Automatic Reformulation Techniques For Mixed-Integer Nonlinear Programs

Ashutosh Mahajan, Meenarli Sharma

Indian Institute of Technology Bombay

Overview

We present

- automatic detection of two reformulation techniques exploiting specific structures involving (1) a group of constraints (2) a single nonlinear constraint, in convex MINLP
- a few applications where such structures appear
- computational results showing impact of these reformulations when applied individually and in combination

Perspective Reformulation (PR)

"Replace a subset of feasible region by its convex hull to tighten the relaxation of the entire feasible region."

Structures S_1 and S_2 defining some such subsets [1]:

- Binary variable z controls all variables in a constraint

$$S_1 = \left\{ (x,z) \in \mathbb{R}^n \times \{0,1\} \;\middle|\; \begin{array}{l} (x,z) \in \Gamma_0, \text{ if } z=0 \\ (x,z) \in \Gamma_1, \text{ if } z=1 \end{array} \right\}$$

where, $\Gamma_0 = \{(x,z) \in \mathbb{R}^n \times \{0,1\} : x = \hat{x}, z = 0\}$, $\Gamma_1 = \{(x,z) \in \mathbb{R}^n \times \{0,1\} : f(x) + cz \le b, zl \le x \le uz, z = 1\}$, $l, u \in \mathbb{R}^n$, $b \ge f(\hat{x})$, and $\hat{x}$ is known.

Lemma: If Γ_1 convex, then $\text{conv}(S_1) = \Gamma_0 \cup \tilde{S}$ where

$$\tilde{S} = \Big\{ (x,z) \in \mathbb{R}^{n+1} : f\Big(\frac{x-(1-z)\hat{x}}{z}\Big) + c \le b, \quad lz + (1-z)\hat{x} \le x \le uz + (1-z)\hat{x}, \quad 0 < z \le 1 \Big\}.$$

Γ_1 is convex if $f(x)$ is convex and for $z > 0$

$$f\Big(\frac{x-(1-z)\hat{x}}{z}, 1\Big) \le b \Leftrightarrow zf\Big(\frac{x-(1-z)\hat{x}}{z}, 1\Big) \le zb$$

- z controls all variables only in nonlinear part of constraint

$$S_2 = \left\{ (x,z) \in \mathbb{R}^n \times \{0,1\} \;\middle|\; \begin{array}{l} (x,z) \in \Gamma_0, \text{ if } z=0 \\ (x,z) \in \Gamma_1, \text{ if } z=1 \end{array} \right\}$$

where, $\Gamma_0 = \{(x,z) \in \mathbb{R}^n \times \{0,1\} : x^1 = \hat{x}^1, f_N(x^1) + f_L(x^2) \le b, z = 0\}$, $\Gamma_1 = \{(x,z) \in \mathbb{R}^n \times \{0,1\} : f(x) + cz \le b, lz \le x^1 \le uz, z = 1\}$, and $b \le f_N(x^1)$. $f_N(x^1)$ and $f_L(x^2)$ are nonlinear and linear part, respectively, of f with no variable in common.

Perspective Cut (PC)[2]: PR can be solved using PC which is outer-approximation cut to a constraint amenable to PR. Given a nonlinear constraint

$$f(x) + cz \le b$$

amenable to S_1 or S_2, the reformulated constraint is:

$$h(x,z) \le 0 \qquad (1)$$

where, $h(x,z) = zf\Big(\frac{x-(1-z)\hat{x}}{z}\Big) + cz - zb$.

Outer-approximation cut to (1) at a point $(\tilde{x}, \tilde{z})$ is

$$h(\tilde{x},\tilde{z}) + [s_1, s_2]([x,z] - [\tilde{x},\tilde{z}])^T \le 0, \quad \forall [s_1,s_2] \in \partial h(\tilde{x},\tilde{z}) \qquad (2)$$

Note $\partial h(x,z)$ is constant on the segments of the form $x = (1-z)\hat{x} + z\overline{x}$ with $z \in [0,1]$ for any fixed $\overline{x} \in \Gamma_1$.

