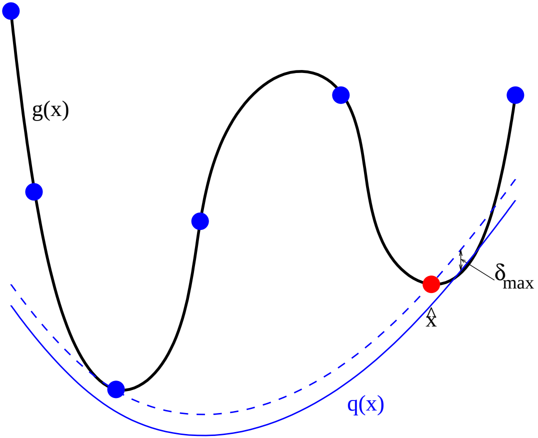
Nonconvex quadratic underestimator (cont.)

- for $x \in S$ with $g(x) = q(x)$, maximize error $q(x) - g(x) \Rightarrow \delta_{\max}$



Nonconvex quadratic underestimator (cont.)

- for $x \in S$ with $g(x) = q(x)$, maximize error $q(x) - g(x) \Rightarrow \delta_{\text{max}}$
- if $\delta_{\text{max}} < \delta_{\text{tol}}$, lower $q(x)$ by δ_{max}



Convex underestimators

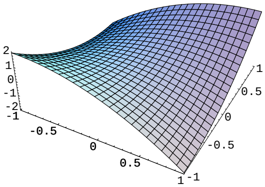
Let $q(x) = x^T A x + b^T x + c$ be a quadratic nonconvex function.

A convex α-underestimator (Adjiman and Floudas 1997) of $q(x)$ is

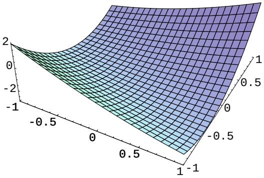
$$\breve{q}(x) = q(x) + \sum_{i=1}^{n} \alpha_i (x_i - \underline{x}_i)(x_i - \bar{x}_i)$$

where

$$\alpha_i = -\lambda_1 \left(\mathrm{Diag}(\bar{x} - \underline{x})\ A\,\mathrm{Diag}(\bar{x} - \underline{x})\right)\ (\bar{x}_i - \underline{x}_i)^{-2}\,.$$



$\Rightarrow$



- Linearizations of $\breve{q}(x)$ are easily updated after reducing the box $[\underline{x}, \bar{x}]$

Cuts for the Linear Relaxation

- for the nonlinear constraints: linearization of convexified constraints

Cuts for the Linear Relaxation

- for the nonlinear constraints: linearization of convexified constraints
- for the integrality constraints: COIN-OR Cut Generator Library (Cgl)
- COIN/Cgl provides several types of cuts to cut off a nonintegral solution of an LP

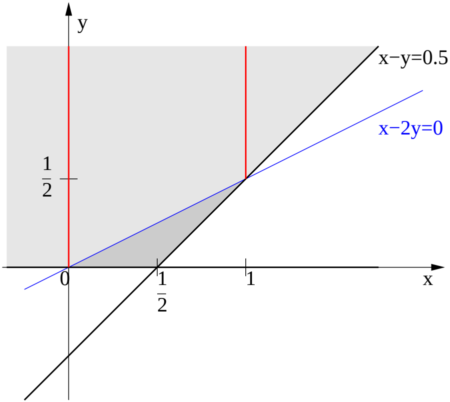
Cuts for the Linear Relaxation

- for the nonlinear constraints: linearization of convexified constraints
- for the integrality constraints: COIN-OR Cut Generator Library (Cgl)
- COIN/Cgl provides several types of cuts to cut off a nonintegral solution of an LP

Mixed Integer Rounding Cut
(Nemhauser, Wolsey 1988) principle:

$$X := \{(x, y) \in \mathbb{Z} \times \mathbb{R}_+ \, | x - y \leq b\}$$

$$x - \frac{1}{1 - (b - \lfloor b \rfloor)} y \leq \lfloor b \rfloor \quad \forall (x, y) \in X$$



Boxreduction by (simple) Constraint Propagation

- Consider a constraint

$$y \leq h(x) \qquad \text{and let} \qquad [\underline{h}, \bar{h}] := h([\underline{x}, \bar{x}])$$

(i.e., $h(x) \in [\underline{h}, \bar{h}]\ \forall x \in [\underline{x}, \bar{x}]$).

