Comparing optimization algorithms and improving their efficiency

Etor Arza

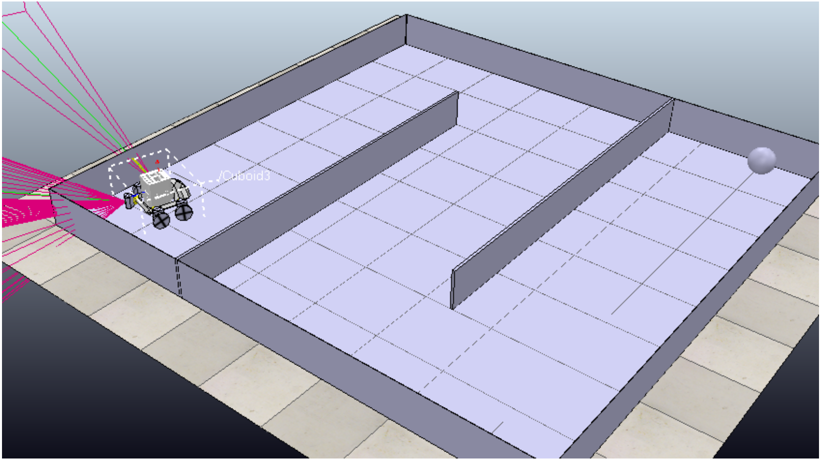
Basque Center for Applied Mathematics

earza@bcamath.org

February 22, 2023

Part IV: Early Stopping for Non-Monotone Objective Functions

Motivation



Motivation

- ▷ Clip 1

Motivation

- Clip 2

Motivation

- ▷ Clip 3

Motivation

- ▷ Clip 4

We propose an early stopping method that ...

- only looks at the objective function.
- requires no problem specific knowledge.
- is applicable to multiple problems.

Innocent Until Proven Guilty: Reducing Robot Shaping from Polynomial to Linear Time

Josh C. Bongard

***Abstract*—In evolutionary algorithms, much time is spent evaluating inferior phenotypes that produce no offspring. A common heuristic to address this inefficiency is to stop evaluations early if they hold little promise of attaining high fitness. However, the form of this heuristic is typically dependent on the fitness function used, and there is a danger of prematurely stopping evaluation of a phenotype that may have recovered in the remainder of the evaluation period. Here a stopping method is introduced that gradually reduces fitness over the phenotype's evaluation, rather than accumulating fitness. This method is independent of the fitness function used, only stops those phenotypes that are guaranteed to become inferior to the current offspring-producing phenotypes, and realizes significant time savings across several evolutionary robotics tasks. It was found that for many tasks, time complexity was reduced from polynomial to sublinear time, and time savings increased with the number of training instances used to evaluate a phenotype as well as with task difficulty.**

***Index Terms*—Early stopping, evolutionary robotics.**

I. Introduction

(e.g., [4]–[6]) of simulated robots are optimized, a single evaluation can require significant computation time. In this domain, phenotype evaluation is often terminated early if a legged robot falls down and is not equipped to right itself [7], [8] or fails to move during some interval [4], [9].

However, for many complex tasks it is non-trivial to determine whether a phenotype may have recovered fitness in the remaining evaluation time: for example in some situations an immobile robot with a recurrent neural network controller may spontaneously begin moving again. In this paper, an early stopping method is introduced that is domain independent and only stops phenotypes guaranteed to have remained inferior even if they had been evaluated fully. This method involves gradually reducing fitness over evaluation time from some theoretical maximum, rather than accumulating it. Once fitness falls below the worst of the current offspring-producing individuals in the population, its evaluation can therefore safely be terminated.

Hyperparameter optimization

- irace by López-Ibáñez et al. (2016).
- Hyperband by Li et al. (2017).
- The sequential halving algorithm by Hutter et al. (2019).
- Performance envelopes by de Souza et al. (2022).

Stop evaluating candidate θ at time step t if

$t > t_{grace}$,

and

$f[t](\theta) < f[t - t_{grace}](\theta_{best})$.

Stop evaluating candidate θ at time step t if

$t > t_{grace}$,

and

$f[t](\theta) < f[t - t_{grace}](\theta_{best})$.

