C-GLIS: Global Optimization under Unknown Constraints

Mengjia Zhu, Dario Piga, Alberto Bemporad *†

June 10, 2021

1 Introduction

This report describes how the derivative-free global optimization method GLIS [1] can be extended by employing the techniques developed in C-GLISp [2] to handle (*i*) black-box constraint functions, and (*ii*) additional information on whether the function value obtained must be considered satisfactory or not. The efficiency and effectiveness of the proposed method, called C-GLIS, is assessed in three numerical benchmarks.

2 Main assumptions

For a given decision vector $x_i \in \mathbb{R}^n$, we assume that besides the objective function $f_i = f(x_i)$ also a feasibility label $G_i = G(x_i) \in \{0, 1\}$ and/or a satisfaction label $S_i = S(x) \in \{0, 1\}$ are provided to the optimization algorithm, where f, G, S are unknown functions that can only be sampled:

$$G(x) = \begin{cases} 0 & \text{if } x \notin \Omega_G \\ 1 & \text{if } x \in \Omega_G, \end{cases} \tag{1}$$

$$S(x) = \begin{cases} 0 & \text{if } x \notin \Omega_S \\ 1 & \text{if } x \in \Omega_S, \end{cases} \tag{2}$$

where Ω_G and Ω_S are the (unknown) feasibility and satisfaction set, respectively.

3 Learning unknown constraint functions

A surrogate of the probability of constraint feasibility and experiment's satisfaction is learned via an *Inverse Distance Weighting* (IDW) interpolant function [1].

We construct the surrogate $\hat{G}$ of G and $\hat{S}$ of S with a *feasibility vector* $G_F = [G_1 \ \ldots \ G_N]' \in \{0, 1\}^N$ and a *satisfaction vector* $S_F = [S_1 \ \ldots \ S_N]' \in \{0, 1\}^N$, where N is the current number of samples collected so far.

The surrogate function $\hat{G} : \mathbb{R}^{n_x} \to \mathbb{R}$ predicting the probability of satisfying the feasibility constraint $x \in \Omega_G$ is defined as

$$\hat{G}(x) = \sum_{i=1}^{N} \nu_i(x) G_i, \tag{3}$$

*M. Zhu and A. Bemporad are with IMT School for Advanced Studies Lucca, Lucca, Italy. `mengjia.zhu@imtlucca.it; alberto.bemporad@imtlucca.it`

†D. Piga is with IDSIA Dalle Molle Institute for Artificial Intelligence, SUPSI-USI, Lugano, Switzerland. `dario.piga@supsi.ch`

where $\nu_i(x) : \mathbb{R}^{n_x} \to \mathbb{R}$ for $i = 1\ldots, N$ is defined as

$$\nu_i(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{if } x = x_j, j \neq i \\ \frac{w_i(x)}{\sum_{i=1}^N w_i(x)} & \text{otherwise.} \end{cases} \quad (4)$$

Here $w_i : \mathbb{R}^{n_x} \setminus \{x_i\} \to \mathbb{R}$ is the following IDW function [3]

$$w_i(x) = \frac{e^{-d^2(x,x_i)}}{d^2(x,x_i)}, \quad (5)$$

where $d : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$ denotes the squared Euclidean distance

$$d(x, x_i) = \|x - x_i\|_2^2. \quad (6)$$

The benefit of using IDW interpolation for predicting probabilities is that $\hat{G}$ is always between 0 and 1 by construction (see [1, Lemma 1-P2]). The surrogate function $\hat{S} : \mathbb{R}^{n_x} \to \mathbb{R}$ is defined similarly.

Note that other *known* constraints $Ax \leq b$, $g(x) \leq 0$ are instead already handled in GLIS by including them during the minimization of the acquisition function, described next.

4 Acquisition function

Similar to C-GLISp [2], we account for feasibility and satisfaction terms in the acquisition function to encourage feasible exploration. The original acquisition function (eqn.[15] in [1]):

$$a(x) = \hat{f}(x) - \alpha s(x) - \delta \Delta F z(x), \quad (7)$$

is modified to

$$a(x) = \hat{f}(x) - \alpha s(x) - \delta_E \Delta F z_N(x) + \delta_G \Delta F(1 - \hat{G}(x)) + \delta_S \Delta F(1 - \hat{S}(x)), \quad (8)$$

where $\delta_E \geq 0$ is the exploration parameter, and $\delta_G, \delta_S \geq 0$ weight the probability of a sample x to be infeasible and/or unsatisfactory, respectively. Naturally, one should select $\delta_G > \delta_S$, so that the possible infeasibility is penalized more than a potential unsatisfactory behavior. For practical implementation, we suggest to adaptively tune δ_G and δ_S based on the sampled standard deviation obtained from leave-one-out cross-validation [4] of $\hat{G}$ and $\hat{S}$, respectively (see more details in [2]).