- If $\bar{h} < \bar{y}$, set

$$\bar{y} := \bar{h}$$

and proceed with other constraints depending on y.

- If $\bar{y} < \underline{h}$, then the node is infeasible.

Boxreduction by (simple) Constraint Propagation

- Consider a constraint

$$y \leq h(x) \qquad \text{and let} \qquad [\underline{h}, \bar{h}] := h([\underline{x}, \bar{x}])$$

(i.e., $h(x) \in [\underline{h}, \bar{h}]\ \forall x \in [\underline{x}, \bar{x}]$).

- If $\bar{h} < \bar{y}$, set

$$\bar{y} := \bar{h}$$

and proceed with other constraints depending on y.

- If $\bar{y} < \underline{h}$, then the node is infeasible.

- Does not rely on relaxations.
- Easy and fast to compute (GAMS-interface and FILIB++).
- Only one constraint at a time.

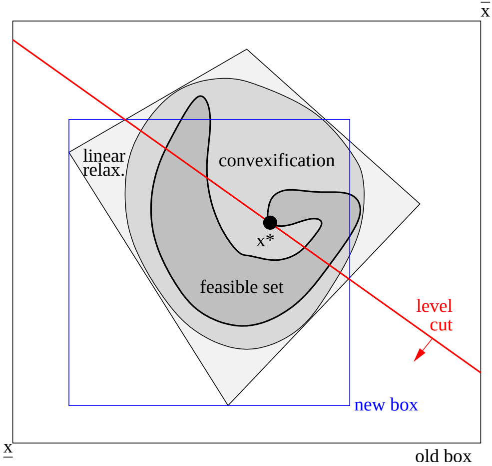
Boxreduction by enclosing the linear relaxation

Consider a linear relaxation with constraints $Ax \leq b$.

Let x^* be the best solution found so far.

$$\underline{x}_i := \min x_i \quad \text{s.t. } Ax \leq b, \quad c^T x \leq c^T x^*$$

$$\overline{x}_i := \max x_i \quad \text{s.t. } Ax \leq b, \quad c^T x \leq c^T x^*$$



Overview

Ingredients of LaGOs Branch and Cut Algorithm

- Preprocessing
- Underestimators and Cutting Planes
- Boxreduction

Numerical Experiments

Future Developments

GAMS MINLPLib and GlobalLib

- at most 1000 variables and no sin, cos, errorf
  ⇒ 77 MIQQPs, 127 (nonquadratic) MINLPs, 162 QQPs
- timelimit: 1 hour
- NLP subsolver: CONOPT; LP subsolver: CPLEX 10.0

	MIQQPs	MINLPs
number of models	77	127
best known optimal solution found	60	68
local optimal solution found	1	18
no feasible point found	16	41

Note on MINLPs:

- LaGO computes only one quadratic underestimator per function and does not update it in Branch and Cut (implementation issue)

⇒ stop of branching when all discrete variables are fixed

LaGO vs. BARON on MINLPs

LaGO and BARON 7.8.1 on (nonquadratic) MINLPs from MINLPLib:

	Total	optimal value: LaGO better	same	BARON better
BARON fail, LaGO not	9	9		
LaGO faster	12	1	7	4
both solvers the same	11		5	6
BARON faster	54		46	8
LaGO fail, BARON not	34			34
LaGO and BARON fail	7		7	
Total	127	10	65	52

