

A Thesis Submitted for the Degree of PhD at the University of Warwick

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/145731>

Copyright and reuse:

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk



**Efficient Pairwise Information Collection for Subset
Selection**

by

MATTHEW GROVES

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Mathematics Institute

Contents

List of Tables	iv
List of Figures	v
Acknowledgments	xi
Declarations	xii
Abstract	xiii
Abbreviations	xiv
Chapter 1 Introduction	1
1.1 Aims and Contributions	2
1.2 Thesis Outline	4
Chapter 2 Preliminaries	6
2.1 Ranking and Selection	6
2.2 Pairwise Ranking and Selection	10
2.2.1 Static Sampling	10
2.2.2 Active Sampling	11
2.2.3 Top- k Selection	12
Chapter 3 Pairwise Top-k Selection	16
3.1 Introduction	16
3.2 Problem Definition	16

3.3	Algorithm Details	18
3.3.1	Pairwise Optimal Computing Budget Allocation (POCBAm)	18
3.3.2	Pairwise Knowledge Gradient (PKG)	22
3.3.3	Asymptotic Correctness	26
3.4	Empirical Testing	32
3.4.1	Top- k Selection (2 from 5)	33
3.4.2	Other System Sizes	37
3.4.3	Value-based Scoring Models	38
3.5	Summary	42
Chapter 4 Applications to Evolutionary Strategies		43
4.1	Introduction	43
4.2	Tackling Noise in Fitness Evaluation	44
4.3	CMA-ES	45
4.4	Synthetic Functions	48
4.4.1	Test functions	48
4.4.2	Results for Single Generation Selection	51
4.4.3	CMA-ES Performance Across Multiple Generations	52
4.5	Evolving Poker Playing Agents	66
4.5.1	Texas Hold'em poker	66
4.5.2	Poker player model	68
4.5.3	Experiments	69
4.6	POCBAm With Different Objectives	75
4.7	Summary	78
Chapter 5 Exploiting Transitivity		79
5.1	Introduction	79
5.2	ML-POCBAm	80
5.3	Empirical Testing	85

<i>CONTENTS</i>	iii
5.3.1 Value-based Thurstone model	87
5.3.2 Analyzing SELECT/TOP	88
5.4 Noise Perturbed Thurstone model	91
5.5 Top- k Selection of Poker Players	96
5.6 Summary	100
Chapter 6 Conclusions	102
6.1 Future Work	105
Appendix A Proofs From Chapter 3	109
Appendix B Kullback-Leibler Knowledge Gradient	114
B.1 Model	115
B.2 Algorithm	117
B.3 Empirical Testing	121
B.3.1 Single Generation	121
B.3.2 Multiple Generation Optimization	126
B.4 Summary and Future Work	128

List of Tables

3.1	The Pairwise Optimal Computing Budget Allocation for Subset Selection Procedure	23
3.2	The Pairwise Knowledge Gradient Procedure	27
3.3	Percentage reduction in sampling budget to match performance of uniform sample allocation for each sampling method. Best values shown in bold.	40
4.1	Input features used in the OFRE and SWRE networks (as used in [64]).	69
4.2	Description of the fixed benchmark opponents	72
4.3	Average performance difference between the final POCBAm and uniform evolved players against each of the five fixed benchmark players. P values are the result of a two-sided t-test.	73
5.1	The Maximum Likelihood POCBAm Procedure	86
B.1	Kullback-Leibler Knowledge Gradient (KL-KG)	120
B.2	Statistical comparison of performance. Standard errors for the difference between the means are given in parenthesis, P values show the result of a two-sided T-test.	126

List of Figures

3.1	Illustration of the expected effect on the approximated posterior score distributions due to allocating a sample to the pair (a_i, a_j) . The expected post-sample distributions $\tilde{S}_i^{i,j}$ and $\tilde{S}_j^{i,j}$ are narrower, increasing the probability mass lying on the “correct” side of the threshold c and thereby increasing AEPCS.	21
3.2	Performance of POCBAm, PKG, AR and H-Race algorithms against random allocation at best 2 of 5 selection for the BTL, SST and unstructured models. For POCBAm, we vary α between 0.5 and 0.01, for PKG between 0.3 and 0.001 and for AR between 0.15 and 0.01. For the H-Race, n_{max} ranges between 5 and 100 for the BTL and SST models, and between 5 and 450 for the Unstructured model, with $\alpha = 0.01$ for all three.	36
3.3	Performance of POCBAm, PKG, AR and H-Race algorithms against random allocation at best 1 of 5 selection (a), best 4 of 10 (b), and best 40 of 100 selection (c), for the SST scoring model. For POCBAm, we vary α between 0.5 and 0.01, for PKG between 0.3 and 0.001 and for AR between 0.2 and 0.01. For the H-Race, (a) uses $\alpha = 1.0$, n_{max} between 5 and 100, (b) uses $\alpha = 0.01$, n_{max} between 5 and 50, and (c) uses $\alpha = 0.01$, n_{max} between 10 and 25.	38

