

LaGO

Branch and Cut for nonconvex block-separable MINLPs in the
absence of algebraic formulations

Stefan Vigerske

Humboldt-University Berlin, Department of Mathematics



GICOLAG, ESI, 7.12.2006

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 MIQQP 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 0.2 (**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

Preprocessing

Underestimators and Cuts

Boxreduction

Numerical Experiments

Further developments

MINLP

We consider problems of the form

$$\begin{aligned}
 & \text{minimize} && c^T x \\
 & \text{such that} && h_i(x) \leq 0, && i \in I, \\
 & && h_j(x) = 0, && j \in E, \\
 & && x_k \in \{0, 1\}, && k \in B, \\
 & && x \in [\underline{x}, \bar{x}]
 \end{aligned}$$

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

- $h \in C^2([\underline{x}, \bar{x}], \mathbb{R}^{|I|+|E|})$ 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

Preprocessing

Underestimators and Cuts

Boxreduction

Numerical Experiments

Further 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**

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

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

Preprocessing

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

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

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

- **curvature** (evaluating 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**

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

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

- **curvature** (evaluating eigenvalues of Hessian in sample points)
- **Reduction of box** $[\underline{x}, \bar{x}]$:
 - based on interval arithmetic
 - enclosing the set defined by linear constraints
 - bounding box for still unbounded variables by “guessing”

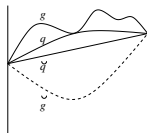
Preprocessing

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

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

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

- curvature** (evaluating eigenvalues of Hessian in sample points)
- Reduction of box** $[\underline{x}, \bar{x}]$:
 - based on interval arithmetic
 - enclosing the set defined by linear constraints
 - bounding box for still unbounded variables by “guessing”
- Initialization of **relaxations**:
 - quadratic (nonconvex) underestimator q
 - quadratic convex underestimator \check{q}
 - linearization, dropping of binary conditions



Overview

Preprocessing

Underestimators and Cuts

Boxreduction

Numerical Experiments

Further developments

Nonconvex 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

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

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

Nonconvex 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

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

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

Nonconvex 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

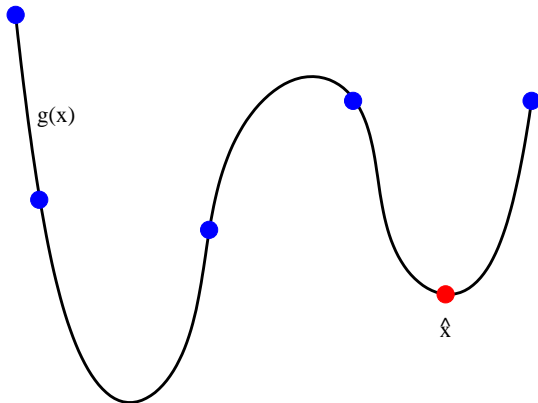
$$\begin{aligned} \min_{A, b, c} \quad & \sum_{x \in S} g(x) - q(x) \\ \text{such that} \quad & q(x) \leq g(x) \quad x \in S \\ & q(\hat{x}) = g(\hat{x}) \end{aligned}$$

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**
- ⇒ A. Neumaier 2006: adaptive choice of S

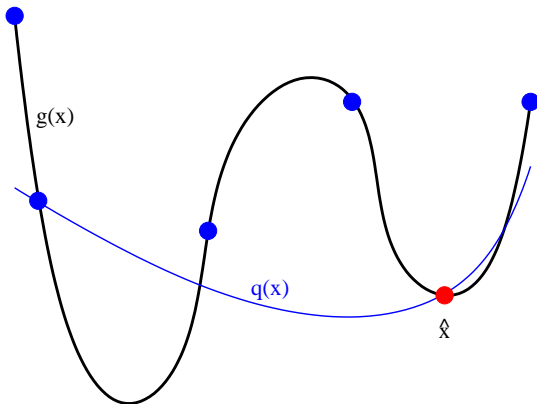
Nonconvex quadratic underestimator (cont.)

