Efficient Black-Box Combinatorial Optimization

Hamid Dadkhahi, Karthikeyan Shanmugam, Jesus Rios, and Payel Das

IBM Research NY

{hdadkhahi, karthikeyan.shanmugam2}@ibm.com, {jriosal, daspa}@us.ibm.com

Abstract

We consider the problem of black-box function optimization over combinatorial domains. Despite the vast literature on black-box function optimization over continuous domains, not much attention has been paid to learning models for optimization over combinatorial domains until recently. Nonetheless, optimization of such functions using state-of-the-art Bayesian optimization techniques, specifically designed for combinatorial domains, still remains challenging due to the associated computational complexity. To address this problem, we propose a computationally efficient model learning algorithm based on exponential weight updates. In particular, we use multilinear polynomials as surrogate model for functions over the Boolean hypercube. We further extend this representation via a group-theoretic Fourier expansion to address functions over categorical variables. Numerical experiments over synthetic benchmarks as well as real-world biological sequence design problems demonstrate the competitive or superior performance of the proposed algorithms versus a state-of-the-art Bayesian optimization algorithm while reducing the computational cost by multiple orders of magnitude.

Keywords: Black-Box functions, Combinatorial Optimization, Learning with Expert Advice

1. Introduction

A plethora of practical optimization problems involve black-box functions, with no simple analytical closed forms, that can be evaluated at any arbitrary point in the domain. Optimization of such black-box functions poses a unique challenge due to restrictions on the number of possible function evaluations, as evaluating functions of real-world complex processes is often expensive and time consuming. Efficient algorithms for global optimization of expensive black-box functions take past queries into account in order to select the next query to the black-box function more intelligently. While black-box optimization of real-world functions defined over integer, continuous, and mixed variables has been studied extensively in the literature, limited work has addressed incorporation of purely categorical type input variables.

Categorical type variables are particularly challenging when compared to integer or continuous variables, as they do not have a natural ordering. However, many real-world functions are defined over categorical variables. One such problem, which is of wide interest, is the design of optimal chemical or biological (protein, RNA, and DNA) polymer sequences, which are constructed using a fixed vocabulary. In particular, in the case of proteins and nucleic acids (DNA/RNA), there are 20 amino acids and 4 nucleotides, respectively. Designing optimal DNA, RNA, and protein sequences with improved or novel structures and/or functionalities is of paramount importance in drug and vaccine design, synthetic biology and many other applications Dixon et al. (2010); Ng et al.

(2019); Hoshika et al. (2019); Yamagami et al. (2019). Design of optimal sequences is a difficult black-box optimization problem over a combinatorially large search space Stephens et al. (2015), in which function evaluations often rely on either wet-lab experiments, physics-inspired simulators, or knowledge-based computational algorithms, which are slow and expensive in practice.

2. Related Work

A variety of discrete search algorithms and meta-heuristics have been studied in the literature for combinatorial optimization over categorical variables. Such algorithms, including Genetic Algorithms, Simulated Annealing, and Particle Swarms, are generally inefficient in finding the global minima. Bayesian Optimization (BO) is a commonly used approach for optimization of black-box functions Shahriari et al. (2015). BO builds a surrogate model for the black-box function via Bayesian models such as a Gaussian Process (GP) and then selects the next candidate point for evaluation via an acquisition function. However, limited work has addressed incorporation of categorical variables in BO. Early attempts based on converting the black-box optimization problem over categorical variables to that of continuous variables have been unsuccessful Gómez-Bombarelli et al. (2018); Golovin et al. (2017); Garrido-Merchán and Hernández-Lobato (2020).

