A Multimethod Approach to Multimodal Function Optimization

Fredrik Foss
NTNU
fossfredrik@yahoo.com

Ole Jakob Mengshoel
NTNU & CMU
ole.j.mengshoel@ntnu.no

ABSTRACT

Multimodal functions play a central role in artificial intelligence. In this paper we attempt to address limitations in existing research on multimodal function optimization by developing a novel multimethod memetic algorithm (MMA). We empirically test MMA on synthetic and natural combinatorial optimization problems, including feature selection. Our initial experiments suggest that MMA preserves diversity well and consistently finds good solutions.

CCS CONCEPTS

• **Computing methodologies** → **Randomized search**; **Discrete space search**; **Heuristic function construction**.

KEYWORDS

Multimodal functions, feature selection, synthetic problems, genetic algorithms, niching, crowding, stochastic local search, clustering, feedback computing

ACM Reference Format:
Fredrik Foss and Ole Jakob Mengshoel. 2021. A Multimethod Approach to Multimodal Function Optimization. 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/10.1145/3449726.3459435

1 INTRODUCTION

Feature selection is a problem that continues to be important as datasets continue to grow, both in terms of number of features and number of cases [3, 7]. Redundant and irrelevant features not only reduce the performance of supervised learning models, but also force users to gather more data than they need in order to continue using the learned model in the future, potentially increasing cost.

Variants of local search (such as backward selection and forward selection) have traditionally been employed for wrapper-based feature selection [3]. More recently, other methods including genetic algorithms [4], stochastic local search [9, 10], and memetic algorithms [12] have been used. The plethora of methods studied suggests that the feature problem is both complex and multimodal.

Problems. While there is a broad and deep literature on multimodal optimization, we find several under-researched areas as follows. *First*, much recent research on multimodal optimization using evolutionary algorithms (EAs) has focused on continuous optimization, while in machine learning one is often concerned with combinatorial optimization. *Second*, many EAs focus on so-called repetitive problems [1], where the goal is to quickly find good solutions to problems. However, many important problems are design problems [1]; one is willing to spend a long time to find a few good solutions. *Third*, in cases where EAs perform combinatorial optimization, there has been little emphasis on diversity maintenance [14]. *Fourth*, existing multimodal optimization EAs often search for "as many local optima as possible," causing the exploring part of the population to dwindle as poor optima are found and preserved [5], thus limiting the explorative power of the algorithm.

Contributions. We attack the above problems and introduce a novel multimodal optimization algorithm for combinatorial design problems. We call it the *multimethod memetic algorithm* (*MMA*). MMA integrates stochastic local search, evolutionary computing, crowding, clustering, and feedback control. We briefly report on empirical results in this short paper.

2 PROBLEM STATEMENT

Many computational problems can be formalized as state space search and optimization. The state space considered here is the lattice of a bitstring of length n: $\{0, 1\}^n$. While broadly applicable, a natural domain (feature selection) and a synthetic domain are used to empirically demonstrate the performance of the MMA.

Feature selection can be described as an n-dimensional multimodal optimization problems, where the objectives are to minimize the number of features and maximize the performance of a learned classifier [14]. MMA uses a single-objective fitness function that linearly combines maximizing classification accuracy and minimizing the number of features.

In addition to feature selection, we study synthetic continuous multimodal optimization problems [11]. These synthetic test functions, which we adapt to our combinatorial setting, are scalable to any number of dimensions and are harder than many other synthetic fitness functions when scaled appropriately.

3 MULTIMETHOD MEMETIC ALGORITHM

The MMA attempts to find multiple locally optimal, or close-to locally optimal, solutions to a combinatorial optimization problem. In large feature selection datasets, the solution with the highest classification accuracy will often be the best choice, but if the dataset is poor, it may be advantageous to have multiple candidate solutions so that a decision maker can sanity check multiple candidate solutions. In order to support this use case, and similar ones, the MMA allows the user to specify the number of optima ℓ to be found. Due to the stochastic nature of the MMA it is impossible to guarantee that exactly ℓ optima are found, but results are usually close.

In short, MMA works as follows: At the start of every generation, a clustering algorithm partitions the population into niches. A feedback controller uses the niching-data to adjust the generalized crowding GA (GCGA) scaling factor ϕ [2]. The goal is to have ℓ

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

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459435

distinct niches in the population. A specialized algorithm, *LazySLS*, is applied to the best individual in each niche. If a niche has more than Ψ individuals, surplus individuals are culled from the population. Afterwards, the best niches are marked for elitist survival. Finally, GCGA evolves a new generation of candidate solutions.[1]

Large values for GCGA's parameter ϕ have a tendency to increase the number of niches in the population, whereas small values of ϕ have the opposite effect, tending to abandon poor niches in favour of better ones [2]. The feedback controller balances exploration and exploitation by modifying ϕ mid-run. Let $|N|$ denote the number of niches in the population and ℓ the number of niches the controller should aim for. If $|N| \neq \ell$, the controller adjusts ϕ based on the difference between $|N|$ and ℓ [13].

One can use any clustering algorithm with MMA, but some traits are important to optimize performance. First, the clustering algorithm needs to include outliers. If outliers are not clustered, lone niches will be ignored by local search and elitism. Second, the clustering algorithm should not cluster too aggressively. If close but distinct niches are grouped together, only one of them will survive. A third desireable trait is speed. Clustering is performed once per MMA generation and some algorithms have high computational complexity. So far, our empirical evidence points to MEC [6] as being the most suitable clustering algorithm.