Also, in (8), z_N is the modified IDW exploration term proposed in [2]:

$$z_N(x) = \left(1 - \frac{N}{N_{max}}\right) \tan^{-1}\left(\frac{\sum_{i=1}^N r_i(x_N^*)}{\sum_{i=1}^N r_i(x)}\right) + \frac{N}{N_{max}} \tan^{-1}\left(\frac{1}{\sum_{i=1}^N r_i(x)}\right), \quad (9)$$

where $r_i(x) = \frac{1}{d^2(x,x_i)}$. This formulation is empirically observed to better escape from local minima.

5 Numerical benchmarks

We test the three numerical benchmarks noted in [2] with the C-GLIS method. Computations are run on an Intel i7-8550U 1.8-GHz CPU laptop with 8GB of RAM. The Latin hypercube sampling method [5] (*lhsdesign* function of the *Statistics and Machine Learning Toolbox* of MATLAB [6]) is used in the initial sampling phase of C-GLIS. A Monte-Carlo simulation with 100 runs of C-GLIS is performed to obtain statistically significant results. The optimizers obtained at the end of each run are depicted in 1, which shows that C-GLIS can find the constrained optimum with high probability.

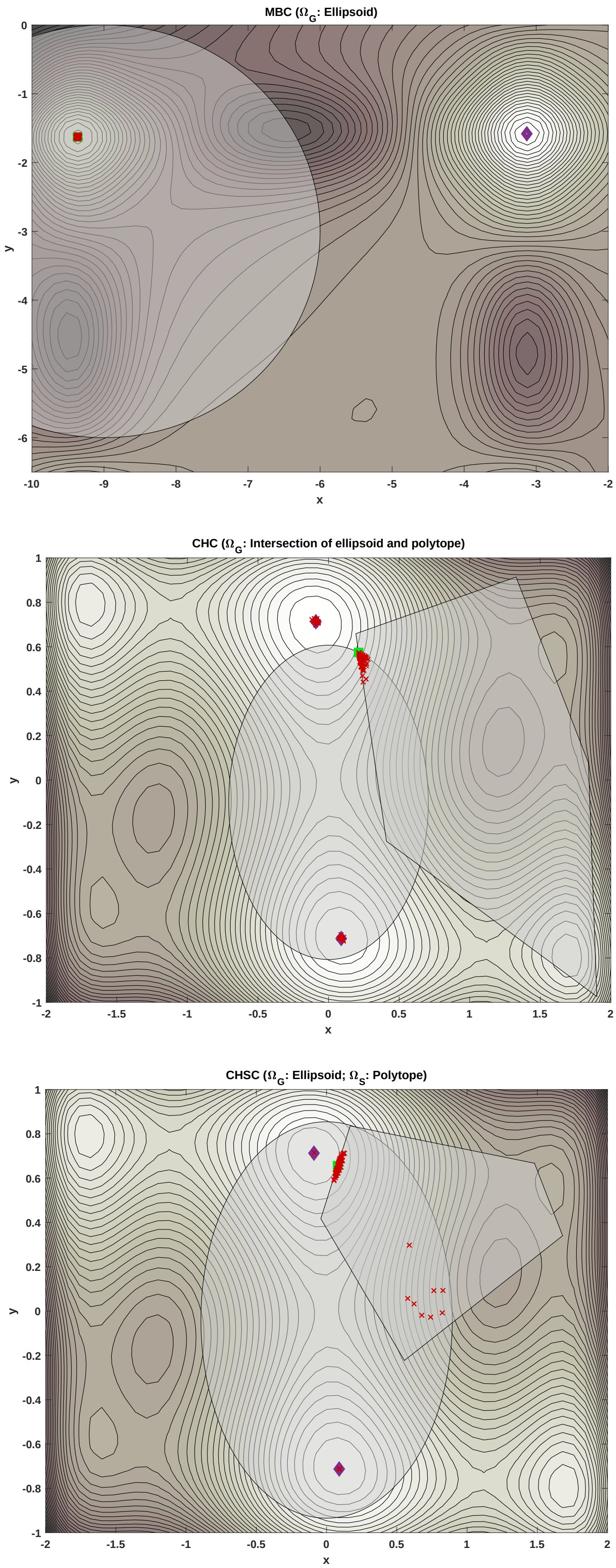

Figure 1: Algorithm C-GLIS. Optimizers computed in 100 runs on benchmark MB, CHC and CHSC. Red $\times$: optimizer computed at the end of each run; purple $\Diamond$: unconstrained optimizer; green $\Diamond$: global constrained optimizer.

References

[1] A. Bemporad, "Global optimization via inverse distance weighting and radial basis functions," *Computational Optimization and Applications*, vol. 77, pp. 571–595, 2020, code available at http://cse.lab.imtlucca.it/~bemporad/glis.

[2] M. Zhu, A. Bemporad, and D. Piga, "C-GLISp: Preference-based global optimization under unknown constraints with applications to controller calibration," submitted for publication.

[3] V. R. Joseph and L. Kang, "Regression-based inverse distance weighting with applications to computer experiments," *Technometrics*, vol. 53, no. 3, pp. 254–265, 2011.

[4] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974.

[5] M. McKay, R. Beckman, and W. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[6] The MathWorks, *Statistics and Machine Learning Toolbox*, Natick, Massachusetts, United State, 2019. [Online]. Available: https://www.mathworks.com/products/statistics.html