Model-based Gradient Search using the Plackett-Luce model

VALENTINO SANTUCCI*, University for Foreigners of Perugia, Italy
JOSU CEBERIO*, University of Basque Country, Spain

CCS Concepts: • **Mathematics of computing → Probabilistic algorithms**; **Combinatorial optimization**; **Permutations and combinations**.

Additional Key Words and Phrases: Gradient search, permutation, probability distribution, combinatorial optimization, parameters adaptation

ACM Reference Format:
Valentino Santucci and Josu Ceberio. 2024. Model-based Gradient Search using the Plackett-Luce model. 1, 1 (August 2024), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 MOTIVATIONS

Computational optimization and optimization algorithms form a cornerstone area of computer science that has been extensively explored due to its myriad applications in many practical and real-world scenarios. At a high level, we can categorize computational optimization algorithms into two main classes: white-box and black-box algorithms. While the former require a certain degree of knowledge about the problem domain, the latter are more versatile and rely solely on basic domain information, such as the structure or representation of a solution. Perhaps, the most prominent examples of white-box algorithms are those which iteratively refine a solution based on the gradient of the objective function at hand. While widely used, particularly in the field of machine learning, these techniques have limited applicability in other domains where gradient information is unavailable. This is the case of Combinatorial Optimization Problems (COPs), where the domain consists of discrete solutions and, for this reason, the concept of gradient is not applicable.

In the field of black-box optimization, COPs are tackled by means of metaheuristic techniques that iteratively evolve the discrete representation of one or more solutions by means of transformation operators specifically designed for the representation at hand. Commonly used transformations are local search moves, genetic mutation and crossover operators. Another popular approach is that used by Estimation of Distribution Algorithms (EDAs), which iteratively learn and sample a probability model defined over the solution space.

However, a form of gradient-based optimization has been introduced for COPs. The idea, which is somehow connected to that of EDAs, is to maintain a probability distribution over the discrete solution space, parameterized by a real-valued vector, and such that its mass function is differentiable with respect to its parameters. Therefore, discrete solutions can be sampled from the probability model and, after being evaluated, used to estimate the expected objective value of the

*The authors contributed equally to this research.

Authors' addresses: Valentino Santucci, valentino.santucci@unistrapg.it, University for Foreigners of Perugia, Perugia, Italy; Josu Ceberio, josu.ceberio@ehu.eus, University of Basque Country, San Sebastian, Spain.

Manuscript submitted to ACM

model. Notably, the expectation of the original objective function can be expressed analytically and, more importantly, differentiated with respect to the model parameters. Interestingly, calculating the gradient of the expectation function only requires a black-box access to the objective values of the discrete samples. As a result, a standard iterative gradient search scheme (commonly referred to as *gradient ascent* for maximization or *gradient descent* for minimization) can be applied to optimize the expected value of a suitable probability model for the discrete space under consideration. Since it is straightforward to demonstrate that the optimal expected value of a model is achieved when all the probability mass is concentrated on the optimal solution of the given COP, such approach optimizes the COP at hand as well.

The approach described above is called *model-based gradient search*. While it is particularly attractive for COPs in general, prior to the very recent work of Ceberio and Santucci [1], to the best of our knowledge it has only found applications in the binary search space, arguably the simplest (and most used) space in combinatorial optimization. However, not all COPs are binary in nature. A notable example is the class of *permutation problems*, i.e., those COPs whose solutions can be represented as permutations of items.

2 THE PROPOSAL

In [1], Ceberio and Santucci introduce a model-based gradient search scheme tailored specifically for permutation problems. The proposed approach, whose acronym is GS, adopts the Plackett-Luce (PL) model – a well known probability model over permutations – and introduces an efficient procedure for computing the gradient of (the logarithm of) its probability mass function with respect to its continuous parameters.

Additionally, other algorithmic components are introduced, including adaptation mechanisms for both the learning rate and sample size parameters, the possibility to use either plain or natural gradient, an asymptotically efficient sampling procedure for the PL model, a utility transformation of the objective values to speedup GS convergence, and a soft restart mechanism to escape stagnation states with probability guarantees.

A natural competitor to GS is an EDA scheme based on the same PL model used by GS. Therefore, in [1], GS and EDA schemes have been experimentally compared on a wide set of commonly adopted benchmark instances of the Linear Ordering Problem (LOP), one of the most investigated permutation problems. The experimental results clearly show that the proposed GS outperforms the EDA scheme both in terms of effectiveness and efficiency.

