Mixed-Integer Constrained Grey-Box Optimization based on Dynamic Surrogate Models and Approximated Interval Analysis

ICORES 2021

Mohamad Omar Nachawati — mnachawa@gmail.com
with Professor Alexander Brodsky (my PhD dissertation director)

George Mason University

February 5th, 2021

0 Outline

1 Outline

1 Mixed-Integer Constrained Grey-Box Optimization (MICGB)

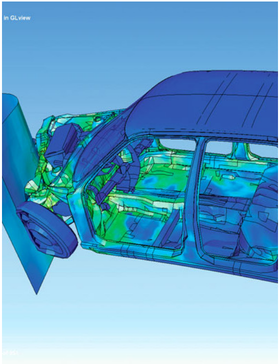
Optimization of simulations over general constrained mixed-integer sets, where simulations are expressed as a grey-box, i.e. computations using a mix of

1. closed-form analytical expressions
2. evaluations of numerical black-box functions that may be
   - non-differentiable
   - computationally expensive

1 MICGB Optimization Applications

Wide-range of real-world applications across diverse commercial and industrial domains:

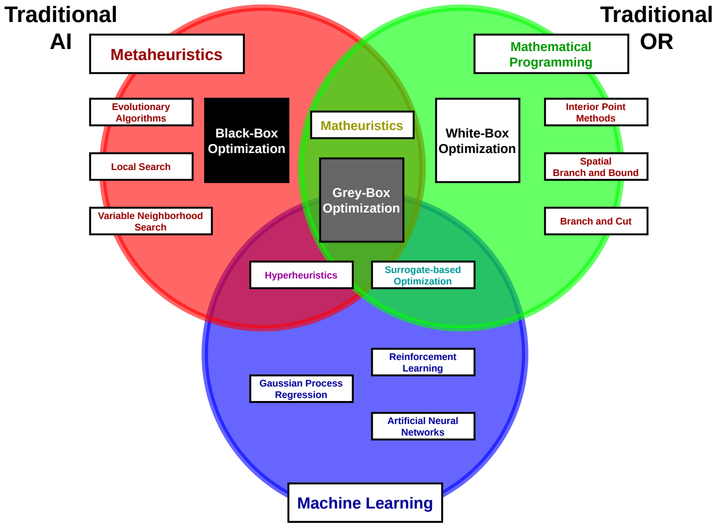
Decision Guidance for Logistics



Decision Guidance for Manufacturing

Decision Guidance for Supply Chains

1 Research Problem: Efficiency versus Versatility



1 Key Contributions

- GreyOpt algorithmic framework that leverages the partially analytical structure for MICGB optimization by:
  - dynamically constructing surrogates for embedded black-box functions in multiple regions of the search space
    - derivative-based solvers on the surrogates for local improvement
  - recursively partitioning regions to refine the best points found
    - extends Moore interval arithmetic (Moore, 1966) with quadric under/over estimators for approximating the intervals of grey-box objective and constraint functions
- Experimental study of GreyOpt's performance on a set of 25 MICGB optimization problems
  - significantly outperforms three derivative-free optimization algorithms
  - significantly outperforms BONMIN with random restart (even for problems with cheap black-box functions)

GEORGE MASON UNIVERSITY

$$
\begin{aligned}
\underset{x\in\mathbb{R}^n,\ y\in\mathbb{R}^m}{\text{minimize}} \quad & f(x,y) && \text{(1a)}\\
\text{subject to} \quad & g_L \le g(x,y) \le g_U && \text{(1b)}\\
& x_L \le x \le x_U && \text{(1c)}\\
& y_L \le y \le y_U && \text{(1d)}\\
& y \in \mathbb{Z}^m && \text{(1e)}
\end{aligned}
$$

where

- $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the objective function
- $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$ is the vector-valued function of constraints
- $x \in \mathbb{R}^n$ are the real decision variables
- $y \in \mathbb{R}^m$ are the integer decision variables
- $n, m, q \in \mathbb{N}_0$