convex MINLPs: 59

LaGO on MINLPs - details: up to 20 variables

example	n	$\|I\|$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
batchdes	20	9	20	n	0	$< 1\%$	1	0.02	0.37	=
ex1221	6	3	6	n	0	1.6%	4	0.02	0.07	=
ex1222	4	1	4	n	0	$< 1\%$	1	0.01	0.05	=
ex1223	12	4	14	y	0	$< 1\%$	3	0.04	0.04	=
ex1223b	8	4	10	y	0	$< 1\%$	4	0.04	0.03	=
ex1224	12	8	8	n	0	2.4%	115	0.75	0.36	=
ex1225	9	6	11	n	0	$< 1\%$	8	0.08	0.15	=
ex1226	6	3	6	n	0	$< 1\%$	3	0.01	0.12	=
gkocis	12	3	9	n	0	$< 1\%$	3	0.03	0.11	=
hmittelman	17	16	8	n	0	$< 1\%$	300	0.27	3.49	=
oaer	10	3	8	n	0	$< 1\%$	4	0.03	0.15	=
procsel	11	3	8	n	0	$< 1\%$	4	0.04	0.10	=
st_e14	12	4	14	y	0	$< 1\%$	3	0.04	0.05	=
st_e15	6	3	6	n	0	7.1%	6	0.03	0.07	=
st_e29	12	8	8	n	0	2.4%	115	0.77	0.35	=
synthes1	7	3	7	y	0	$< 1\%$	4	0.04	0.08	=
synthes2	12	5	15	y	0	$< 1\%$	8	0.07	0.07	=
synthes3	18	8	24	y	0	$< 1\%$	8	0.14	0.11	=
eg_all_s	8	7	28	n	infeas	$< 1\%$	subsolver reports infeas. point			f
eg_disc_s	8	4	28	n	0	54.4%	3078	42:31.67	1:25.07	f
eg_disc2_s	8	3	28	n	0	57.5%	3736	57:42.83	2:17.61	f
eg_int_s	8	3	28	n	0	42%	28	20.07	57.37	f
gear	5	4	1	n	0	$< 1\%$	4	0.00	0.18	=
gear3	9	4	5	n	0	$< 1\%$	4	0.02	0.20	=
gear4	7	4	2	n	100%	96.8%	50000	11:35.34	0.55	+

LaGO on MINLPs - details: up to 20 variables

example	n	$\lvert I \rvert$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
nvs01	4	2	4	n	0	$< 1\%$	233	2.02	0.72	=
nvs02	9	5	4	n	0	$< 1\%$	668	9.84	0.19	=
nvs04	3	2	1	n	0	$< 1\%$	201	0.12	2.07	=
nvs05	9	2	10	n	N/A		bad modeling*			+
nvs06	3	2	1	n	0	$< 1\%$	7	0.00	0.14	=
nvs07	4	3	3	n	0	$< 1\%$	4	0.00	0.20	=
nvs08	4	2	4	n	0	22.3%	19	0.10	0.24	=
nvs09	11	10	1	n	0	$< 1\%$	10	0.00	0.49	=
nvs16	3	2	1	n	0	$< 1\%$	400	0.25	0.18	=
nvs20	17	5	9	y	0	$< 1\%$	5	0.15	0.37	=
nvs21	4	2	3	n	84%	96.9%	1306	8.33	0.86	+
nvs22	9	4	10	n	0	$< 1\%$	22	0.24	1:16.29	=
prob10	3	1	3	n	0	17.8%	3	0.02	0.50	f
spring	18	12	9	n	0	43.6%	665	6.54	1.52	=
st_e36	3	1	3	n	infeas	-9.3%	subsolver reports infeas. point			+
st_e38	5	2	4	n	0	47.8%	395	2.05	0.40	=
st_e40	5	3	9	n	0	$< 1\%$	58	0.60	0.32	=
windfac	15	3	14	n	0	20.3%	15	0.18	22.37	f

LaGO on MINLPs - details: 21 - 100 variables

example	n	$\lvert I \rvert$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
batch	47	24	74	y	0	< 1%	4	0.16	0.32	=
csched1	77	63	23	n	4%	-66%	wrong guess on variable bound*			+
ex1233	53	12	65	n	0	25.8%	175	3.21	2.83	=
ex1243	69	16	97	n	0	33.5%	108	2.64	4.01	=
ex1244	96	23	130	n	0	21.1%	2474	1:38.14	4.36	=
ex1252	40	15	44	n	0	96.5%	2727	51.60	2.87	f
ex3	33	8	32	n	0	< 1%	5	0.08	0.44	=
fac1	23	6	19	y	7%	< 1%	bad scaled objective function*			+
fac2	67	12	34	n	0	99.4%	62	0.73	6.34	=
feedtray	98	7	92	n	0	81%	13	1.53	19.92	=
gasnet	91	10	70	n	0	98.9%	2047	2:44.73	13.05	f
gear2	29	24	5	n	0	< 1%	53	0.39	0.25	=
m3	27	6	44	y	0	< 1%	20	0.44	0.12	=
m6	87	30	158	y	0	< 1%	15123	26:40.97	0.76	=
minlphix	85	20	93	n	0	99.7%	44	0.85	21.78	–
ortez	88	18	75	n	0	2.2%	119	1.77	1.49	=
st_e35	33	7	40	y	15%	< 1%	failure in convexity check*			+
synheat	57	12	65	n	0	28.1%	257	4.29	2.22	=
waterx	71	14	55	n	3%	110.9%	13216	8:47.74	6.48	+
ex1252a	25	9	35	n	0	95.1%	3266	58.08	2.38	f
pump	25	9	35	n	3%	95.1%	3574	1:05.99	2.66	f
st_e32	36	19	19	n	N/A		failure in quadratic estimation*			+
tls2	38	33	25	y	0	< 1%	112	2.29	0.23	=