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



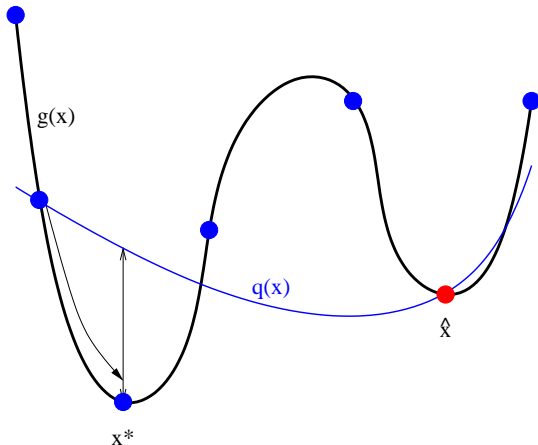
Nonconvex quadratic underestimator (cont.)

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



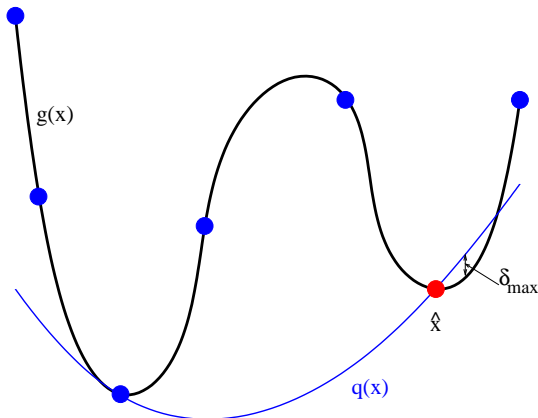
Nonconvex quadratic underestimator (cont.)

- 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 $q(x)$

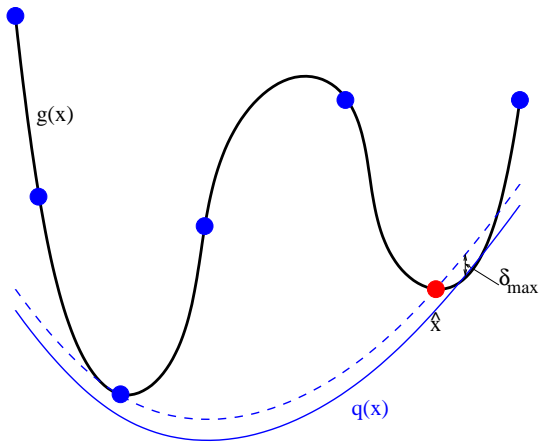


Nonconvex quadratic underestimator (cont.)

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



Nonconvex quadratic underestimator (cont.)



- currently only one underestimator per function and no update in Branch and Cut

Convex underestimators

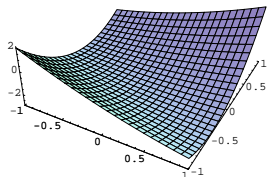
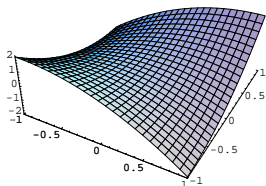
Let $q(x) = x^T A x + b^T x + c$ be a quadratic nonconvex function.

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

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

where

$$\alpha_i = -\lambda_1(\text{Diag}(\bar{x} - \underline{x}) A \text{Diag}(\bar{x} - \underline{x})) (\bar{x}_i - \underline{x}_i)^{-2}.$$



- Linearizations of $\check{q}(x)$ 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

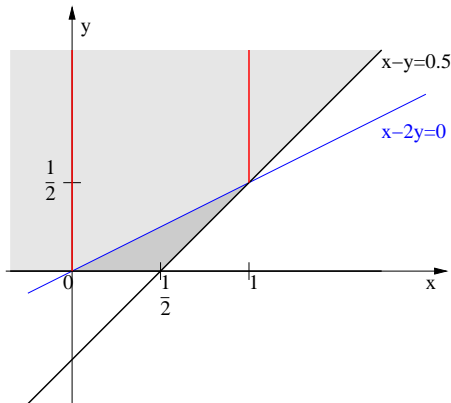
Mixed Integer Rounding Cut

(Nemhauser, Wolsey 1988) principle:

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

$$x - \frac{1}{1 - (b - \lfloor b \rfloor)} y \leq \lfloor b \rfloor$$

$$\forall (x, y) \in X$$



Overview

Preprocessing

Underestimators and Cuts

Boxreduction

Numerical Experiments

Further developments

Boxreduction based on intervalarithmetic

- Consider a constraint

$$g(x, y) := h(x) + y \leq 0,$$

i.e., $y \leq -h(x)$, and let

$$[\underline{h}, \bar{h}] := -h([\underline{x}, \bar{x}]).$$

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

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

and proceed with other constraints depending on y .