Determining Structures S_1 and S_2 in Convex MINLP

- Determining S_1 and S_2 is NP-Hard.
- Given a convex MINLP, we look for collection of constraints that are sufficient to identify S_1 and S_2.
  1. Constraints (3)-(5) indicates S_1

$$f(x) + cz \le b, \qquad (3)$$
$$lz + \hat{x} \le x \le uz + \hat{x} \qquad (4)$$
$$z \in \{0,1\}. \qquad (5)$$

  where, $b \ge f(\hat{x})$, and $l, u, \hat{x} \in \mathbb{R}^n$.

  2. Constraints (6)-(8) indicates S_2

$$f_N(x^1) + f_L(x^2) + cz \le b, \qquad (6)$$
$$lz + \hat{x}^1 \le x^1 \le uz + \hat{x}^1, \qquad (7)$$
$$z \in \{0,1\}. \qquad (8)$$

  where, $l, u, \hat{x} \in \mathbb{R}^{|N^1|}$, and $b \le f_N(x^1)$.

We **automated in MINOTAUR**

1. identification of constraints amenable to PR
2. generation of PC in a branch-and-cut framework, LP/NLP based branch-and-bound algorithm (qg)

MINOTAUR: **M**ixed **I**nteger **N**onlinear **O**ptimization **T**oolkit - **A**lgorithms,**U**nderestimators, **R**elaxations, an **open-source** framework for solving Mixed-Integer Nonlinear Programs.

`https://github.com/minotaur-solver/minotaur`

Reformulation Using Function Separability

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called **'group separable'** if there exist $f^i : \mathbb{R}^{n_i} \to \mathbb{R}, \; i = 1, \ldots, m$ such that

$$f(x) = \sum_{i=1}^{m} f^i(x^i)$$

and f^i and $f^j, j \ne i$ do not have any variable in common. Fully and partially separable functions are special cases.

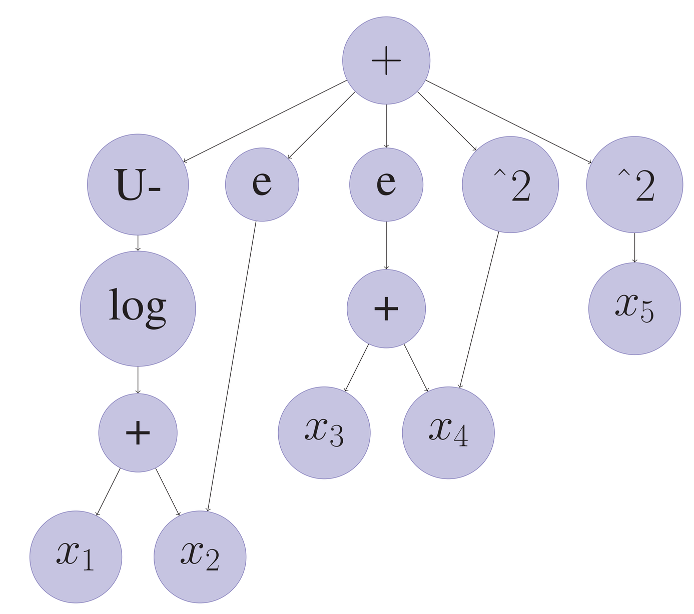
Why exploit function separability?

- Generates tighter relaxations in branch-and-cut [3].
- Improves computational efficiency in numerical differentiation (e.g. calculating Jacobian matrix of f at any x).
- Aids other reformulation techniques like PR.

Separability Detecting Using Computational Graph

- We **automated** reformulation based on group separable functions using 'computational graph' in MINOTAUR.
- Computational graph is a representation of a nonlinear function as a directed graph for computational purposes.