LaGO on MINLPs - details: 101 - 1000 variables

example	n	$\|I\|$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
beuster	158	52	115	n	N/A		function not defined over box*			f
cecil_13	841	180	899	n	N/A		1834	58:57.06	1:04.70	+
contvar	297	88	285	n	4%	62.5%	2847	57:32.90	2:27.62	+
csched2	401	308	138	n	19%	-93%	wrong guess on variable bound			+
eniplac	142	24	190	n	0	16.4%	12805	59:51.35	9.03	=
enpro48	154	92	215	n	4%	46.5%	16047	59:57.62	2.64	+
enpro56	128	73	192	n	0	28.2%	14163	59:58.01	2.09	=
fo7_2	115	42	212	y	0	93.1%	16578	59:58.97	1.24	+
fo7	115	42	212	y	N/A		14713	59:59.08	1.23	+
fo8	147	56	274	y	N/A		11110	59:58.35	1.96	+
fo9	183	72	344	y	N/A		7959	59:57.52	2.90	+
gastrans	107	21	150	n	0	$< 1\%$	11	0.50	6.26	=
hda	723	13	719	n	0	90.7%	209	5:41.58	2:25.39	f
johnall	195	190	193	n	0	$< 1\%$	1	0.37	41.10	=
m7	115	42	212	y	0	76.9%	14738	59:58.81	1.25	=
mbtd	211	200	71	n	N/A		1	2.09	64:25.51	f
o7_2	115	42	212	y	N/A		15012	59:58.86	1.25	+
o7	115	42	212	y	N/A		15672	59:58.85	1.24	+
oil2	937	2	927	n	0	$< 1\%$	1	0.54	3:57.11	=
parallel	206	25	116	n	0	100.8%	88	6.25	20.65	=
ravem	113	54	187	n	0	23.7%	15612	32:30.78	1.77	=
risk2b	464	14	581	y	0	$< 1\%$	7	2.72	13.25	=
stockcycle	481	432	98	y	1%	2%	13292	59:55.21	4.80	+

LaGO on MINLPs - details: 101 - 1000 variables

example	n	$\lvert I \rvert$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
water4	196	126	138	n	N/A		15564	59:42.48	17.67	+
waterz	196	126	138	n	N/A		14245	59:42.76	17.31	f
fo7_ar25_1	113	42	270	y	N/A		15240	59:58.74	1.49	+
fo7_ar2_1	113	42	270	y	N/A		15419	59:58.80	1.48	+
fo7_ar3_1	113	42	270	y	N/A		14945	59:58.77	1.51	+
fo7_ar4_1	113	42	270	y	59%	58.3%	14474	59:58.65	1.50	+
fo7_ar5_1	113	42	270	y	N/A		16976	59:58.57	1.53	+
fo8_ar25_1	145	56	348	y	N/A		9132	59:57.88	2.33	+
fo8_ar2_1	145	56	348	y	N/A		10653	59:57.94	2.35	+
fo8_ar3_1	145	56	348	y	N/A		8878	59:57.59	2.41	+
fo8_ar4_1	145	56	348	y	N/A		13195	59:57.59	2.43	+
fo8_ar5_1	145	56	348	y	N/A		11615	59:57.78	2.45	+
fo9_ar25_1	181	72	436	y	N/A		5889	59:56.90	3.57	+
fo9_ar2_1	181	72	436	y	N/A		6439	59:56.46	3.58	+
fo9_ar3_1	181	72	436	y	N/A		6051	59:56.52	3.65	f
fo9_ar4_1	181	72	436	y	N/A		9067	59:56.58	3.68	+
fo9_ar5_1	181	72	436	y	N/A		9719	59:56.40	3.73	f
m7_ar25_1	113	42	270	y	0	$<1\%$	3288	5:54.38	1.46	=
m7_ar2_1	113	42	270	y	0	$<1\%$	18314	40:11.04	1.45	=
m7_ar3_1	113	42	270	y	11%	20.4%	19332	59:58.66	1.50	+
m7_ar4_1	113	42	270	y	0	$<1\%$	9474	30:23.16	1.50	=
m7_ar5_1	113	42	270	y	14%	20.6%	17296	59:58.58	1.53	+