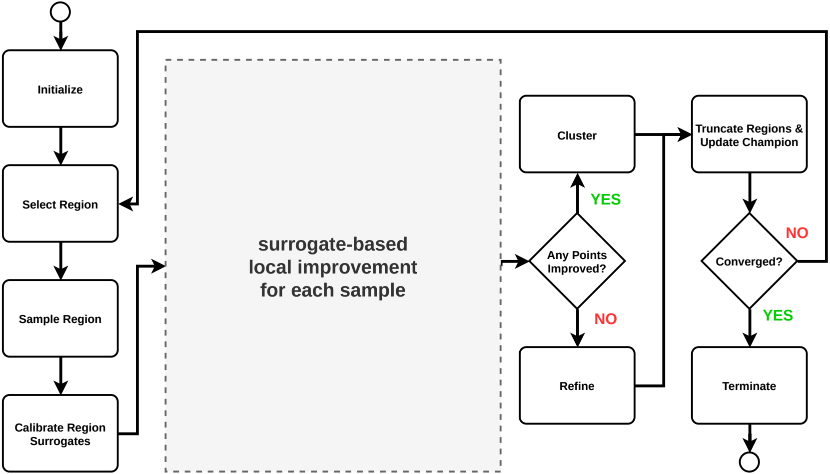
Functions $f(x,y)$ and $g(x,y)$ provided as a factorized grey-box simulation of $K \in \mathbb{N}$ assignments:

$$(e_i \triangleq E_i(a_i))_{i=1}^{K} \tag{2}$$

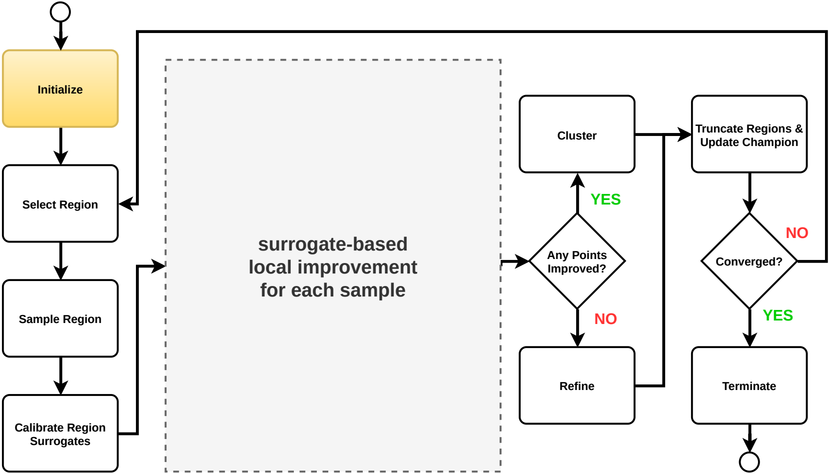
- the values of $f(x,y)$ and $g(x,y)$ correspond to particular e_i in the list
- a_i is a sequence of zero or more elements from x, y, and $(e_1, \ldots, e_{i-1})$
- E_i is one of the following:
  - a constant tensor of real-valued numbers
  - a tensor of real-valued expressions from a_i, or any slice thereof
  - a closed-form analytical expression in terms of the elements of input a_i
  - an evaluation of a black-box function $\mathbb{R}^N \to \mathbb{R}^M$ on input a_i