A few BO algorithms have been specifically designed for black-box functions over combinatorial domains. In particular, the BOCS algorithm Ricardo Baptista (2018), primarily devised for boolean functions, employs a sparse monomial representation to model the interactions among different variables. A sparse Bayesian linear regression framework is then used to learn the coefficients of the model. The COMBO algorithm Oh et al. (2019) upgrades BOCS in that it is capable of accounting for arbitrarily high orders of interactions among variables. Their main technique is to use Graph Fourier Transform (GFT) over a combinatorial graph, constructed via graph cartesian product of variable subgraphs, to gauge the smoothness of the black-box function. A GP, equipped with an automatic relevance determination diffusion kernel, is proposed for which GFT can be carried out tractably. However, both BOCS and COMBO are hindered by associated high computational costs. The computational complexity of the latter algorithms not only increases with the number of variables, but also grows polynomially with respect to the number of function evaluations.

3. Black-Box Optimization over Boolean or Categorical Variables

Given the combinatorial categorical domain $\mathcal{X} = [k]^n$, with n variables each of cardinality k, the objective is to find

$$x^* = \arg\min_{x \in \mathcal{X}} f(x) \tag{1}$$

where $f : \mathcal{X} \mapsto \mathbb{R}$ is a real-valued combinatorial function. We assume that f is a black-box function, which is potentially noisy and computationally expensive to evaluate. As such, we are interested in finding x^* with as few evaluations as possible.

In order to address this problem, we adopt a surrogate model learning framework, where an estimate for the black-box function (i.e. the surrogate model) is updated sequentially using the black-box function evaluation observed at any given time step t. The selection of candidate points for black-box function evaluation is carried out via an acquisition function, which takes advantage of the surrogate model as an inexpensive proxy for the true black-box function. Finally, the generated sample is plugged into the black-box function for evaluation. This process is repeated until a stopping criterion, such as an evaluation budget or a time budget, is met.

In the sequel, we first propose a surrogate model based on multilinear polynomial representation for functions over the Boolean hypercube. We then extend this model to functions over categorical variables using a group-theoretic Fourier expansion. This is a direct generalization in the sense that the latter reduces to the former representation for real-valued Boolean functions when the cardinality of the categorical variables is two.

3.1 Surrogate Model

Boolean Case: Any real-valued Boolean function can be uniquely expressed by its *multilinear polynomial* representation O'Donnell (2014):

$$f(x) = \sum_{\mathcal{I} \subseteq [n]} \alpha^*_{\mathcal{I}} \psi_{\mathcal{I}}(x) \tag{2}$$

which is referred to as the *Fourier* expansion of f, the real number $\alpha^*_{\mathcal{I}}$ is called the Fourier coefficient of f on $\mathcal{I}$, and $\psi_{\mathcal{I}}(x) = \Pi_{i \in \mathcal{I}} x_i$ are monomials of order $|\mathcal{I}|$. The generality of Fourier expansions and the monomials' capability to capture interactions among different variables, make this representation particularly attractive for problems over the Boolean hypercube. In addition, in many applications of interest monomials of orders up to $m << n$ are sufficient to capture interactions among the variables, reducing the number of Fourier coefficients from 2^n to $d = \sum_{i=0}^{m} \binom{n}{i}$. This leads to the following approximate surrogate model for f:

$$\widehat{f}_\alpha(x) = \sum_{i \in [d]} \alpha_i \psi_i(x). \tag{3}$$

We employ the latter representation as the surrogate model in our proposed algorithm.

Categorical Case: We define a cyclic group structure $\mathbb{Z}/k_i\mathbb{Z}$ over the elements of each categorical variable x_i ($i \in [n]$), where k_i is the cardinality of the latter variable. From the fundamental theorem of abelian groups Terras (1999), there exists an abelian group G which is isomorphic to the direct sum (a.k.a direct product) of the cyclic groups $\mathbb{Z}/k_i\mathbb{Z}$ corresponding to the n categorical variables:

$$G \cong \mathbb{Z}/k_1\mathbb{Z} \oplus \mathbb{Z}/k_2\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}/k_n\mathbb{Z} \tag{4}$$

where the latter group consists of all vectors $(a_1, a_2, \ldots, a_n)$ such that $a_i \in \mathbb{Z}/k_i\mathbb{Z}$ and $\cong$ denotes group isomorphism. We assume that $k_i = k$ ($\forall i \in [n]$) for simplicity, but the following representation could be easily generalized to the case of arbitrary cardinalities for different variables.