LaGO on MINLPs - details: 101 - 1000 variables

example	n	$\|I\|$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
no7_ar25_1	113	42	270	y	12%	49%	13791	59:58.68	1.52	+
no7_ar2_1	113	42	270	y	N/A		14020	59:58.61	1.49	+
no7_ar3_1	113	42	270	y	N/A		14858	59:58.49	1.53	+
no7_ar4_1	113	42	270	y	N/A		14189	59:58.77	1.53	+
no7_ar5_1	113	42	270	y	51%	67.4%	14168	59:58.69	1.53	+
o7_ar25_1	113	42	270	y	N/A		13292	59:58.66	1.49	+
o7_ar2_1	113	42	270	y	N/A		13459	59:58.52	1.48	+
o7_ar3_1	113	42	270	y	N/A		13735	59:58.56	1.52	+
o7_ar4_1	113	42	270	y	N/A		13954	59:58.47	1.52	+
o7_ar5_1	113	42	270	y	50%	65.9%	14293	59:58.54	1.54	+
o8_ar4_1	145	56	348	y	N/A		11431	59:57.75	2.44	+
o9_ar4_1	181	72	436	y	N/A		7974	59:56.47	3.76	+
tls12	813	668	385	y	N/A		1430	59:10.46	50.29	f
tls4	106	89	65	y	N/A		50000	43:26.17	2.08	+
tls5	162	136	91	y	N/A		14677	59:56.27	3.79	+
tls6	216	179	121	y	N/A		8996	59:54.68	5.64	+
tls7	346	296	155	y	N/A		7489	59:50.29	9.80	f

LaGO vs. BARON on MIQQPs

LaGO and BARON 7.8.1 on MIQQPs from MINLPLib:

		optimal value		
	Total	LaGO better	same	BARON better
BARON fail, LaGO not	3	3		
LaGO faster	13		13	
both solvers the same	21		21	
BARON faster	24		24	
LaGO fail, BARON not	9			9
LaGO and BARON fail	7		7	
Total	77	3	65	9

convex MIQQPs: 22

LaGO on MIQQPs - details: up to 20 variables

example	n	$\|I\|$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
alan	9	4	8	y	0	$< 1\%$	3	0.04	0.02	=
ex1223a	8	4	10	y	0	$< 1\%$	3	0.02	0.03	=
fuel	16	3	16	n	0	$< 1\%$	5	0.06	0.04	=
gbd	5	3	5	y	0	$< 1\%$	1	0.01	0.01	=
st_e13	3	1	3	n	0	$< 1\%$	2	0.02	0.01	=
st_e27	5	2	7	n	0	$< 1\%$	2	0.01	0.02	=
nvs03	3	2	3	y	0	$< 1\%$	3	0.00	0.02	=
nvs10	3	2	3	y	0	$< 1\%$	3	0.01	0.02	=
nvs11	4	3	4	y	0	$< 1\%$	5	0.04	0.02	=
nvs12	5	4	5	y	0	$< 1\%$	3	0.05	0.03	=
nvs13	6	5	6	n	0	$< 1\%$	4	0.07	0.03	=
nvs14	9	5	4	n	0	$< 1\%$	103	1.31	0.03	=
nvs15	4	3	2	y	0	$< 1\%$	3	0.01	0.02	=
nvs17	8	7	8	n	0	$< 1\%$	9	0.27	0.05	=
nvs18	7	6	7	n	0	$< 1\%$	6	0.17	0.04	=
nvs19	9	8	9	n	0	$< 1\%$	15	0.54	0.06	=
nvs23	10	9	10	n	0	$< 1\%$	37	1.24	0.11	=
nvs24	11	10	11	n	0	$< 1\%$	16	1.04	0.14	=
prob02	7	6	9	n	0	$< 1\%$	33	0.07	0.03	=
prob03	3	2	2	n	0	$< 1\%$	6	0.00	0.01	=
st_miqp1	6	5	2	y	0	$< 1\%$	2	0.01	0.02	=
st_miqp2	5	4	4	y	0	$< 1\%$	4	0.01	0.02	=
st_miqp3	3	2	2	y	0	$< 1\%$	3	0.00	0.02	=
st_miqp4	7	3	5	y	0	$< 1\%$	2	0.01	0.02	=