3 Outline

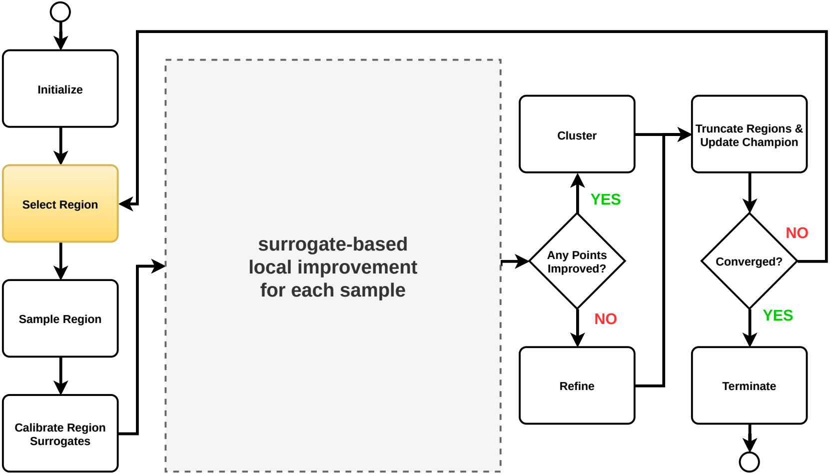
3 The GreyOpt Algorithm Framework



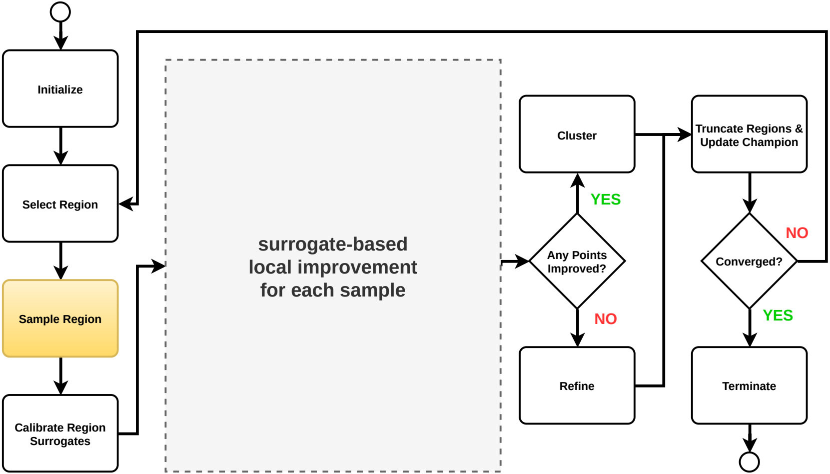
3 The GreyOpt Algorithm Framework



3 The GreyOpt Algorithm Framework



3 The GreyOpt Algorithm Framework



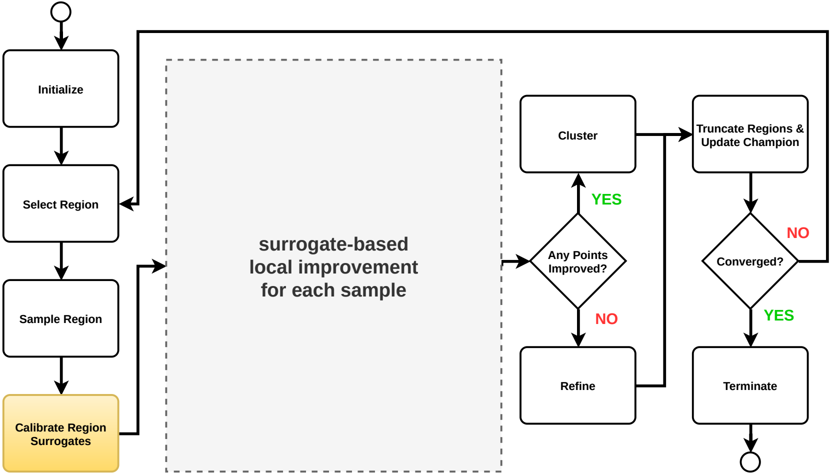
3 Sample Restoration for White-Box Constraints

$$
\begin{aligned}
\underset{x\in\mathbb{R}^n,\ y\in\mathbb{R}^m}{\text{minimize}} \quad & \|x_0 - x\|_2 + \|y_0 - y\|_2 && \text{(3a)}\\
\text{subject to} \quad & w_L \le w(x,y) \le w_U && \text{(3b)}\\
& x_L \le x \le x_U && \text{(3c)}\\
& y_L \le y \le y_U && \text{(3d)}\\
& y \in \mathbb{Z}^m && \text{(3e)}
\end{aligned}
$$

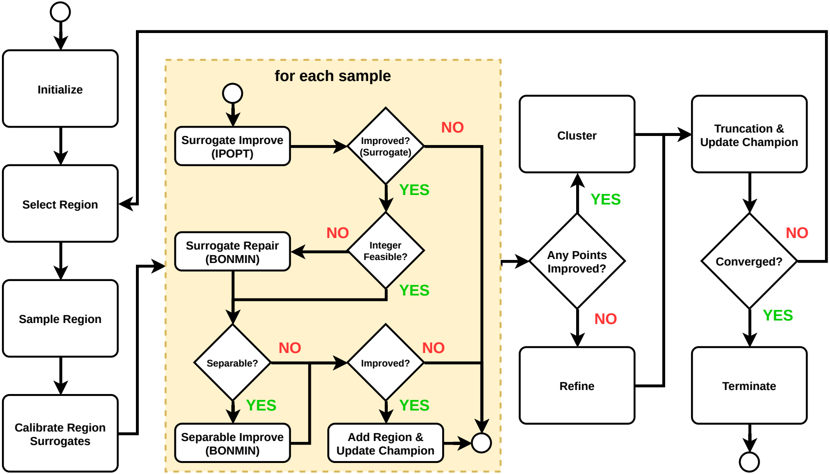
where

- (x_0, y_0) is the sample point to restore
- (x, y) are the decision variables representing the restored point
- $w : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$ is the vector-valued function of the white-box constraints from the original problem (i.e. no black-box functions)

3 The GreyOpt Algorithm Framework



3 The GreyOpt Algorithm Framework



White-Box Decision Variable

A decision variable that does not contribute to the input of any black-box functions.