3.4	Performance of POCBAm, PKG, AR and H-Race algorithms against random allocation at best 2 of 5 selection with normally distributed sample results for SST (a) and unstructured (b) models. Sub-figure (a) uses α between 0.5 and 0.01 for POCBAm, between 0.05 and 10^{-5} for PKG and between 0.15 and 0.01 for AR with n_{max} between 5 and 80, and $\alpha = 1.0$. Sub-figure (b) uses α between 0.3 and 0.001 for POCBAm, between 0.03 and 10^{-5} for PKG, and between 0.15 and 0.001 for AR with th n_{max} between 5 and 2500, and $\alpha = 0.01$	41
4.1	The Sphere function. Global minimum at $[0, \dots, 0]$ marked by cross.	48
4.2	The Ackley function	49
4.3	The Rosenbrock function	50
4.4	The Rastrigin function	51
4.5	Performance of POCBAm and uniform sample allocation at pairwise top-3 of 6 selection of the initial population randomly generated for CMA-ES for different 2D test functions.	53
4.6	Example of the convergence of the CMA-ES optimizer on the 2D Sphere function with noisy pairwise sampling. Samples chosen using POCBAm, with the selected top- k individuals for each generation highlighted in orange. Global optimum highlighted by x.	55
4.7	Convergence of CMA-ES on the 2D Rosenbrock function with noisy pairwise samples selected using POCBAm. We see that the optimizer is able to identify the near-optimal trough, before steadily traversing towards the global optimum. In this figure every only 3rd generation is shown.	56
4.8	Convergence of CMA-ES on the 2D Rastrigin function with noisy pairwise samples selected using POCBAm.	57

4.9 Generally stochastic optimization methods like CMA-ES should be more resilient to local optima than gradient-based methods. In this figure, we see an example of a case when CMA-ES becomes trapped in a local minimum. In generation 5 POCBAm selects 3 points that are very close to each other, greatly shrinking the variance of the recombination distribution. Without sufficient variability in the subsequent generations, CMA-ES was unable to escape the pull of a nearby local minimum. 58

4.10 Performance of CMA-ES on the Sphere function with random top- k selection. On the y-axis, Error is the difference between objective function value at the mean of the optimizer’s distribution and the global optimum. 59

4.11 Performance of CMA-ES with POCBAm and uniform sample selection over 50 generations on the pairwise 2D sphere function. Subfigures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Subfigure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum. The “Oracle” method in subfigure (c) shows CMA-ES performance with perfect, noise-less fitness evaluation. 61

- 4.12 Performance of CMA-ES with POCBA_m and uniform sample selection on the pairwise 2D Ackley function. Sub-figures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Sub-figure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum, with subfigure (d) showing the same result plotted on a log scale. As before, the “Oracle” method shows CMA-ES performance with perfect, noise-less fitness evaluation. 63
- 4.13 Performance of CMA-ES with POCBA_m and uniform sample selection on the pairwise 2D Rosenbrock function. Sub-figures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Subfigure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum. The “Oracle” method in sub-figure (c) shows CMA-ES performance with perfect, noise-less fitness evaluation. . . . 64
- 4.14 Performance of CMA-ES with POCBA_m and uniform sample selection on the pairwise 2D Rastrigin function. Sub-figures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Subfigure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum. The “Oracle” method in sub-figure (c) shows CMA-ES performance with perfect, noise-less fitness evaluation. . . . 65

4.15 Performance of the evolved players from each of the 20 CMA-ES runs against the final 10 players from the 2000th generation of the CMA-ES runs that used POCBAm sampling. Y-axis scale is in average Big Blinds won per Hand (BB/H), averaged over 1000 hands per pair. 71

4.16 Performance of the evolved poker players for each sampling method against each of the 5 benchmark players, average over the 10 replications for each method, with each combination of benchmark and evolved player playing each other for 1000 hands. Subfigure (a) shows the results for POCBAm and subfigure (b) shows the results for uniform sampling. Subfigure (c) shows the difference in performance for the POCBAm and uniform evolved players. Scale is in Big Blinds won per Hand (BB/H). 74