Stop evaluating candidate θ at time step t if

$t > t_{grace}$,

and

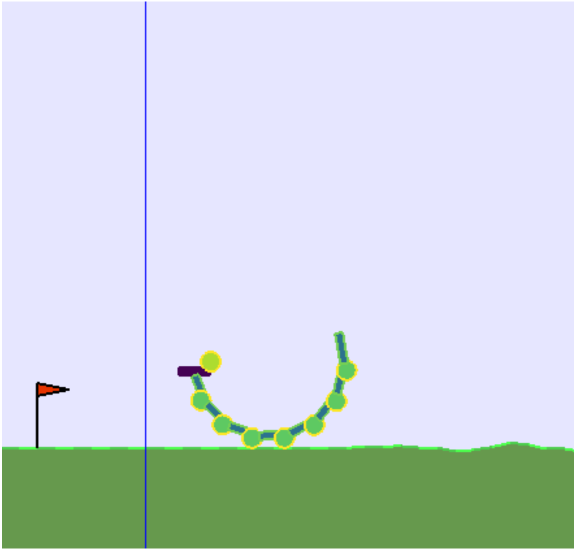
$\max\{f[t](\theta), f[t-t_{grace}](\theta)\} < \min\{f[t](\theta_{best}), f[t-t_{grace}](\theta_{best})\}$.

Experimentation

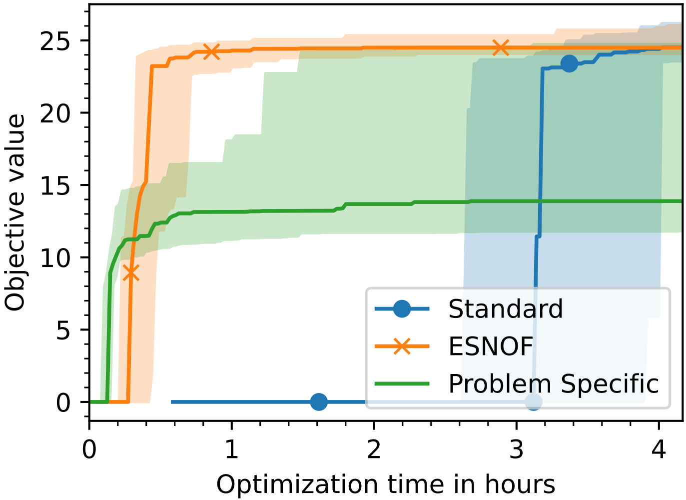
We showed that it works on:

- NIPES in the ARE framework by Le Goff et al. (2020)
- Super Mario by Verma (2020)
- Classic control by OpenAI
- MujoCo by OpenAI
- gym_rem2D by Veenstra and Glette (2020)

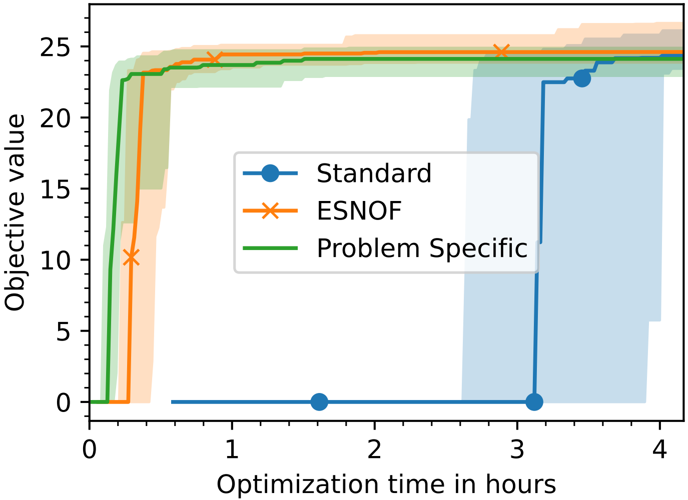
gym_rem2D by Veenstra and Glette (2020)



gym_rem2D by Veenstra and Glette (2020)



gym_rem2D by Veenstra and Glette (2020)



DANGER!

- Reporting objective values in decreasing objective functions.
- Parameter t_{grace}: $t_{grace} = 0.2 \cdot t_{max}$.
- Incompatible with Novelty search.