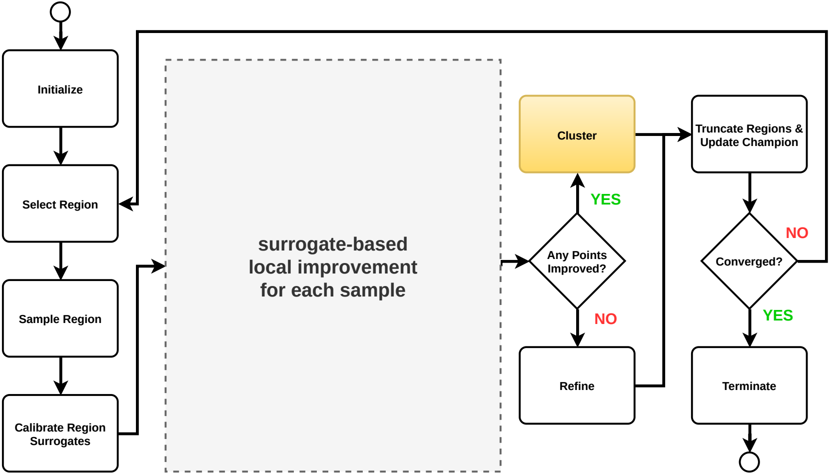
Separable Problem

A problem with at least one white-box decision variable.

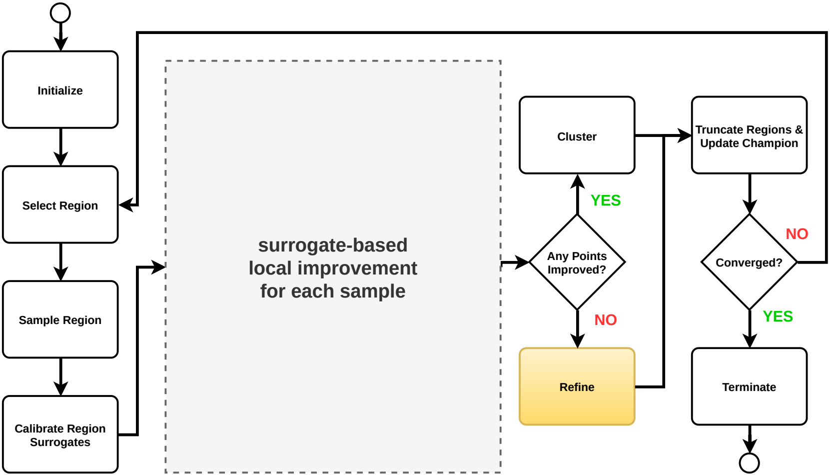
If the problem is separable, run convex mixed-integer nonlinear solver directly on the problem:

- uses initial values from improved sample for white-box decision variables
- fixes all other decision variables to the values of the current region's champion point
- uses cached values for black-box functions to prevent costly re-evaluations

3 The GreyOpt Algorithm Framework



3 The GreyOpt Algorithm Framework



3 Refinement

Otherwise, the Refine routine replaces current region with a set new regions by

- expanding the champion point of the current region, one variable at a time, into a new region until
  - constraint interval is approximately feasible
  - lower bound of objective interval is approximately lower than objectives of all other feasible champion points
- recursively partitioning this region with fathoming based on approximated interval analysis
  - want to ignore expected non-feasible and sub-optimal regions
  - Moore interval arithmetic (Moore, 1966) used with quadric under/over estimators for embedded black-box functions