Boxreduction based on intervalarithmetic

- Consider a constraint

$$g(x, y) := h(x) + y \leq 0,$$

i.e., $y \leq -h(x)$, and let

$$[\underline{h}, \bar{h}] := -h([\underline{x}, \bar{x}]).$$

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

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

and proceed with other constraints depending on y .

- 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.

$$\underline{x}_j := \min x_j$$

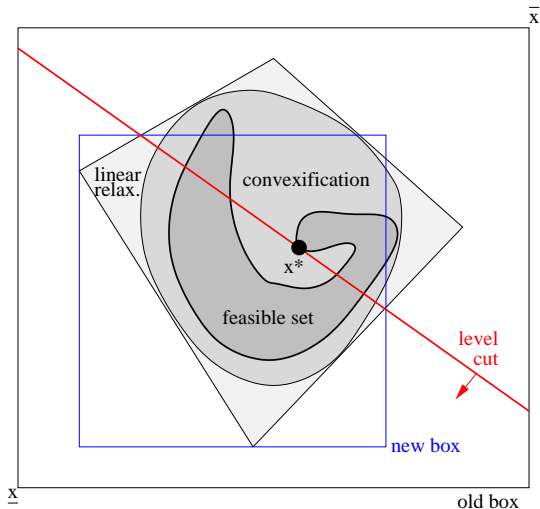
$$\text{s.t. } Ax \leq b$$

$$c^T x \leq c^T x^*$$

$$\bar{x}_j := \max x_j$$

$$\text{s.t. } Ax \leq b$$

$$c^T x \leq c^T x^*$$



Overview

Preprocessing

Underestimators and Cuts

Boxreduction

Numerical Experiments

Further developments

GAMS MINLPLib and GlobalLib

- at most 1000 variables and no integrality conditions except for binary
 ⇒ 33 MIQQP, 72 (nonquadratic) MINLP, 163 QQP
- timelimit: 1 hour
- NLP subsolver: CONOPT; LP subsolver: CPLEX 10.0

	MIQQP	MINLP
number of models	33	72
best known optimal solution found	21	42
local optimal solution found	4	10
no feasible point found	8	20

Pentium IV 3.00 Ghz, 1 GB RAM, Linux 2.16.11

LaGO vs. BARON on MIQPPs

LaGO and BARON 7.5 on MIQPPs from MINLPLib:

	Total	optimal value		
		LaGO better	same	BARON better
BARON fail, LaGO not	2	2		
LaGO faster	1		1	
both solvers the same	7		5	2
BARON faster	15	2	11	2
LaGO fail, BARON not	2			2
LaGO and BARON fail	6		6	
Total	33	4	23	6

convex MIQPPs: 5

LaGO vs. BARON on MINLPs

LaGO and BARON 7.5 on (nonquadratic) MINLPs from MINLPLib:

	Total	optimal value		
		LaGO better	same	BARON better
BARON fail, LaGO not	4	4		
LaGO faster	6	1	2	3
both solvers the same	7	1	5	1
BARON faster	35	4	25	6
LaGO fail, BARON not	8			8
LaGO and BARON fail	12		12	
Total	72	10	44	18

convex MINLPs: 20

LaGO stops branching when all binary variables are fixed (due to missing update of quadratic underestimators)

LaGO vs. BARON on QQP_s

running LaGO and BARON 7.5 on QQP_s from GlobalLib:

	Total	optimal value		
		LaGO better	same	BARON better
BARON fail, LaGO not	1	1		
LaGO faster	13	1	12	
both solvers the same	82		81	1
BARON faster	65		65	
LaGO fail, BARON not	2			2
Total	163	2	158	3

convex QQP_s: 12

Model of a complex energy conversion plant

- superstructure for electric power output of ≤ 400 MW and process steam production of ≤ 500 t/h
- degrees of freedom: 27 structural and 48 process variables

Model of a complex energy conversion plant

- superstructure for **electric power** output of ≤ 400 MW and **process steam** production of ≤ 500 t/h
- degrees of freedom: **27 structural** and **48 process** variables
- **constraints**:
 - **logic** of the superstructure (connecting binary variables)
 - **thermodynamic** behavior (highly nonlinear), mass+energy balances
 - purchase equipment **costs**
- **objective**: total cost for cogeneration plant
investment cost, operation and maintenance cost, taxes and insurances,...