3 THE ALGORITHM

By denoting the space of n-length permutations by $\mathbb{S}_n$, a PL model for $\mathbb{S}_n$ is parameterized by an n-length vector of non-negative numbers, whose entries represent the preferences of the corresponding permutation items to appear in a top rank within a permutation. To avoid violating the non-negative constraints, we maintain the logarithms of the PL parameters, therefore the probability of sampling a permutation σ from a PL model parameterized by $\mathbf{z} \in \mathbb{R}^n$ is

$$P(\sigma|\mathbf{z}) = \prod_{i=1}^{n-1} \frac{\exp z_{\sigma(i)}}{\sum_{j=i}^{n} \exp z_{\sigma(j)}}. \tag{1}$$

Hence, given the objective function $f : \mathbb{S}_n \to \mathbb{R}$ of the permutation problem at hand, the expectation function optimized by GS is

$$J(\mathbf{z}) = \mathbb{E}_{\mathbb{S}_n}\big[f(\sigma)\big] = \sum_{\sigma \in \mathbb{S}_n} f(\sigma) P(\sigma|\mathbf{z}), \tag{2}$$

while its gradient can be estimated from λ samples $\{\sigma_1, \ldots, \sigma_\lambda\}$ as

$$\nabla_{\mathbf{z}} J(\mathbf{z}) = \mathbb{E}_{\mathbb{S}_n}\big[f(\sigma) \nabla_{\mathbf{z}} \log P(\sigma|\mathbf{z})\big] \approx \frac{1}{\lambda} \sum_{i=1}^{\lambda} f(\sigma_i) \nabla_{\mathbf{z}} \log P(\sigma_i|\mathbf{z}). \tag{3}$$

In [1] it was shown that the entries of the gradient of the log-probability appearing in Eq. (3) can be efficiently calculated in $\Theta(n)$ time steps using the following formula:

$$\frac{\partial \log P(\sigma|\mathbf{z})}{\partial z_{\sigma(i)}} = 1 - \exp z_{\sigma(i)} \sum_{k=1}^{i} \frac{1}{\sum_{j=k}^{n} \exp z_{\sigma(j)}}. \tag{4}$$

Therefore, the main scheme of GS is described in Alg. 1. It requires as input: the objective function f to be (without loss of generality) maximized, the learning rate η, the sample size λ, and a utility function U.

Algorithm 1 Pseudocode of GS (maximization assumed)

Input: $f : \mathbb{S}_n \to \mathbb{R}$, $\eta \in \mathbb{R}^+$, $\lambda \in \mathbb{N}^+$, $U : \mathbb{R}^\lambda \to \mathbb{R}^\lambda$
1: $t \leftarrow 0$
2: $\mathbf{z_t} \leftarrow (0, 0, \ldots, 0)$ ▷ Uniform distribution
3: σ^* maintains the best permutation so far
4: **while** stopping criterion is not met **do**
5: **for** $i \leftarrow 1$ **to** λ **do**
6: draw σ_i from the PL model parameterized by $\mathbf{z_t}$
7: evaluate $f(\sigma_i)$ and update σ^* if an improvement is found
8: $\mathbf{g_i} \leftarrow \nabla_{\mathbf{z}} \log P(\sigma_i|\mathbf{z_t})$ ▷ See Eq. (4)
9: $u_1, u_2, \ldots, u_\lambda \leftarrow U(f(\sigma_1), f(\sigma_2), \ldots, f(\sigma_\lambda))$
10: $\nabla J(\mathbf{z_t}) \leftarrow \frac{1}{\lambda} \sum_{i=1}^{\lambda} u_i \mathbf{g_i}$
11: $\mathbf{z_{t+1}} \leftarrow \mathbf{z_t} + \eta \nabla J(\mathbf{z_t})$
12: **if** the probability mass becomes too concentrated **then**
13: $\mathbf{z_{t+1}} \leftarrow$ PL distribution with σ^* as mode solution ▷ Soft restart
14: $t \leftarrow t + 1$
15: **return** σ^*

In line 2, the $\mathbf{z}$ parameters are initialized to the same value, making the initial PL model equivalent to a uniform distribution over $\mathbb{S}_n$. Throughout each iteration of the main loop in lines 4–14, the $\mathbf{z}$ parameters are updated as follows. First, λ permutations are drawn from the current PL distribution (line 6) and evaluated (line 7). Then, the gradients of the log-probabilities (line 8) and the utilities (line 9) are used to compute the plain gradient (line 10), which is scaled by the learning rate and employed to update the PL parameters (line 11). Before proceeding to the next iteration, if the probability mass becomes too concentrated around a possibly locally optimal solution (line 12), the $\mathbf{z}$ parameters are reinitialized by setting the best solution so far as the mode of the PL distribution (line 13). Finally, the best permutation sampled is returned in line 15.

