Mallows, Black-box combinatorial optimization, limited budget, ???

Ekhiñe Irurozki[1] and Manuel López-Ibáñez$^{2[0000-0001-9974-1295]}$

[1] Basque Center for Applied Mathematics
eirurozki@bcamath.org
[2] University of Málaga, Málaga, Spain
manuel.lopez-ibanez@uma.es

Abstract. Black-box combinatorial optimization problems arise in practice when the objective function is evaluated by means of a simulator or a real-world experiment. In such cases, classical techniques such as mixed-integer programming and local search cannot be applied. Moreover, often each solution evaluation is expensive in terms of time or resources, thus only a limited number of evaluations is possible, typically several order of magnitude smaller than in white-box optimization problems. In the continuous case, Bayesian optimization, in particular using Gaussian processes, has proven very effective under these conditions. Much less research is available in the combinatorial case. In this paper, we propose and analyze an estimation-of-distribution (EDA) algorithm based on a Mallows probabilistic model and compare it with CEGO, a Bayesian optimization algorithm for combinatorial optimization. Experimental results show that the Bayesian algorithm is able to obtain very good solutions with very few evaluations, however, at a significant computational cost, whereas the proposed EDA outperforms CEGO when the number of solutions evaluated approaches 400, and it is significantly faster. These results suggest that the combination of Bayesian optimization and a Mallows-based EDA may be an interesting direction for future research.

Keywords: Combinatorial optimization · Bayesian optimization · Expensive black-box optimization · Estimation of distribution algorithms

1 Introduction

Motivation: Manuel

In many practical optimization problems, the objective function contexts

Related work: Ekhine

Our contribution: Ekhine

–

? ?] ?]

[?]

2 Background

Permutations are defined as bijections of the set $[n]$ integer onto itself. The set of all permutations of n items is denoted as S_n and has cardinality $n!$. We denote permutations with Greek letters with exception of the inverse permutation denoted as $e = 1, 2, 3, \ldots, n$. We denote the composition of σ and π as $\sigma\pi$ and the inverse of σ as σ^{-1}, for which the relation $\sigma\sigma_{-1} = e$ always holds.

Distributions over permutations are functions that assign a probability value to each of the permutations in S_n, $p(\sigma) \in [0, 1]$ [**?**]. One of the most popular distributions is the Mallows Model (MM), which is considered as an analogous to the Gaussian distribution for permutations. The MM defines the probability of each permutation σ as follows:

$$p(\sigma) = \frac{\exp(-\theta d(\sigma, \sigma_0))}{\psi} \tag{1}$$

with two parameters, θ and σ_0, where permutation σ_0 a reference permutation that has the largest probability value, i.e., the mode of the distribution. The probability of every permutation $\sigma \in S_n$ decays exponentially as its distance $d(\sigma, \sigma_0)$ increases, and θ, the dispersion parameter controls this decay. The distance $d(\sigma, \sigma_0)$ is the Kendall's-τ distance. The normalization constant ψ can be easily computed for the Kendall's-τ distance as well as for the Hamming, Cayley and Ulam distance [**?**].

One of the most common problems related ti probability distributions is that of learning the maximum likelihood parameters given a sample of data S. For the MM, this problem tranlates to learning θ and σ_0 that best describe a sample of permutations.

The learning process is divide in two stages: first, we estimate the central permutation of the distribution, $\hat{\sigma}0$ and, second, compute the dispersion parameter, $\hat{\theta}$.

The exact maximum likelihood estimation is computationally hard [**?**]. However, the approximate learning requires polynomilal computational time and is guarantied to obtain high quality parameters [**? ?**].

This process of sample S is as follows: first, compute $\hat{\sigma}0$ with the Borda count algorithm. Borda orders the items $[n]$ by their *Borda score* increasingly, where the Borda score $B(i)$ is the average of each position i, $B(i) \propto \sum_{t \in S} \sigma_t(i)$. Second, the computation of θ is casted as a numerical optimization problem [**?**].

Recently, it has been proposed an extension of Borda by the name of uBorda. uBorda considers a sample of permutations S along with a weight $w(\sigma)$ for each permutation σ []. Intuitively it is equivalent to replicating the permutations in the sample proportionally to their weight. It has been used to learn a MM in an evolving preference context. In this paper, we propose to use uBorda where each permutation in the sample $\sigma \in S$ is weighted by its fitness function $w(\sigma) = f(\sigma)$.

LOP: Ekhine

Quizás luego: PFSP y QAP

3 Methods

CEGO: Manuel.

Combinatorial Efficient Global Optimization (CEGO) [?] is an extension of the well-known EGO method [?] to unconstrained black-box combinatorial optimization problems. In EGO, Gaussian process models are used as a surrogate of the landscape of the expensive original problem. An optimization method searches for solutions in the surrogate model by optimizing the expected improvement criterion, which balances the expected mean and variance of the chosen solution. Once a solution is chosen, it is evaluated on the actual objective function and the result is used to update the surrogate-model, hopefully increasing its predictive power.

CEGO replaces the Euclidean distance measure, used by the surrogate model in EGO, with a distance measure appropriate to combinatorial landscapes [?], such as the Kendall distance for permutations [?]. In CEGO, the surrogate model is explored by a GA with crossover and mutation operators appropriate for the particular combinatorial problem. The original paper notes that coupling the GA with local search does not improve the results significantly since the model is anyway an inexact estimation of the original objective function [? , p. 875].

What else do we need to say?

Description of UMM: Ekhine

new approach: add sample, learn distri, sample distri

learn distri: uMM with param ρ and binary search rho

sample: peakig distri

Quality guarantees

4 Experimental setup

We use the implementation of GECO provided by the authors[3]. Following the original paper [?], we use and the GA that optimizes the surrogate models uses a population size of 20, crossover rate of 0.5, mutation rate of 1, tournament selection of size 2 and probability of 0.9, interchange mutation (i.e., exchanging two randomly selected elements of the permutation) and cycle crossover for permutations. The budget of each run of the GA is 10^5 evaluations using the surrogate-model. Although it is never stated in the original paper, the implementation of CEGO generates a set of initial solutions of size m_{ini} by means of a max-min-distance sequential design: new solutions are added to the set sequentially by maximizing the minimum distance to solutions already in the set. These initial solutions are then evaluated on the actual objective function and the result is used to build the initial surrogate model.

Here we use $m_{\text{ini}} = 20$.

In all experiments, we consider a maximum budget of $FE_{\max} = 400$ evaluations of the actual objective function. In a white-box context, state-of-the-art

[3] https://cran.r-project.org/package=CEGO

algorithms for the problems considered here typically evaluate hundreds? thousands? of solutions, thus, the budget considered here for the black-box context is extremely limited.

We implemented UMM in Python. The parameter settings that we use are...

Complete this with details of uMM: MANUEL

Computing system details: MANUEL

Describir como se generan las intancias de LOP: Ekhine

Describir las instancias: Ekhine

5 Experimental analysis

Bayesian optimization methods using a global GP model, such as CEGO, are known to have trouble optimizing locally [**?**]. Our intuition is that this problem becomes worse in rugged combinatorial landscapes, where small steps may produce drastic changes.

6 Conclusions

Acknowledgements. Thanks to Hao Wang (Leiden University) for pointing us to the arguments of **?**].