LaGO on MIQQPs - details: up to 20 variables

example	n	$\vert I\vert$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
st_miqp5	8	2	14	y	0	$< 1\%$	2	0.02	0.04	$=$
st_test1	6	5	2	y	0	$< 1\%$	7	0.03	0.02	$=$
st_test2	7	6	3	y	0	$< 1\%$	3	0.00	0.02	$=$
st_test3	14	13	11	y	0	$< 1\%$	4	0.03	0.03	$=$
st_test4	7	6	6	y	0	$< 1\%$	1	0.00	0.02	$=$
st_test5	11	10	12	y	0	$< 1\%$	5	0.01	0.05	$=$
st_test6	11	10	6	y	0	$< 1\%$	4	0.04	0.04	$=$
st_testgr1	11	10	6	y	0	$< 1\%$	3	0.04	0.03	$=$
st_testph4	4	3	11	y	0	$< 1\%$	2	0.00	0.02	$=$
tln2	9	8	13	n	0	$< 1\%$	45	0.13	0.02	$=$

LaGO on MIQQPs - details: 21 - 100 variables

example	n	$\|I\|$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
elf	55	24	39	n	0	$< 1\%$	343	14.74	0.23	=
ex1263	93	72	56	n	0	$< 1\%$	4135	4:43.18	0.26	=
ex1264	89	68	56	n	0	$< 1\%$	2992	2:50.49	0.23	=
ex4	37	25	31	n	0	$< 1\%$	50	3.02	0.25	=
fac3	67	12	34	y	0	$< 1\%$	2	0.11	0.39	=
feedtray2	88	36	284	n	0	$< 1\%$	8	2.22	5.45	=
meanvarx	36	14	45	y	0	$< 1\%$	2	0.03	0.08	=
nous1	51	2	44	n	N/A		9338	59:59.76	0.28	+
nous2	51	2	44	n	0	93.6%	8634	1:00:00.38	0.25	=
sep1	30	2	32	n	0	$< 1\%$	46	1.09	0.10	=
spectra2	70	30	73	n	0	$< 1\%$	83	6.99	0.87	=
du-opt5	21	13	10	y	0	$< 1\%$	71	2.40	0.26	=
du-opt	21	13	10	y	0	$< 1\%$	29864	13:24.68	0.28	=
ex1263a	25	24	36	n	0	$< 1\%$	1317	15.27	0.09	=
ex1264a	25	24	36	n	0	$< 1\%$	4465	1:50.72	0.08	=
ex1265a	36	35	45	n	0	$< 1\%$	819	12.78	0.14	=
ex1266a	49	48	54	n	0	$< 1\%$	178	3.12	0.18	=
st_test8	25	24	21	y	0	$< 1\%$	1	0.00	0.12	=
st_testgr3	21	20	21	y	0	$< 1\%$	21	0.27	0.09	=
tln4	25	24	25	n	N/A		19920	1:00:00.17	0.05	+
tln5	36	35	31	n	N/A		11278	1:00:00.41	0.08	+
tln6	49	48	37	n	N/A		9904	1:00:00.15	0.13	+
tln7	64	63	43	n	N/A		9909	1:00:00.12	0.19	+
tloss	49	48	54	n	0	$< 1\%$	817	16.58	0.28	=
tltr	49	48	55	n	0	$< 1\%$	2031	1:06.99	0.21	=