The Fourier representation of any complex-valued function $f(x)$ on the finite abelian group G is given by Terras (1999)

$$f(x) = \sum_{\mathcal{I} \in [k]^n} \alpha_{\mathcal{I}} \psi_{\mathcal{I}}(x) \tag{5}$$

where $\alpha_{\mathcal{I}}$ are (in general complex) Fourier coefficients, $[k]^n$ is the n-fold cartesian product of the set $[k]$ and $\psi_{\mathcal{I}}(x)$ are complex exponentials (k-th roots of unity) given by

$$\psi_{\mathcal{I}}(x) = \exp(2\pi j \langle x, \mathcal{I} \rangle / k).$$

Note that the latter complex exponentials are the *characters* of the representation, and reduce to the *monomials* (i.e. in $\{-1, 1\}$) when the cardinality of each variable is two. A second order approximation of the representation in (5) can be written as:

$$\widehat{f}_\alpha(x) = \alpha_0 + \sum_{i\in[n]} \sum_{\ell\in[k-1]} \alpha_{i\ell} \exp(2\pi j x_i \ell / k) + \sum_{(i,j)\in\binom{[n]}{2}} \sum_{(p,q)\in[k-1]^2} \alpha_{ijpq} \exp(2\pi j (x_i p + x_j q) / k). \quad (6)$$

For a real-valued function $f_\alpha(x)$ (which is of interest here), the representation in (5) reduces to

$$f_\alpha(x) = \Re\Big\{ \sum_{\mathcal{I}\in[k]^n} \alpha_{\mathcal{I}} \psi_{\mathcal{I}}(x) \Big\} = \sum_{\mathcal{I}\in[k]^n} \alpha_{r,\mathcal{I}} \psi_{r,\mathcal{I}}(x) - \sum_{\mathcal{I}\in[k]^n} \alpha_{i,\mathcal{I}} \psi_{i,\mathcal{I}}(x) \quad (7)$$

where

$$\psi_{r,\mathcal{I}}(x) = \cos(2\pi \langle x, \mathcal{I}\rangle / k) \quad \texttt{and} \quad \psi_{i,\mathcal{I}}(x) = \sin(2\pi \langle x, \mathcal{I}\rangle / k) \quad (8)$$

The number of characters utilized in this representation with maximum order of interactions m is equal to $d = 2\sum_{i=0}^{m} \binom{n}{i}(k-1)^i - 1$.

3.2 The Algorithm

Motivated by the properties of the hedge algorithm Arora et al. (2012), we adopt an exponential weight update rule for our surrogate model. More precisely, we maintain a pool of experts, each of which corresponds to a monomial $\psi_i(x)$ or a character $\psi_{\ell,\mathcal{I}}(x)$ in the Boolean and categorical cases, respectively. Henceforth, we denote both types of experts with ψ_i ($i \in [d]$) for ease of notation. In particular, we are interested in finding the optimal Fourier coefficient α_i for the expert ψ_i. Note that exponential weights are non-negative, while the Fourier coefficients could be either negative or positive. Following the same approach as sparse online linear regression literature Kivinen and Warmuth (1997), we maintain two non-negative coefficients for each Fourier coefficient α_i^t at time step t: $\alpha_{i,+}^t$ and $\alpha_{i,-}^t$. The value of the Fourier coefficient is then obtained via the subtraction $\alpha_i^t = (\alpha_{i,+}^t - \alpha_{i,-}^t)$.