3 Calibration of Quadric Surrogates for Underestimation

For each black-box function $\mathcal{B} : \mathbb{R}^n \to \mathbb{R}^m$, fit quadric underestimator:

$$\underset{A,B,C}{\text{minimize}} \quad \|Y - (AX^{\circ 2} + BX + CJ)\|_2 \tag{4a}$$

$$\text{subject to} \quad Y - (AX^{\circ 2} + BX + CJ) \geq \epsilon \tag{4b}$$

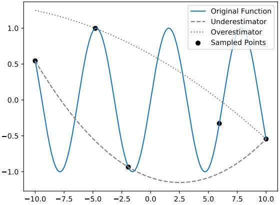
$$\forall i\ \forall j, A_{ij} \geq \epsilon \tag{4c}$$

where

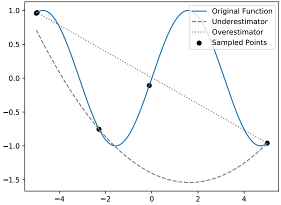
- $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{m\times 1}$ and $J \in \{1\}^{1\times s}$
- $X \in \mathbb{R}^{n\times s}$ (i.e. input samples)
- $Y \in \mathbb{R}^{m\times s}$ (i.e. output samples)
- s is the sample size
- $\circ$ denotes element-wise exponentiation
- ϵ small positive number constant to ensure convexity

Corresponding problem for quadric overestimator

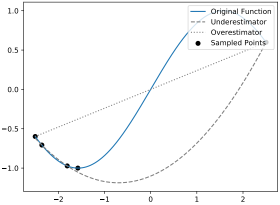
3 Calibration of Quadric Surrogates for Over/Underestimation



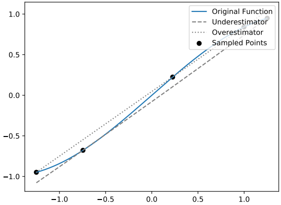
(a) Interval [-10, 10]



(b) Interval [-5, 5]

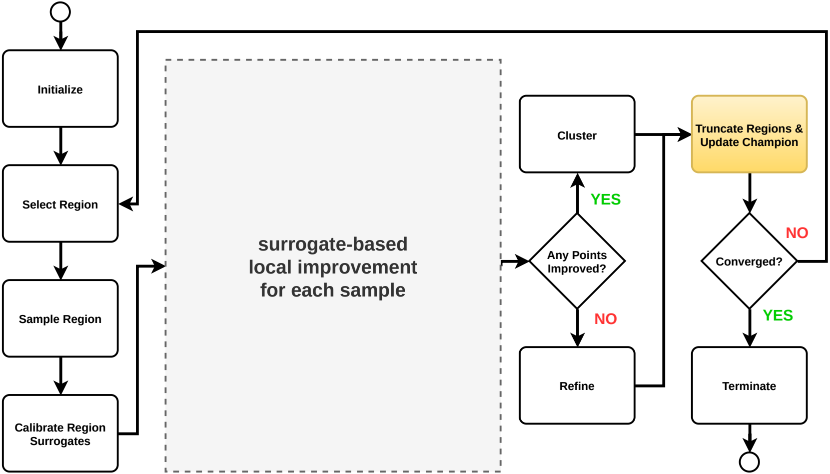


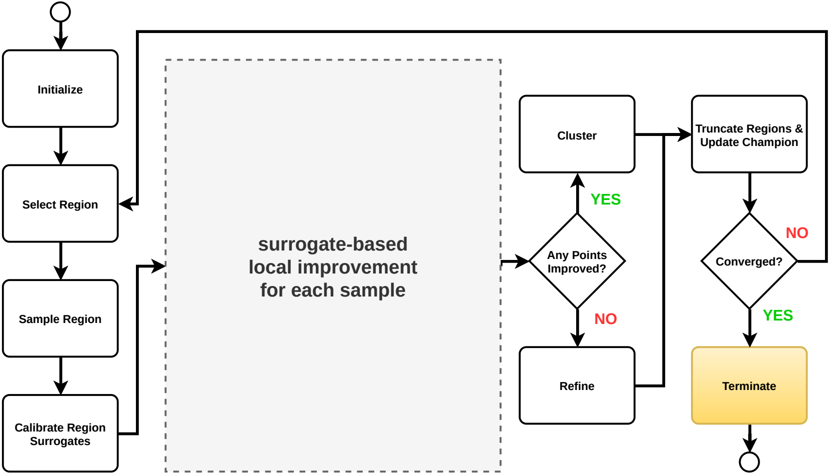
(c) Interval [-2.5, 2.5]



(d) Interval [-1.25, 1.25]