Model of a complex energy conversion plant

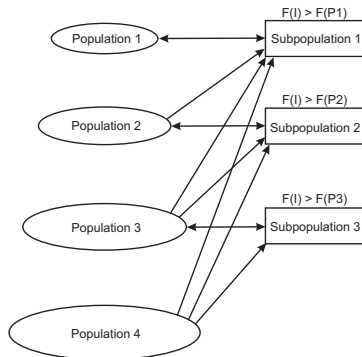
- superstructure for **electric power** output of ≤ 400 MW and **process steam** production of ≤ 500 t/h
- degrees of freedom: **27 structural** and **48 process** variables
- **constraints**:
 - **logic** of the superstructure (connecting binary variables)
 - **thermodynamic** behavior (highly nonlinear), mass+energy balances
 - purchase equipment **costs**
- **objective**: total cost for cogeneration plant
investment cost, operation and maintenance cost, taxes and insurances,...
- **MINLP model**:
 - 1308 variables (44 binary, 3%)
 - 1659 constraints (376 nonlinear, 23%)
 - maximal block size: 6
 - quadratic underestimators computed: 329
- GAMS MINLPLib models super1, super2, super3, and super3t

Distributed genetic algorithm

- individual = set of decision variables
- fitness obtained by **simulation** of the superstructure

HSC-GA: hierarchical social competition algorithm:

- handling **several populations** in parallel
- organized by fitness of inhabitants
- individuals from lower population can **move into subpopulation** at higher level
- after evolving for some time, they **migrate into higher population**



Optimization of the superstructure

- HSC-GA and LaGO run for 24 hours
 - HSC-GA: ≈ 20000 generations
 - LaGO: ≈ 30000 Branch and Bound iterations

demand	method	efficiency	cost
electric power: 300 MW	LaGO	56.7%	12674 Euro/h
	HSC-GA	55.4%	12774 Euro/h
electric power: 290 MW process steam: 150 t/h	LaGO	68.5%	13424 Euro/h
	HSC-GA	67.7%	13399 Euro/h
electric power: 400 MW	LaGO	58.6%	16771 Euro/h
	HSC-GA	58.7%	17229 Euro/h

Turang Ahadi-Oskui (2006): Optimierung des Entwurfs komplexer Energieumwandlungsanlagen, Fortschritts-Berichte VDI, Reihe 6, Nr. 543.

Overview

Preprocessing

Underestimators and Cuts

Boxreduction

Numerical Experiments

Further developments

Intervalgradient Cuts

- Replace linear relaxation by a **mixed-integer** linear relaxation.

Allows use of **intervalgradient cuts** (Boddy and Johnson 2003 for MIQQP_s):

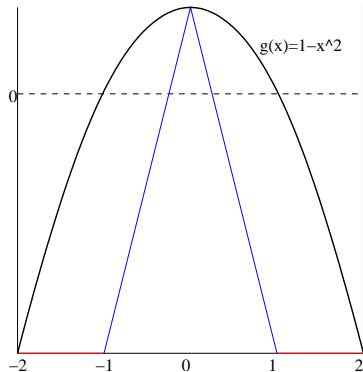
Intervalgradient of $g(x)$:

$$[\underline{d}, \bar{d}] := \nabla g([\underline{x}, \bar{x}])$$

Intervalgradient cut w.r.t. $\hat{x} \in [\underline{x}, \bar{x}]$:

$$g(\hat{x}) + \min_{d \in [\underline{d}, \bar{d}]} d^T (x - \hat{x}) \leq 0$$

- better approximation of nonconvexities
- local exact \Rightarrow finite convergence
- independent of relaxations



Next steps...

many things to improve, e.g.,

- support of integer variables
- branching rules
- node selection rules
- generation of quadratic underestimators during Branch and Bound
- ...

your contribution is welcome at

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

Next steps...

many things to improve, e.g.,

- support of integer variables
- branching rules
- node selection rules
- generation of quadratic underestimators during Branch and Bound
- ...

your contribution is welcome at

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

Thank you!