More specifically, our algorithm works in the following way. We initialize the Fourier coefficients $\alpha_{i,-}$ and $\alpha_{i,+}$ ($\forall i \in [d]$) with a uniform prior. In each time step t, the algorithm produces a sample point x_t via simulated annealing over its current estimate for the Fourier representation $\widehat{f}_{\alpha^t}$ with Fourier coefficients α^t. We then observe the black-box function evaluation $f(x_t)$ for our query x_t. This leads to a mixture loss ℓ_t which is equal to the difference between the evaluations obtained by our estimate model and the black-box function. This mixture loss, in turn, leads to the individual losses $\ell_i^t = 2\,\lambda\,\ell_t\,\psi_i(x_t)$ for the experts $\psi_i : \forall i \in [d]$. Finally, we update the current estimate for the Fourier coefficients α^t via the exponential weight update rule, incorporating the incurred losses. We repeat this process until the stopping criteria are met. Note that we use the anytime learning rate schedule of Gerchinovitz and Yu (2011), which is a decreasing function of time t. A summary of the proposed algorithm, which we refer to as *Expert-Based Combinatorial Optimization* (ECO), is given in Algorithm 1.

4. Experiments and Results

In this section, we compare the performance of the proposed algorithm with two baselines, random search (RS) and simulated annealing (SA), as well as a state-of-the-art Bayesian combinatorial optimization algorithm (COMBO) Oh et al. (2019). In particular, we consider a synthetic benchmark (Latin square problem) and a real-word sequence design problem in biology: RNA sequence

Algorithm 1: Expert Combinatorial Optimization

```
Input: sparsity $\lambda$, maximum order of interactions $m$
$t = 0$
$\forall \gamma \in \{-,+\}$ and $\forall i \in [d] : \alpha_{i,\gamma}^{t} = \frac{1}{2d}$
repeat
    $x_t \sim \widehat{f}_{\alpha^t}$
    Observe $f(x_t)$
    $\widehat{f}_{\alpha^t}(x) \leftarrow \sum_{i \in [d]} \big(\alpha_{i,+}^{t} - \alpha_{i,-}^{t}\big)\, \psi_i(x)$
    $\ell^{t+1} \leftarrow \widehat{f}_{\alpha^t}(x_t) - f(x_t)$
    for $i \in [d]$ and $\gamma \in \{-,+\}$ do
        $\ell_i^{t+1} \leftarrow 2\,\lambda\,\ell^{t+1}\,\psi_i(x_t)$
        $\alpha_{i,\gamma}^{t+1} \leftarrow \alpha_{i,\gamma}^{t} \exp\big(-\,\gamma\,\eta_t\,\ell_i^{t+1}\big)$
        $\alpha_{i,\gamma}^{t+1} \leftarrow \lambda \cdot \frac{\alpha_{i,\gamma}^{t+1}}{\sum_{\mu \in \{-,+\}} \sum_{j \in [d]} \alpha_{j,\mu}^{t+1}}$
    end
    $t \leftarrow t + 1$
until Stopping Criteria
return $\widehat{x}* = \arg\min_{\{x_i : \forall i \in [t]\}} f(x_i)$
```

optimization. In addition to the performance of the algorithms in terms of the best value of $f(x)$ observed until a given time step t, we measure the average computation time per time step of our algorithm versus that of COMBO. In each experiment, we report the results averaged over 20 runs ± one standard error of the mean. The maximum degree of interactions used in our surrogate models is set to two in all the problems. The sparsity parameter λ in exponential weight updates is set to 1 in all the experiments. See Dadkhahi et al. (2020) for results on black-box optimization over the Boolean hypercube.

Synthetic Benchmark: We first consider the Latin square problem Colbourn and Dinitz (2006), which is a commonly used combinatorial optimization benchmark. We set $n = 25$ categorical variables, with each variable of cardinality $k = 5$. A Latin square of order k is a $k \times k$ matrix of elements $x_{ij} \in [k]$, such that each number appears in each row and column exactly once. When $k = 5$, the problem of finding a Latin square has $161,280$ solutions in a space of dimensionality 5^{25}. We formulate the problem of finding a Latin square of order k as a black-box function by imposing an additive penalty of one for any repetition of numbers in any row or column. As a result, function evaluations are in the range $[0, 2k(k-1)]$, and a function evaluation of zero corresponds with a Latin square of order k. We consider a noisy version of this problem, where an additive Gaussian noise with zero mean and standard deviation of $\sigma = 0.1$ is added to function evaluations observed by each algorithm. As depicted in Figure 1, ECO outperforms the baselines with a considerable margin. In addition, ECO is able to match COMBO's performance until time step $t = 150$. At larger time steps, COMBO outperforms the other algorithms; however, this performance comes at the price of a far larger computation time. As demonstrated in Table 4, ECO offers a speed-up over COMBO by a factor of approximately 50.