3 The GreyOpt Algorithm Framework




Initialize
Select Region
Sample Region
Calibrate Region Surrogates
surrogate-based local improvement for each sample
Cluster
Any Points Improved?
YES
NO
Refine
Truncate Regions & Update Champion
Converged?
NO
YES
Terminate

4 Test Problems

- Currently, no standard benchmark problem sets exist for MICGB
  - MINLPLib for mixed-integer nonlinear programming
  - MIPLIB for mixed-integer linear programming
  - BBOB for black-box optimization
- Developed tool to generate MICGB problems modeled in Python from MINLP problems modeled in AMPL
  - nonlinear terms in the objective and constraints replaced with calls to, otherwise equivalent, black-box functions
- From all 1704 problems in MINLPLib, study considered problems with file size less than 10KB (636 problems)
  - 310 problems successfully translated by tool (out of the 636)
  - 25 problems randomly selected for study (from the 310)
    - over 39 CPU days to complete

- Compared the performance of GreyOpt against all heuristic global optimization algorithms in Pygmo2 that support mixed-integer programming:
  - *GACO* – Extended Ant Colony Optimization (Schlüter et al., 2009)
  - *IHS* – Improved Harmony Search (Mahdavi et al., 2007)
  - *SGA* – Simple Genetic Algorithm (Oliveto et al., 2007)
    - Using Pygmo2's self-adaptive constraint handling algorithm
- Also compared against BONMIN (Bonami et al., 2008) with random restarts
  - Gradients computed by CasADi (Andersson et al., 2019)
    - Automatic differentiation for analytical expressions
    - Finite differences for black-box function calls
- Algorithm parameters were set to their defaults

4 Experimental Setup

- Black-box time (BBT) parameter controls how much additional CPU seconds for each black-box function call
  - three BBT levels tested: 0 seconds, 1 second and 10 seconds
  - implemented without wasting additional CPU cycles (i.e. accounting mechanism)
- All experiments were run on ARGO-1, a research computing cluster provided by the Office of Research Computing at George Mason University.
  - 3 BBT levels $\times$ 25 problems $\times$ 5 algorithms $=$ 375 experiments
  - 15 trials per experiment (median BBT-adjusted CPU time reported)
  - 10 CPU minutes per trial (before BBT-adjustment)
  - 937.5 CPU hours on cluster (before BBT-adjustment)

- No algorithm in the set $\mathcal{A}$ of algorithms compared is globally convergent for the set $\mathcal{P}$ of problems of the study
- Relative convergence test used for each algorithm $a \in \mathcal{A}$ on each problem $p \in \mathcal{P}$:

$$f_* - f_a >= (1 - \tau)(f_* - f^*) \tag{5}$$

  - f_*: worst objective value of the **first** feasible points found by each algorithm in $\mathcal{A}$ for problem p
  - f^*: best objective value of all feasible points found by each algorithm in $\mathcal{A}$ for problem p
  - f_a: objective value of best point found by algorithm a for problem p (∞ if not feasible)
  - $\tau \triangleq 10^{-3}$ is the tolerance parameter (same as Costa and Nannicini (2018))

4 Evaluation Methodology (continued)

Data profile (Moré and Wild, 2009) for each algorithm $a \in \mathcal{A}$:

$$d_a(x) \triangleq \frac{|\{p \in \mathcal{P} : t_{p,a} \leq x\}|}{|\mathcal{P}|} \tag{6}$$

Performance profile (Dolan and Moré, 2002) for each algorithm $a \in \mathcal{A}$:

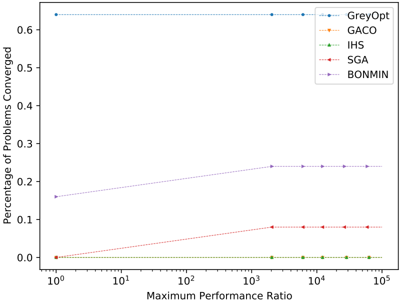
$$\rho_a(x) \triangleq \frac{|\{p \in \mathcal{P} : r_{p,a} \leq x\}|}{|\mathcal{P}|} \tag{7}$$

- $t_{p,a}$: minimum BBT-adjusted CPU seconds that algorithm a needed to converge for problem p (∞ if it failed to converge)
- $r_{p,a}$: performance ratio (Dolan and Moré, 2002) for problem $p \in \mathcal{P}$ and algorithm $a \in \mathcal{A}$