5.1 Performance of ML-POCBAm, POCBAm, SELECT/TOP and uniform sample allocation on top 4 of 10, top 1 of 10 and top 40 of 100 selection with a Thurstonian latent preference model. 89

5.2 Performance of ML-POCBAm, uniform sampling and SELECT/TOP on top 1 of 100 selection with a Thurstonian latent preference model and a large separation between the quality value of the top alternative and the rest of the population. Here $\gamma_0 = 1.2$, with γ_i selected uniformly at random from $[0, 1]$ for all other alternatives. 91

5.3 Sample pairwise preference matrices for populations of 10 alternatives generated according to the noise perturbed Thurstonian model described in 5.4 with different values of d . Alternatives are indexed by Borda score. We can clearly see this increasing degree of pairwise intransitivity as d increases. For an example of an intransitive triplet, consider (a_0, a_1, a_2) for $d = 0.4$, we have $\mu_{0,1} > 0, \mu_{1,3} > 0, \mu_{0,3} < 0$ 94

5.4	Performance of ML-POCBAM and POCBAM on top 4 of 10 selection problems with noise disturbed Thurstone preference with various different degrees of noise.	95
5.5	Subfigure 5.5a shows relative performance of ML-POCBAM and POCBAM after 250 samples for different values of d , with estimated intransitivity index values for each different d shown in 5.5b. 5.5c shows how the empirically estimated intransitivity index changes as more samples are taken for different d values.	97
5.6	Visualization of the pairwise preference matrices of the ten randomly generated populations of poker players used in Section 5.5. Players are indexed in true ranking order.	99
5.7	Performance of ML-POCBAM, POCBAM and uniform sampling on top 4 of 10 selection for each of the populations of poker players shown in Figure 5.6.	101
B.1	Pairwise difference in K-L divergence for KL-KG against standard KG and Uniform sample allocation for sampling budgets between 20 and 400. Averaged over 10,000 repetitions.	124
B.2	Pairwise difference in K-L divergence for KL-KG compared to standard KG and Uniform sample allocation for different noise levels. $N = 200$	125
B.3	Performance of CMA-ES optimization for each function and sampling method over 40 generations. Error refers to the difference between a function's value evaluated at the mean of the CMA-ES distribution and the global minimum.	127

Acknowledgments

This thesis has been a long and challenging journey, which I could not have completed without the support and guidance of many amazing people I have been lucky enough to meet along the way.

Firstly, I'd like to express my sincere gratitude to Professor Juergen Branke, who has been a brilliant mentor and had helped to guide this project from start to finish. I cannot imagine a better research supervisor.

Special thanks to my colleague and best friend Zhangdaihong Liu, for her wonderful caring nature, her companionship, her advice and for her delicious cooking; and to my parents, Richard and Lesley, for the many years of love and support they have given me and for their patience, especially while I was writing up.

Additionally I'd like to thank Professor Walter Gutjahr for his input on exploiting pairwise transitivity, Dr. Jeremy Reizenstein for his advice and support during the years I spent living in Canley, and Michael Pierce for many interesting and illuminating discussions and for his enthusiasm and collaboration on several projects not included in this work.

I am hugely grateful for the funding I received from the Mathematics for Real World Systems CDT, The Engineering and Physical Sciences Research Council and the Association for Computing Machinery that made it possible not only to embark on this PhD, but also to travel to several interesting conferences along the way.

Declarations

This work has been composed by myself and has not been submitted for any other degree or professional qualification.

- Part of the work in Chapter 3 has been published in the following conference paper: **Groves, M.**, and Branke, J., (2016). Optimal subset selection with pairwise comparisons. In *Proceedings of the DA2PL'2016 EURO Mini Conference*, pp. 15-20. EURO.
- The majority of the work in Chapter 3 has been published in the following paper: **Groves, M.**, and Branke, J. (2019). Top- κ selection with pairwise comparisons. In *European Journal of Operational Research*, 274(2), pp. 615-626. Elsevier.
- The work in Chapter 5 has been submitted for publication to the Journal of Machine Learning Research.
- The work in Appendix B on integrating KG into CMA-ES has been published as the following conference paper: **Groves, M.**, and Branke, J. (2018, July). Sequential sampling for noisy optimisation with CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2018)* pp. 1023-1030. ACM.
- This research utilised Queen Mary University's Apocrita HPC facility, supported by QMUL Research-IT, as well as the Tinis HPC facility at the University of Warwick Centre for Scientific Computing.