Utilities are simple monotonic transformations of the objective values of the sampled permutations. Rank-based exponentially decreasing utilities, as considered in [1], offer several advantages, including making GS invariant to monotonic transformations of the objective function in input. Furthermore, [1] also presents a variant of GS (referred to as NES) which incorporates the natural gradient concept and, more importantly, introduces two adaptation algorithms for both the learning rate and sample size parameters. Specifically, the learning rate η is dynamically adjusted by means of a novel implementation of the *cumulative stepsize adaptation* scheme, while the sample size λ is adapted by considering an entropy-like measure of the PL distribution at each iteration.

4 EXPERIMENTAL RESULTS

In [1], GS, NES and an EDA scheme based on the PL model have been experimentally compared on widely adopted benchmarks for the LOP: the IO set formed by 50 instances with sizes from 44 to 79, and the xLOLIB set with 78 instances of size 150 and 250. The algorithms have been executed 30 times per instance with a computational budget of $1000n^2$ evaluations. Effectiveness and efficiency analyses have been conducted.

The effectiveness analysis investigates the best objective values returned by any single execution of each algorithm. Fig. 1 illustrates the Bayesian analysis conducted, presenting the credibility intervals, aggregated by benchmark set, of the probability that an algorithm outperforms its competitors.

Fig. 2 shows the execution time (log seconds) required by each algorithm aggregated across executions and different instance sizes.

Figs. 1 and 2 clearly show that GS outperforms NES and EDA both in terms of effectiveness and efficiency, thus validating the proposed approach. Finally, it is worthwhile to note that [1] provides further analyses showing: the convergence behaviour of the PL distribution analyzed by different perspectives, the impact on the performance of the adaptive evolution of the algorithm parameters

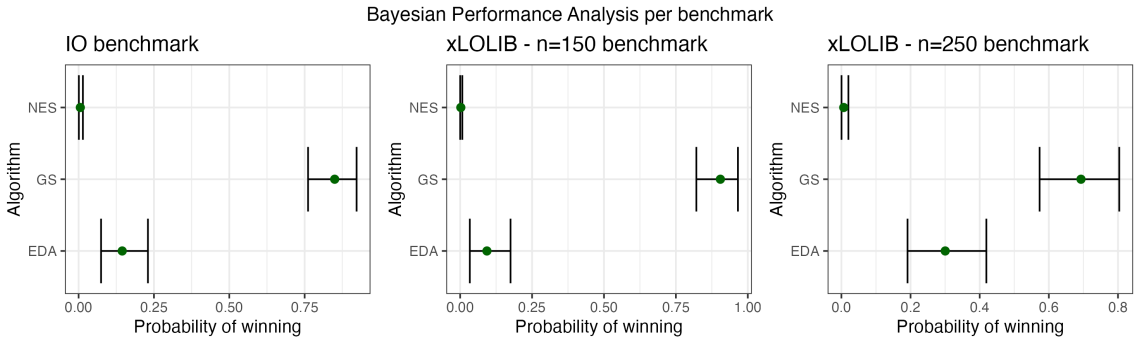

Fig. 1. Effectiveness analysis.

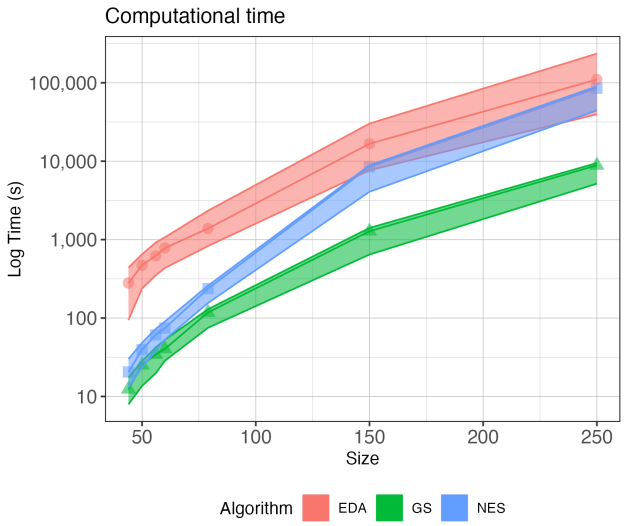

Fig. 2. Efficiency analysis.

and, though not the primary objective of the article, a comparison with respect to the best known results for the considered benchmark instances.

REFERENCES

[1] Josu Ceberio and Valentino Santucci. 2023. Model-based Gradient Search for Permutation Problems. *ACM Transactions on Evolutionary Learning and Optimization* 3, 4 (2023), 1–35.