RNA Sequence Optimization Problem: Structured RNA molecules play a critical role in many biological applications, ranging from control of gene expression to protein translation. The

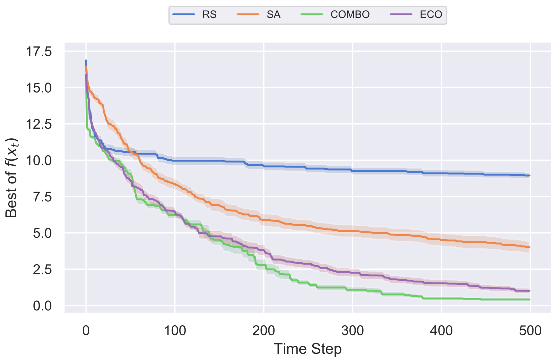

Figure 1: Latin Square Problem

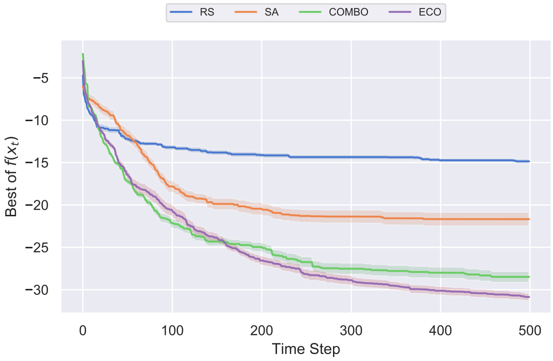

Figure 2: RNA Sequence Optimization Problem

Dataset	n	k	COMBO	ECO
Latin Square	25	5	170.4	3.6
Sequence Prediction	30	4	253.8	5.7

Table 1: Average computation time per step (in Seconds) over different problems and algorithms.

native secondary structure of a RNA molecule is usually the minimum free energy (MFE) structure. Consider an RNA sequence as a string $A = a_1 \ldots a_n$ of n letters (nucleotides) over the alphabet $\Sigma = \{A, U, G, C\}$. A pair of complementary nucleotides a_i and a_j, where $(i < j)$, can interact with each other and form a base pair (denoted by (i, j)), A-U, C-G and G-U being the energetically stable pairs. Thus, the secondary structure of an RNA can be represented by an ensemble of pairing bases.

Finding the most stable RNA sequences has immediate applications in material and biomedical applications Li et al. (2015). Studies show that by controlling the structure and free energy of a RNA molecule, one may modulate its translation rate and half-life in a cell Buchan and Stansfield (2007); Davis et al. (2008), which is important in the context of viral RNA. A number of RNA folding algorithms Lorenz et al. (2011); Markham and Zuker (2008) use a thermodynamic model (e.g. Zuker and Stiegler (1981)) and dynamic programming to estimate MFE of a sequence. However, the $O(n^3)$ time complexity of these algorithms prohibits their use for evaluating substantial numbers of RNA sequences Gould et al. (2014) and exhaustively searching the space to identify the global free energy minimum, as the number of sequences grows exponentially as 4^n.

Here, we formulate the RNA sequence optimization problem as follows: For a sequence of length n, find the RNA sequence that will fold into the secondary structure with the lowest minimum free energy. In our experiments, we set $n = 30$ and use the popular RNAfold package Lorenz et al. (2011) to evaluate the MFE for a given sequence. The goal is to find the lowest MFE sequence by calling the MFE evaluator minimum number of times. The performance of different algorithms is depicted in Figure 2, where ECO outperforms the baselines as well as COMBO by a considerable margin.