Stochastic local search (SLS) seeks to reach local optima by means of different heuristics, such as noise, greedy, and restart steps [8, 9]. The noise step provides an explorative drive by flipping a random bit, while the greedy step works by testing all possible bitflips until an improvement is found. Feature selection problems vary in size, with high-dimensional problems having thousands of features. With a naive implementation of the greedy step, simple SLS is very slow when applied to high-dimensional problems. As MMA attempts to optimize multiple local optima at once, using simple SLS on all niches is too time-consuming for high-dimensional problems.

To handle this problem, a novel SLS variant dubbed *LazySLS* is introduced. Instead of trying all possible bitflips, LazySLS will try γ randomly chosen indices. If no improvement is found, LazySLS assumes optimality and stops local search. If run a sufficient number of times, LazySLS is guaranteed to reach a local optima. Individual niches will be optimized multiple times over the course of a run. Instead of exhaustively optimizing each niche at one time and marking it complete, this MMA approach will reach local optima in a slower, more controlled fashion. This allows MMA to do global search and local search concurrently.

4 EXPERIMENTS AND CONCLUSION

While space does not allow for a detailed discussion of our experiments, we provide some highlights for a feature selection experiment in which a naive Bayes classifier was used. As seen in Figure 1, MMA uses fewer fitness evaluations to reach better results when γ is optimized. This improvement happens, we hypothesize, mostly because LazySLS spends less time optimizing optima that later are abandoned. Another potential reason for the performance increase is paradoxically because local optima are being reached

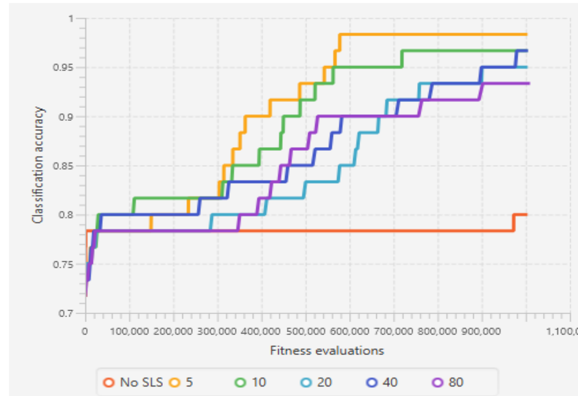

Figure 1: MMA without SLS and with various values for lazySLS's γ; γ varies from $\gamma = 5$ to $\gamma = 80$. Fitness evaluation is on the x-axis; classification accuracy is on the y-axis. The best value is $\gamma = 5$ from around 300 000 evaluations.

more slowly, preventing local optima from dominating promising global search avenues in the parent selection phase.

More generally, experiments suggest that MMA preserves diversity and consistently finds good solutions for both natural and synthetic functions. Various ablation studies and deeper dives into the particular contributions of the different parts of MMA have been performed. Results indicate that LazySLS is an interesting contribution worth further research, as it provides the benefits without spending too much time exploring non-optimal niches, thus significantly outperforming simple SLS.

[1] The reason for this perhaps unconventional ordering of heuristics is the following. The EA-step depends upon the elitism, which in turn depends upon the niching data.

REFERENCES

[1] A. E. Eiben and J. E. Smith. 2015. *Introduction to Evolutionary Computing* (2nd ed.). Springer-Verlag Berlin Heidelberg, Chapter 9.1.
[2] S. F. Galan and O. J. Mengshoel. 2010. Generalized Crowding for Genetic Algorithms. In *Proc. of the Genetic and Evolutionary Computation Conference 2010 (GECCO-10)*. 775–782.
[3] I. Guyon and A. Elisseeff. 2003. An introduction to variable and feature selection. *JMLR* 3 (2003), 1157–1182.
[4] M. M. Kabir, M. Shahjahan, and K. Murase. 2011. A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing* 74, 17 (2011), 2914–2928.
[5] M. Kronfeld and A. Zell. 2010. Towards scalability in niching methods. In *IEEE Congress on Evolutionary Computation*. 1–8.
[6] H. Li, K. Zhang, and T. Jiang. 2004. Minimum Entropy Clustering and Applications to Gene Expression Analysis. (2004).
[7] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. 2017. Feature Selection: A Data Perspective. *ACM Comput. Surv.* 50, 6 (2017).
[8] O. J. Mengshoel. 2008. Understanding the Role of Noise in Stochastic Local Search: Analysis and Experiments. *Artificial Intelligence* 172, 8-9 (2008), 955–990.
[9] O. J. Mengshoel, Y. Ahres, and T. Yu. 2016. Markov Chain Analysis of Noise and Restart in Stochastic Local Search. In *Proc. IJCAI*. 639–646. http://www.ijcai.org/Abstract/16/097
[10] O. J. Mengshoel, D. C. Wilkins, and D. Roth. 2011. Initialization and Restart in Stochastic Local Search: Computing a Most Probable Explanation in Bayesian Networks. *IEEE TKDE* 23, 2 (2011), 235–247.
[11] B. Y. Qu, J. J. Liang, Z. Y. Wang, Q. Chen, and P. N. Suganthan. 2015. Novel benchmark functions for continuous multimodal optimization with comparative results. *Swarm and Evolutionary Computation* 26 (2015), 23–24.
[12] W. Sheng, X. Liu, and M. Fairhurst. 2008. A Niching Memetic Algorithm for Simultaneous Clustering and Feature Selection. *IEEE Transactions on Knowledge and Data Engineering* 20, 7 (2008), 868–879.
[13] J. Shi, O. J. Mengshoel, and D. K. Pal. 2014. Feedback Control for Multi-Modal Optimization Using Genetic Algorithms. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 839–846.
[14] B. Xue, M. Zhang, W. N. Browne, and X. Yao. 2016. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation* 20, 4 (2016), 606–626.