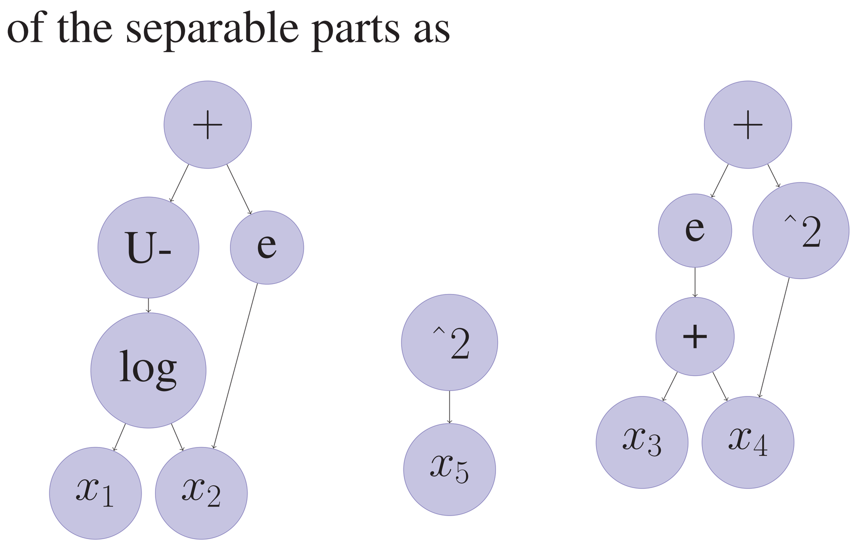
Given $f(x) = -log(x_1 + x_2) + e^{x_2} + e^{(x_3+x_4)} + {x_4}^2 + x_5^2$, computational graph can be



Reformulation: $f(x) \le b$ is automatically reformulated as

$$v_1 + v_1 + v_3 \le b$$
$$-log(x_1 + x_2) + e^{x_2} \le v_1$$
$$e^{(x_3+x_4)} + {x_4}^2 \le v_2$$
$${x_5}^2 \le v_3$$

computational graph of f is replaced by the computational graphs of the separable parts as



Applications and Test Problems

Applications of PR: Production planning, separable quadratic uncapacitated facility location, network design with congestion constraints, etc.

Problems with separable functions: Layout problems, portfolio optimization, facility location, unit commitment, water network contamination

Test instances from **MINLPLib2** [4]

- 48 **MBNLP** synthesis instances with 9-84 PR constraints
- 114 instances (MBNLP, **MBQCP**, **MBQP**, MINLP) with either separable constraints (24) or objective (74) or both (17) (ranging 1-60 in numbers)

Algorithms: `qg`, `qg` with PR (`qgpr`), `qg` with separability exploitation (`qgsep`), `qgep` with PR(`qgseppr`)

Summary of Results

- Both the reformulations improve root relaxation value and reduce no. of nodes processed upto 97%
- `qgsep` reduces solving time in the range 7%-99% and solved 9 that qg could not
- All squfl- and unitcommit1 instances become PR amenable after exploiting separability
- All squfl- instances that reached time limit in `qg` and `qgsep` get solved by `qgseppr`
- Solving time and no. of nodes processed reduce **drastically** in all the instances using `qgseppr`