LaGO on MIQQPs - details: 101 - 1000 variables

example	n	$\|I\|$	m	c	error	gap	iter	Time: B&B	Prepr.	BARON
ex1265	131	100	75	n	0	$< 1\%$	2148	3:19.32	0.43	=
ex1266	181	138	96	n	0	$< 1\%$	424	40.79	0.68	=
nuclear14a	993	600	634	n	0	100%	468	54:48.07	5:20.13	f
nuclear24a	993	600	634	n	0	100%	468	54:50.97	5:20.10	f
nuclearva	352	168	318	n	N/A		2488	57:28.17	2:33.59	f
nuclearvb	352	168	318	n	N/A		2433	59:03.20	57.54	f
nuclearvc	352	168	318	n	N/A		2243	59:35.17	26.92	f
nuclearvd	352	168	318	n	N/A		1974	59:31.55	31.61	f
nuclearve	352	168	318	n	N/A		2224	59:28.36	31.66	f
nuclearvf	352	168	318	n	N/A		1932	59:35.33	24.91	f
qap	226	225	31	n	12%	100%	10478	58:47.89	1:12.15	f
qapw	451	225	256	n	N/A		382	57:37.84	2:28.85	+
space25	894	750	236	n	N/A		6413	59:41.49	18.69	+
space25a	384	240	202	n	N/A		11559	59:51.93	8.09	+
st_e31	113	24	136	n	0	6.4%	18406	59:59.63	0.36	=
util	146	28	168	n	0	$< 1\%$	747	1:27.59	1.02	=
lop97icx	987	899	88	n	N/A		1523	59:33.58	28.18	+
tln12	169	168	73	n	N/A		7073	59:58.87	1.15	f

LaGO vs. BARON on QQPs

running LaGO and BARON 7.8.1 on QQPs from GlobalLib:

	Total	optimal value: LaGO better	optimal value: same	optimal value: BARON better
BARON fail, LaGO not				
LaGO faster	10		10	
both solvers the same	86		86	
BARON faster	65	1	64	
LaGO fail, BARON not	1			1
Total	162	1	160	1

convex QQPs: 11

Overview

Ingredients of LaGOs Branch and Cut Algorithm

- Preprocessing
- Underestimators and Cutting Planes
- Boxreduction

Numerical Experiments

Future Developments

Next Steps

Improve current techniques ...

- generation of quadratic underestimators during Branch and Bound
- improve robustness
- ...

Next Steps

Improve current techniques ...

- generation of quadratic underestimators during Branch and Bound
- improve robustness
- ...

and introduce new ...

- Directed Acyclic Graph (DAG) Representation ⇒ constraint propagation (cf. Branch and Infer, COCONUT Solver)
- mixed-integer linear relaxation + interval gradient cuts ⇒ better approximation of nonconvexities
- ...

LaGO + Bonmin

COIN-OR / LaGO

Strengths:

- handling of nonconvex black-box functions
- boxreduction

Weakenings:

- Branch and Bound algorithm (branching rule, node selection, ...)

COIN-OR / Bonmin

Strengths:

- Mixed-Integer linear relaxation
- Branch and Bound algorithm (use of MIP solver Cbc)

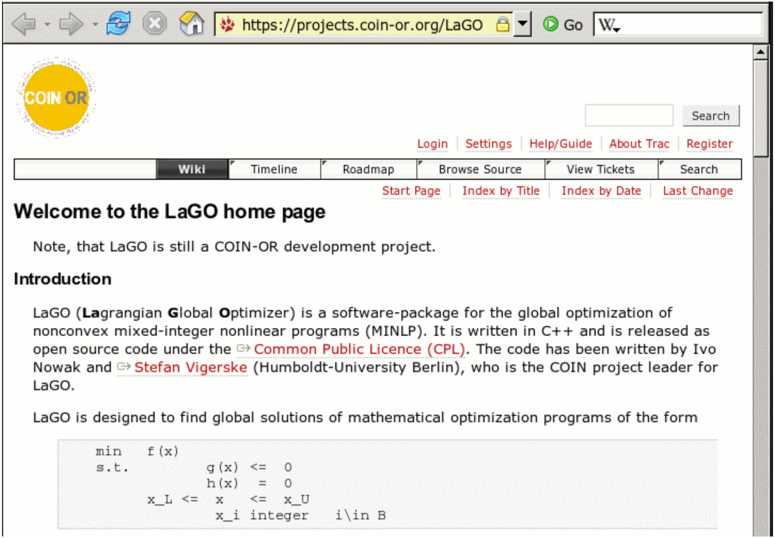
Weakenings:

- handling of nonconvexities (but in preparation for algebraic formulations)

$\Rightarrow$ integrate LaGOs convexification and boxreduction techniques into Bonmin

Thank you!

PS: Your contribution is welcome at


https://projects.coin-or.org/LaGO

COIN OR

Search

Login | Settings | Help/Guide | About Trac | Register

Wiki | Timeline | Roadmap | Browse Source | View Tickets | Search

Start Page | Index by Title | Index by Date | Last Change

Welcome to the LaGO home page

Note, that LaGO is still a COIN-OR development project.

