An Evolutionary and Neighborhood-based Algorithm for Optimization under Low Budget Requirements

Jordi Pereira
jorge.pereira@uai.cl
Universidad Adolfo Ibáñez
Viña del Mar, Chile

ABSTRACT

Noisy optimization problems pose several challenges to optimization algorithms under limited computational resources. The algorithm must balance the need to explore the search space and to exploit promising regions of this space. In this work we describe an algorithm that combines previous ideas to tackle the GECCO 2021 industrial challenge using an exploration and an exploitation step. The first step consists in an evolutionary algorithm combined with an experimental design, while the second phase is a neighborhood search embedded within a multi-armed ϵ-greedy approach. The resulting algorithm is not only applicable to the challenge but also to more general problems with varying conditions.

CCS CONCEPTS

• **Applied computing** → **Decision analysis**; • **Computing methodologies** → **Simulation evaluation**.

KEYWORDS

Estimation of distribution algorithms, design of experiments, optimization of simulations, ϵ-greedy.

ACM Reference Format:
Jordi Pereira. 2021. An Evolutionary and Neighborhood-based Algorithm for Optimization under Low Budget Requirements. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3463282

1 INTRODUCTION

The GECCO 2021 industrial competition deals with an optimization problem found in capacity and resource planning for hospitals where the performance of the solutions corresponds to the result of a simulation, see [1] for more information. In addition to its practical interest, the problem itself is a challenging theoretical test-bed to compare optimization algorithms. The problem has 29 continuous decision variables, each with a suggested lower and upper value. The aim is to find a value for each variable that reports the best objective value with a limit of 200 simulation runs. The budget limits the running time to reasonable values within practical settings (for instance, it takes over 30 seconds to run one simulation on a 2017 3.2 GHz Intel Xeon W processor) and discourages the use of some common techniques, like averaging, to reduce the noise within objective function evaluation.

This work describes the implementation submitted to the challenge by the author. The method combines previous ideas from different. First, exploration is performed by an estimation of distribution algorithm (EDA) [2] where candidate solutions and parameter updates follow a Taguchi experimental design [5]. The exploration phase is followed by an intensification phase where an epsilon-greedy strategy [6] balances the exploration between the neighborhood of an incumbent solution and the probabilistic model provided by the EDA phase.

2 EXPLORATION PHASE

An EDA can be seen as an evolutionary algorithm in which the population is encoded as a probabilistic model. In our case, the probabilistic model consists in a continuous uniform distribution $U(a_i, b_i)$ for each variable i, initialized with the minimum and maximum values suggested by the challenge. The main loop of the EDA performs the following actions: (1) create candidate solutions sampling from the probabilistic model; (2) evaluate solutions and (3) update the parameters of the probabilistic model to direct the search to promising regions of the search space.

The major difference between a classical EDA and the proposed implementation consists in the use of a Taguchi-like experimental design to sample (i.e., create solutions) and to update the probabilistic model. A Taguchi experimental design, see [5] for more details, identifies the controllable parameters that affect the outcome of the process (our problem). For each parameter (our variables), the experimenter defines the levels that need to be considered and creates experiments according to an orthogonal design. The result is a balanced fractional factorial design where the effect of each factor is independent and higher-order interactions are assumed to be nonexistent. Once the experiments are performed, an analysis of means of a loss function is conducted by averaging the results of the experiments, and decisions are made according to these results. These ideas are incorporated within the EDA framework as follows.

First, candidate solutions are created considering each solution as an experiment from the design. Each variable is considered as a parameter with two levels. The levels identify whether the value of the variable for the experiment will be "below" or "above" the average of its corresponding uniform distribution. The experiments are then constructed according to the smallest cardinality orthogonal array that fits the number of parameters and levels (for 29 parameters with 2 levels, the experimental design requires 36 experiments [4]).

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3463282

Second, each experiment evaluates a solution constructed according to the experimental design, i.e., if a variable i is set to "below", then the value of the said variable is a randomly drawn value from a continuous uniform distribution $U(a_i, \frac{a_i+b_i}{2})$ (if the variable is set to the "above" level, then the value is drawn from $U(\frac{a_i+b_i}{2}, b_i)$).

After performing the experiments, averages are calculated. Let m_i^- (m_i^+) be the average among all experiments with "below" ("above") level for variable i. Then, if $m_i^- < m_i^+$ ($m_i^- > m_i^+$) the probabilistic model is biased to provide candidate solutions with smaller (larger) values by shifting the mean of the probabilistic model. Additionally, the update step also reduces the range of the model to focus the search into smaller areas.

Let α and β be two algorithmic parameters to control the shift and the range reduction. Let μ_i be the mean of the uniform distribution $U(a_i, b_i)$ for any given variable i, that is $\frac{a_i+b_i}{2}$. Then, define the new range of the probabilistic model as $\beta(b_i - a_i)$ and the new mean, μ', as $\mu_i - \alpha(b_i - a_i)$ or $\mu_i + \alpha(b_i - a_i)$ according to the desired negative or positive bias. Combining both updates, the distribution becomes $U(\mu' - \beta\frac{b_i-a_i}{2}, \mu' + \beta\frac{b_i-a_i}{2})$. If the minimum or maximum parameters fall outside the ranges provided by the challenge, the parameters are modified to comply with the said limits.