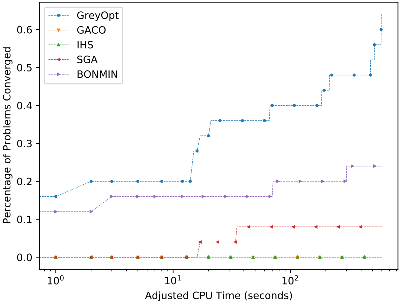
$$r_{p,a} \triangleq \frac{t_{p,a}}{min\{t_{p,a} : a \in \mathcal{A}\}} \tag{8}$$

GEORGE MASON UNIVERSITY

4 Results: Black-Box Time (BBT) = 0 seconds (per call)

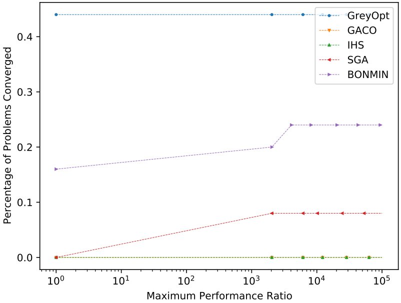


(a) Performance Profile

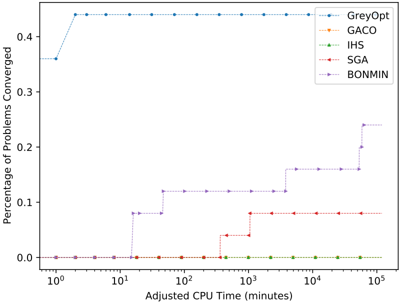


(b) Data Profile

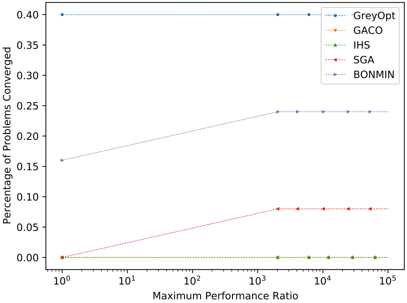
4 Results: Black-Box Time (BBT) = 1 second (per call)



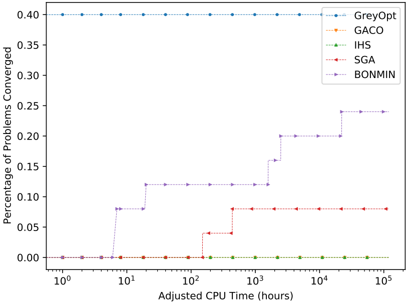
(a) Performance Profile



(b) Data Profile



(a) Performance Profile



(b) Data Profile

5 Outline

5 Conclusions

- Proposed the GreyOpt algorithmic framework for the heuristic global optimization of MICGB optimization problems
- GreyOpt shows how the partially analytical structure of MICGB optimization problems can be used to guide the exploration of the search space
  - dynamically constructed surrogates
  - approximated interval analysis
- GreyOpt significantly outperforms three black-box optimization algorithms, as well as BONMIN with random restarts, on 25 MICGB optimization problems derived from MINLPLib

5 Future Work

Possible directions for future work include:

- support for user-provided surrogate models
- support for multi-objective optimization
- better support for the optimization of problems with noisy black-box functions
- incorporating meta-optimization techniques for the problem-specific configuration of GreyOpt's parameters

6 References I

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36.

Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., and Wächter, A. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186 – 204.

Costa, A. and Nannicini, G. (2018). RBFOpt: an open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation*, 10(4):597–629.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213.

Mahdavi, M., Fesanghary, M., and Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188(2):1567–1579.

Moore, R. E. (1966). *Interval analysis*. Prentice-Hall series in automatic computation. Prentice-Hall, Englewood Cliffs, NJ.

Moré, J. J. and Wild, S. M. (2009). Benchmarking Derivative-Free Optimization Algorithms. *SIAM Journal on Optimization*, 20(1):172–191. Publisher: Society for Industrial and Applied Mathematics.

Oliveto, P. S., He, J., and Yao, X. (2007). Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, 4(3):281–293.

Schlüter, M., Egea, J. A., and Banga, J. R. (2009). Extended ant colony optimization for non-convex mixed integer nonlinear programming. *Computers & Operations Research*, 36(7):2217–2229.

Questions?

https://mason.gmu.edu/~mnachawa

mnachawa@gmail.com