Introduction

LaGO (**La**grangian **G**lobal **O**ptimizer) is a software-package for the global optimization of nonconvex mixed-integer nonlinear programs (MINLP). It is written in C++ and is released as open source code under the Common Public Licence (CPL). The code has been written by Ivo Nowak and Stefan Vigerske (Humboldt-University Berlin), who is the COIN project leader for LaGO.

LaGO is designed to find global solutions of mathematical optimization programs of the form

```
min    f(x)
s.t.          g(x)  <=   0
              h(x)   =   0
       x_L <=  x    <=   x_U
               x_i integer    i\in B
```

nvs05

$$\min \quad 1.10471 x_3^2 x_4 + 0.04811 y_1 y_2 (14 + x_4)$$

$$\text{s.t.} \quad -\frac{4243.28147100424}{x_3 x_4} + x_5 = 0$$

$$-\sqrt{\frac{1}{4} x_4 + \left(\frac{1}{2} y_1 + \frac{1}{2} x_3\right)^2} + x_7 = 0 \qquad -\frac{1}{2}\frac{x_4}{x_7} + x_8 = 0$$

$$-\frac{0.71\,(84000 + 3000 x_4)\, x_7}{x_3 x_4 \left(0.08 x_4^2 + \left(\frac{1}{2} y_1 + \frac{1}{2} x_3\right)^2\right)} + x_6 = 0 \qquad \frac{0.21952}{y_1^3 y_2} \leq \frac{1}{4}$$

$$\sqrt{x_5^2 + 2 x_5 x_6 x_8 + x_6^2} \leq 13600 \qquad \frac{504000}{y_1^2 y_2} \leq 30000$$

$$0.02 \sqrt{10^{15} y_1^2 y_2^6}\,(1 - 0.03 y_1) \geq 6000 \qquad y_2 - y_3 \geq 0$$

$$y_1, y_2 \in \{0, \ldots, 200\} \qquad x_3, x_4 \in [0.01, 200]$$

st_e35

Problem to determine convexity (due to Hessian evaluation errors) of

$$\sum_{i=0}^{3} 670 \left(\frac{x_{17+i}}{\frac{1}{2}\sqrt[3]{x_{8+i}^2 x_{9+i} + x_{8+i} x_{9+i}^2}} \right)^{0.83} + \sum_{i=0}^{2} \left(\frac{x_{21+i}}{\frac{1}{2}\sqrt[3]{C_i x_{14+i}^2}} \right)^{0.83}$$

with $x_8, \ldots, x_{16} \in [0.01, 1000]$

Return

csched1

Problem to determine bounds on variable z in

$$\begin{aligned}
\min \quad & z \\
\text{s.t.} \quad x_{13} z = \; & 416000 x_4 \left(1 - \exp\left(-\frac{x_1}{10 x_4}\right)\right) + 37400 x_1 - 100 x_4 \\
& + 124615 x_5 \left(1 - \exp\left(-\frac{13 x_2}{100 x_5}\right)\right) + 9000 x_2 - 90 x_5 \\
& + 278666.7 x_6 \left(1 - \exp\left(-\frac{9 x_3}{100 x_6}\right)\right) + 15840 x_3 - 80 x_6 \\
& \vdots
\end{aligned}$$

with $x_1, x_2, x_3, x_{13} \geq 0, x_4, x_5, x_6 \geq \frac{1}{10}$.

Return

fac1

Objective function

$$50\left(x_1+x_2+x_3+x_4+x_9+x_{10}+x_{11}+x_{12}\right)^{2.5} \\ +70\left(x_5+x_6+x_7+x_8+x_{13}+x_{15}+x_{16}\right)^{2.5} \\ +\text{linear term}$$

with $x_1,\ldots,x_{16} \geq 300$.

Return

st_e32

CPLEX reports infeasibility for LP which computes quadratic underestimator of

$$\exp\left(-\frac{x}{z}\left(1 + y\,(3+z)\,(1-z)^3 - z^2\right)\right)$$

with $x \in [0.001, 10]$, $y \in [-10, 10]$, $z \in [0.01, 1]$.

Return

beuster

Error in the evaluation of

$$0.0028x_{40} \log\left(\frac{x_{44} - x_{28}}{x_{20} - x_{28}}\right) - 10^7 x_{56}$$

for $x_{20} \geq 0$, $x_{28} \geq 0.001$, $x_{44} \geq 0.01$.

Return