These steps are repeated for a given number of iterations that depend on the computational budget. To further limit the number of evaluations required in each step, the use of smaller experimental designs was considered. If a smaller experimental design is used, the most important features of the previous designs are kept within the experiment, i.e., those with larger average difference between means, and the remaining are sampled directly from their probabilistic model without enforcing balance between levels.

3 INTENSIFICATION PHASE

The exploration phase provides an incumbent, the best-found solution, and a probabilistic model. The intensification phase tries to explore the neighborhood of the incumbent and the probabilistic model using an ϵ-greedy approach based on a multi-armed bandit model. The procedure considers a new candidate solution in each step of the phase. With probability $1 - \epsilon$ the candidate is a neighbor solution from the incumbent, and with probability ϵ it is a solution sampled from the probabilistic model. Candidates are evaluated, and replace the incumbent if they are considered to be better. These operations are repeated until exhausting the computational budget.

To generate a neighbor, the algorithm modifies the incumbent by slightly perturbing the value of each variable. For each variable i of a given solution $\mathbf{x}$, let x_i be the value of variable i in the incumbent and let ρ be a parameter that controls the neighborhood size, i.e., the level of perturbation introduced to the solution. Then, a neighbor solution $\hat{\mathbf{x}}$ is constructed as $\hat{x}_i = x_i + r(x_i^- - x_i^+, x_i^+ - x_i^-)\rho, \ \forall i$, where $r(a, b)$ is a pseudo random number generation function. To generate a solution from the probabilistic model, each value is randomly drawn from the uniform distribution associated with the said variable.

Solutions are compared according to the median value among a given number of evaluations, which was set to three for the challenge. Multiple evaluations lessen the impact of outliers within the procedure and the suggested number of evaluations still maintains a low computational footprint. Additionally, the evaluation can be stopped if an optimistic bound on the objective is shown to be worse than the incumbent. In our case, we can provide a bound of the median with two evaluations, potentially saving an evaluation for a different candidate solution. Note that the challenge aims at finding the minimum objective value among all 200 evaluations. Consequently, the incumbent of the intensification phase may differ from the best solution reported at the end of the algorithm.

4 PARAMETER SETTING AND COMPUTATIONAL RESULTS

The proposed method uses different control parameters to fine-tune its performance to specific problems and conditions. For competition purposes, the parameters were tuned using two independent runs of Irace [3], each with a computational budget of 1000 evaluations. The parameters for the first run were $\alpha = 0.188$, $\beta = 0.122$, $\epsilon = 0.18$, $\rho = 0.0962$ with two iterations of the exploration phase, while the parameters for the second run were $\alpha = 0.228$, $\beta = 0.0307$, $\epsilon = 0.0153$, $\rho = 0.0568$ with four iterations of the exploration phase, two with a 36-experiments orthogonal array followed by two with a 28-experiments orthogonal array.

These combinations were then compared by running sixty independent runs of the algorithm with each parameter set. A t-test and a Mann-Whitney test showed no statistical difference between both parameter settings. The average and standard deviation among the 120 tests were 15.4 and 1.3 units, respectively. Note some of the variability may be attributed to the objective function. We ran 100 independent evaluations for a single solution with an average objective of 24.6 units, and standard deviation of 2.9.

A detailed examination of the parameters show that the algorithm devotes between one and two thirds of the computational budget to the exploration phase, and adapts the degree of exploration within the second phase according to the time spent within the first phase (i.e., if the number of experiments allotted to the first phase is low, then it increases ϵ and ρ to include more exploration within the second phase). Consequently, while the parameters may differ significantly, the algorithm reaches a balance between exploration and exploitation within the search.

5 ACKNOWLEDGMENTS

This research has been partially funded by the Chilean Council of Scientific and Technological Research (CONICYT) through the Fondecyt grant 1191624 "Assembly line balancing for industry 4.0".

REFERENCES

[1] Thomas Bartz-Beielstein, Frederik Rehbach, Olaf Mersmann, and Eva Bartz. 2020. Hospital Capacity Planning Using Discrete Event Simulation Under Special Consideration of the COVID-19 Pandemic. arXiv:2012.07188 [stat.AP]

[2] Pedro Larrañaga and Jose A. Lozano. 2001. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Kluwer Academic Publishers, USA.

[3] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Thomas Stützle, and Mauro Birattari. 2016. The irace package: Iterated Racing for Automatic Algorithm Configuration. *Operations Research Perspectives* 3 (2016), 43–58. https://doi.org/10.1016/j.orp.2016.09.002

[4] National Institute of Standards and Technology. 2017. *Dataplot: Tabulated designs.* Retrieved April 12, 2021 from https://www.itl.nist.gov/div898/software/dataplot/designs.htm

[5] Madhan Shridhar Phadke. 1995. *Quality Engineering Using Robust Design* (1st ed.). Prentice Hall PTR, USA.

[6] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction.* A Bradford Book, Cambridge, MA, USA.