In summary, the proposed black-box combinatorial optimization algorithm based on expert advice performs competitively or better than its state-of-the-art Bayesian counterparts, while reducing the computation time by multiple orders of magnitude.

References

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

J Ross Buchan and Ian Stansfield. Halting a cellular production line: responses to ribosomal pausing during translation. *Biology of the Cell*, 99(9):475–487, 2007.

Charles J. Colbourn and Jeffrey H. Dinitz. *Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2006.

Hamid Dadkhahi, Karthikeyan Shanmugam, Jesus Rios, Payel Das, Samuel Hoffman, Troy David Loeffler, and Subramanian Sankaranarayanan. Combinatorial black-box optimization with expert advice. In *To appear in KDD; preprint: arXiv:2006.03963*, 2020.

Matthew Davis, Selena M Sagan, John P Pezacki, David J Evans, and Peter Simmonds. Bioinformatic and physical characterizations of genome-scale ordered rna structure in mammalian rna viruses. *Journal of virology*, 82(23):11824–11836, 2008.

Neil Dixon, John N Duncan, Torsten Geerlings, Mark S Dunstan, John EG McCarthy, David Leys, and Jason Micklefield. Reengineering orthogonally selective riboswitches. *Proceedings of the National Academy of Sciences*, 107(7):2830–2835, 2010.

Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35, 2020.

Sébastien Gerchinovitz and Jia Yuan Yu. Adaptive and optimal online linear regression on $\ell1$-balls. In *Algorithmic Learning Theory*, pages 99–113. Springer Berlin Heidelberg, 2011.

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D Sculley. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1487–1495, 2017.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Nathan Gould, Oliver Hendy, and Dimitris Papamichail. Computational tools and algorithms for designing customized synthetic genes. *Frontiers in bioengineering and biotechnology*, 2:41, 2014.

Shuichi Hoshika, Nicole A Leal, Myong-Jung Kim, Myong-Sang Kim, Nilesh B Karalkar, Hyo-Joong Kim, Alison M Bates, Norman E Watkins, Holly A SantaLucia, Adam J Meyer, et al. Hachimoji dna and rna: A genetic system with eight building blocks. *Science*, 363(6429):884–887, 2019.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1 – 63, 1997.

Hui Li, Taek Lee, Thomas Dziubla, Fengmei Pi, Sijin Guo, Jing Xu, Chan Li, Farzin Haque, Xing-Jie Liang, and Peixuan Guo. Rna as a stable polymer to build controllable and defined nanostructures for material and biomedical applications. *Nano today*, 10(5):631–655, 2015.

Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.

Nicholas R Markham and Michael Zuker. Unafold. In *Bioinformatics*, pages 3–31. Springer, 2008.

Andrew H Ng, Taylor H Nguyen, Mariana Gómez-Schiavon, Galen Dods, Robert A Langan, Scott E Boyken, Jennifer A Samson, Lucas M Waldburger, John E Dueber, David Baker, et al. Modular and tunable biological feedback control using a de novo protein switch. *Nature*, 572(7768): 265–269, 2019.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial bayesian optimization using the graph cartesian product. In *Advances in Neural Information Processing Systems 32*, pages 2910–2920. Curran Associates, Inc., 2019.

Matthias Poloczek Ricardo Baptista. Bayesian optimization of combinatorial structures. In *ICML*, 2018.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha, and Gene E Robinson. Big data: astronomical or genomical? *PLoS biology*, 13(7):e1002195, 2015.

Audrey Terras. *Fourier Analysis on Finite Groups and Applications*. London Mathematical Society Student Texts. Cambridge University Press, 1999. doi: 10.1017/CBO9780511626265.

Ryota Yamagami, Mohammad Kayedkhordeh, David H Mathews, and Philip C Bevilacqua. Design of highly active double-pseudoknotted ribozymes: a combined computational and experimental study. *Nucleic acids research*, 47(1):29–42, 2019.

Michael Zuker and Patrick Stiegler. Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–148, 1981.