Repo of the code of the paper

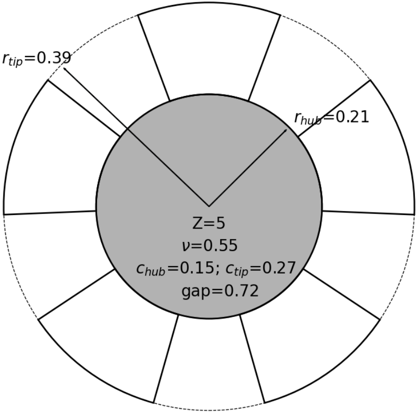
https://github.com/EtorArza/ESNOF

Part V: Variable accuracy optimization

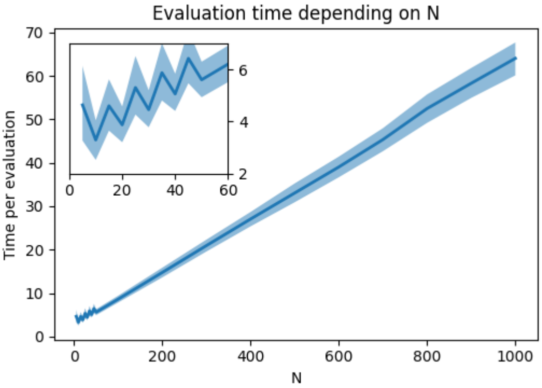
Our student Judith is working on this.

Motivating example

Motivating example



Motivating example



How can we adjust N during optimization...

- Reduce the total evaluation time
- Maximize the solution quality

Applications...

- Choose the step size in robotic simulations...
- Choose the number of monte carlo samples in optimization involving fluids...
- Choose sample size in symbolic regression...

Repo (work in progress)

https://github.com/EtorArza/VariableModelAccuracy

Part VI: Nested optimization algorithms

EvoGym Bhatia et al. (2021)

▷ Clip

RoboGrammar Zhao et al. (2020)

$\triangleright$ Clip

Microbots Liao et al. (2019)

▷ Clip

References I

Bhatia, J., H. Jackson, Y. Tian, J. Xu, and W. Matusik (2021). Evolution gym: A large-scale benchmark for evolving soft robots. *Advances in Neural Information Processing Systems 34*.

de Souza, M., M. Ritt, and M. López-Ibáñez (2022, March). Capping methods for the automatic configuration of optimization algorithms. *Computers & Operations Research 139*, 105615.

F. J. Anscombe (1973). Graphs in statistical analysis. *The American Statistician 27*(1), 17–21.

Hutter, F., L. Kotthoff, and J. Vanschoren (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.

Larrañaga, P. and J. A. Lozano (2001). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Volume 2. Springer Science & Business Media.

Le Goff, L. K., E. Buchanan, E. Hart, A. E. Eiben, W. Li, M. de Carlo, M. F. Hale, M. Angus, R. Woolley, J. Timmis, A. Winfield, and A. M. Tyrrell (2020). Sample and time efficient policy learning with CMA-ES and Bayesian Optimisation. In *The 2020 Conference on Artificial Life*, Online, pp. 432–440. MIT Press.

References II

Li, L., K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research 18*(1), 6765–6816.

Liao, T., G. Wang, B. Yang, R. Lee, K. Pister, S. Levine, and R. Calandra (2019). Data-efficient Learning of Morphology and Controller for a Microrobot. In *IEEE International Conference on Robotics and Automation (ICRA)*.

López-Ibáñez, M., J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives 3*, 43–58.

Matejka, J. and G. Fitzmaurice (2017). Same stats, different graphs: Generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1290–1294.

Ochoa, G., M. Hyde, T. Curtois, J. A. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A. J. Parkes, S. Petrovic, and E. K. Burke (2012). HyFlex: A benchmark framework for cross-domain heuristic search. In J.-K. Hao and M. Middendorf (Eds.), *Evolutionary Computation in Combinatorial Optimization*, Berlin, Heidelberg, pp. 136–147. Springer Berlin Heidelberg.

References III

Santucci, V., J. Ceberio, and M. Baioletti (2020, July). Gradient search in the space of permutations: An application for the linear ordering problem. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, Cancún Mexico, pp. 1704–1711. ACM.

Veenstra, F. and K. Glette (2020). How different encodings affect performance and diversification when evolving the morphology and control of 2D virtual creatures. In *ALIFE: Proceedings of the Artificial Life Conference*, pp. 592–601. MIT Press.

Verma, V. (2020). Applying Neural Networks and Neuroevolution of Augmenting Topologies to play Super Mario Bros.

Zhao, A., J. Xu, M. Konaković-Luković, J. Hughes, A. Spielberg, D. Rus, and W. Matusik (2020, November). RoboGrammar: Graph grammar for terrain-optimized robot design. *ACM Trans. Graph. 39*(6).