Computational Results

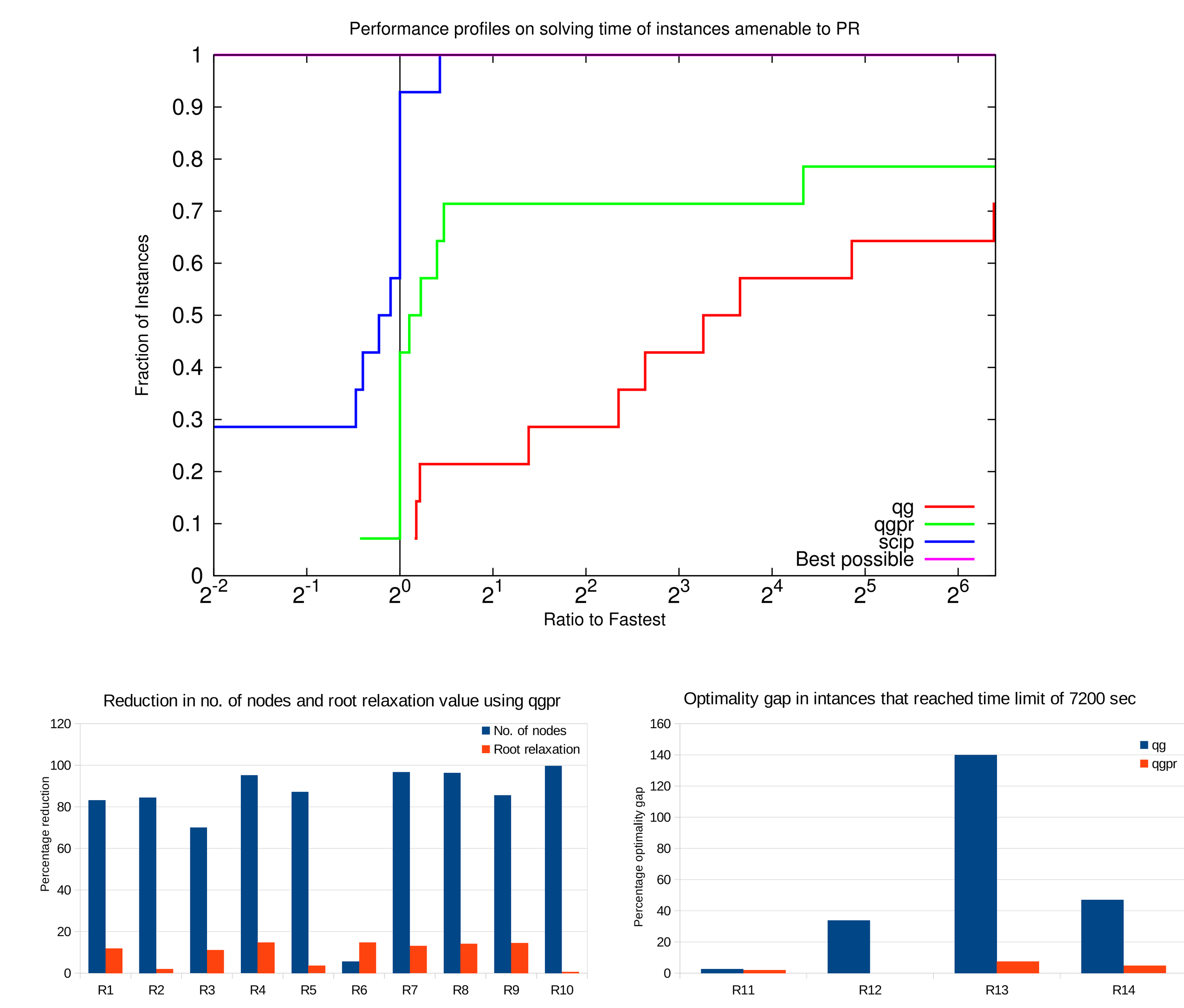


Figure 1 : Impact of perspective reformulation

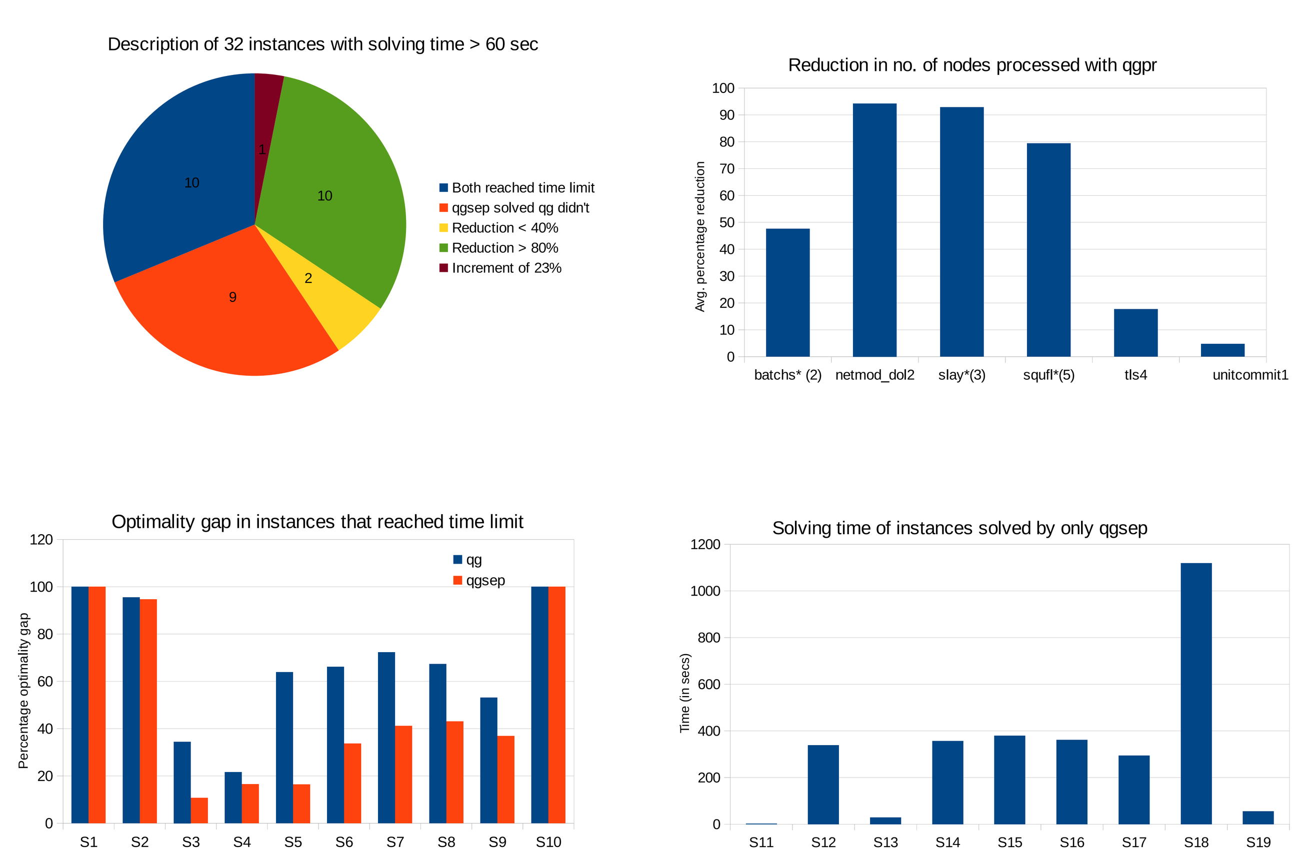


Figure 2 : Impact of separability exploitation

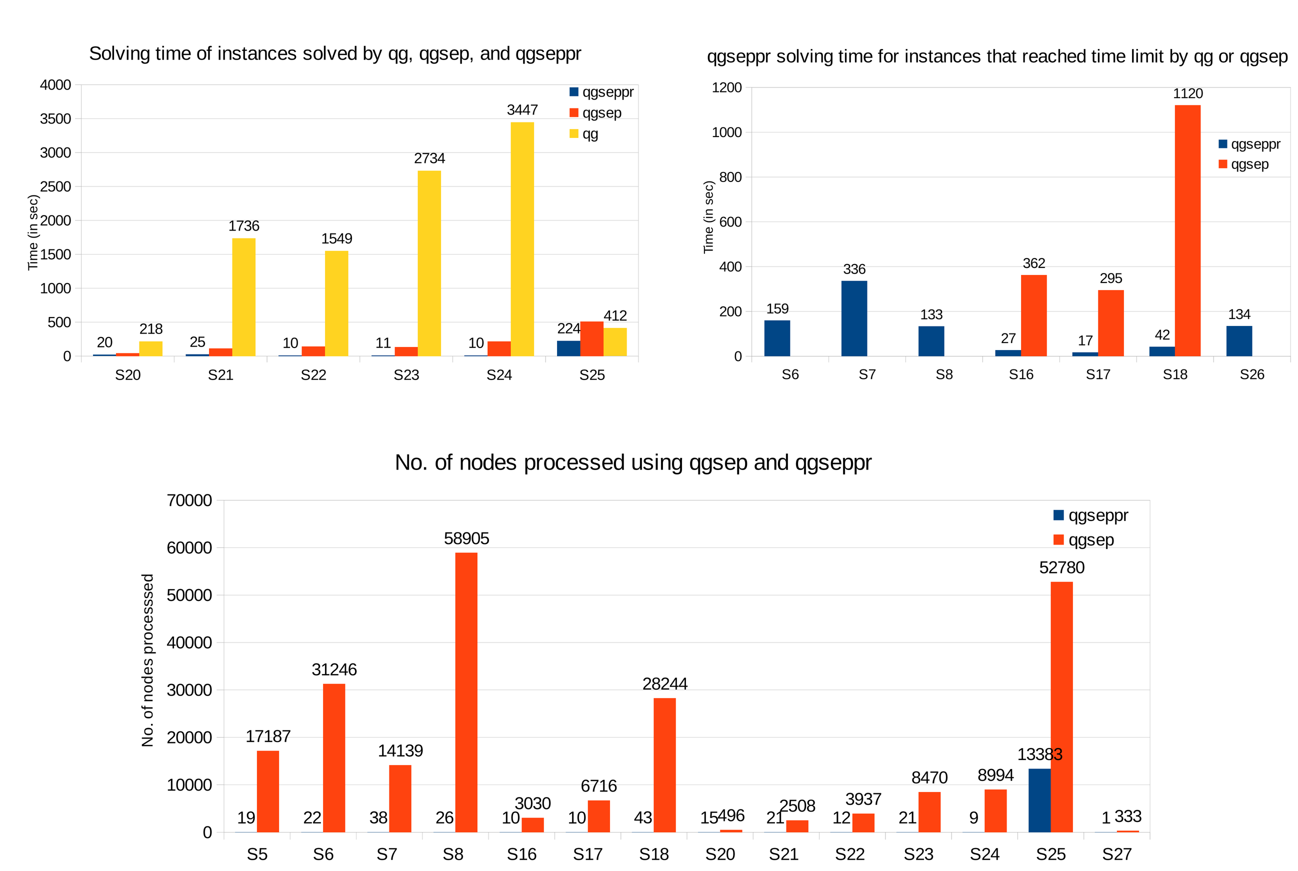


Figure 3 : Impact of separability inducing PR

Aliases: rsyn0805m-03m,04m(R1,2), 0810m02m,03m,04m(R3,4,5), 0815m03m,04m (R6,11), 0820m03m,04m(R7,12), 0830m03m,04m(R8,13), 0840m02m,03m,04m(R9,10,14), ball_mk4_05,k3_20(S1,11), gams01(S2), netmod_dol1(S3), portfol_classical200_2,050_1(S4,13), squfl020-150,030-100,030-150,040-080,015-080,020-050,025-040,010-080,015-060,020-040,025-025,025-030,010-040(S5-8,16-18,20-27), tls5,6(S9-10), ibs2(S12), slay10h,10m(S14-15), stockcycle(S19), unitcommit1(S25)

Planned Work

- Identification of more structures amenable to PR
- Identification of other forms of separability in the problem
- Implementation of cut-manager to generate and manage cuts in branch-and-cut framework

References

[1] O. Günlük and J. Linderoth, "Perspective reformulations of mixed integer nonlinear programs with indicator variables," *Mathematical programming*, vol. 124, no. 1-2.

[2] A. Frangioni and C. Gentile, "Perspective cuts for a class of convex 0–1 mixed integer programs," *Mathematical Programming*, vol. 106, no. 2, pp. 225–236, 2006.

[3] H. Hijazi, P. Bonami, and A. Ouorou, "An outer-inner approximation for separable mixed-integer nonlinear programs," *INFORMS Journal on Computing*, vol. 26, no. 1, pp. 31–44, 2013.

[4] `www.gamsworld.org/minlp/minlplib2/html/`.

Mail: `meenarli@iitb.ac.in`