Abstract

In this work, we consider the problems of selecting the subset of the top- k best of a set of alternatives, where the fitness of alternatives must be estimated through noisy pairwise sampling. To do this, we propose two novel active pairwise sampling methods, adapted from popular non-pairwise ranking and selection frameworks. We prove that our proposed methods have desirable asymptotic properties, and demonstrate empirically that they can perform better than current state-of-the-art pairwise selection algorithms on a range of tasks. We show how our proposed methods can be integrated into the Covariance Matrix Adaptation Evolutionary Strategy, to improve fitness evaluation and optimizer performance including in the evolution of neural network based agents for playing No Limit Texas Hold'em poker. Finally, we demonstrate how parametric models can be used to help our proposed sampling algorithms exploit transitive preference structure between alternative pairs.

Abbreviations

AR: Active Ranking method
AEPCS: Approximated Expected Probability of Correct Selection
BB/H: Big Blinds per Hand
BTL: Bradley-Terry-Luce (model)
CLT: Central Limit Theorem
CMA-ES: Covariance Matrix Adaptation Evolution Strategy
DB: Dueling Bandit
EA: Evolutionary Algorithm
ES: Evolutionary Strategy
EPCS: Expected Probability of Correct Selection
HOF: Hall of Fame
II: Intransitivity Index
KG: Knowledge Gradient
KL: Kullback Leibler (divergence)
KL-KG: Kullback-Leibler (minimizing) KG
MAB: Multi-Armed Bandit
ML-POCBAm: Maximum Likelihood POCBAm
NLTH: No limit Texas Hold'em
OCBA: Optimal Computing Budget Allocation
OCBAm: Optimal Computing Budget Allocation for subset selection
OFRE: Opponent Fold Rate Estimator
OppC: Opportunity Cost
PAC: Probably Approximately Correct
PBR: Preference Based Racing

ABBREVIATIONS

xv

PCS: Probability of Correct Selection

PD: Pairwise Distinguishability

PKG: Pairwise Knowledge Gradient

POCBAm: Pairwise OCBAm

PRT: Pattern Recognition Tree

RV: Random Variable

SST: Strong Stochastic Transitivity

SWRE: Showdown Win Rate Estimator

w.r.t: with respect to

CHAPTER 1

Introduction

Top- k selection is a well known problem, with applications in many different areas, including player ranking in games, selection in evolutionary algorithms, optimizing search engine result relevance, and preference elicitation in decision making or other social contexts. It is a more general version of the common top-1 problem, where the task is only to find the single best of a set of alternatives.

In the standard top- k problem, the score or “quality” of each of K possible alternatives is modeled by the expectation of a real-valued random variable, a statistic estimated through repeated sampling. The setting of the problem can be static or active: a set of sampling results may be provided a priori, or the ranker may be allowed to sequentially select which alternatives to sample as the algorithm progresses. The aim is to efficiently and accurately select the best subset of given size k from the set of alternatives. In the active setting, the total number of samples available is generally restricted, giving rise to an optimization problem, with the objective of devising sampling procedures to maximize the probability of correctly selecting the highest scoring alternative or subset of alternatives within the sampling budget constraint.

However, in many real world applications it can be difficult or impractical to directly estimate an alternative’s quality through sampling. Instead, it may only be possible to obtain pairwise information, either in the form of a numerical value, or

as a binary preference, expressing the result of a comparison between two items. For a motivating example, consider the ranking of two football teams; it is unclear how one might accurately assess the strength of each team in isolation, but by playing the teams against each other and recording the result, we obtain pairwise information that can be translated into a ranking. Thus, we consider an adaptation of the standard top- k selection problem that restricts the sampling process to allow only pairwise comparisons between alternatives. Here, rather than modeling the score of an alternative as a random variable that can be sampled directly, we instead treat the outcome of each possible pairwise comparison between alternatives as a random variable (R.V.). The score of an alternative is then considered to be the sum of the expectations of the $K - 1$ R.V.'s for the pairwise comparisons with all other alternatives.

This sampling restriction increases the complexity of the problem. In general, the number of individual samples required to obtain a single measurement of the score of all possible alternatives increases from K to $\frac{K(K-1)}{2}$. In addition, the outcomes of the pairwise comparisons need not be transitive. For example, in the context of game players, differences in playing styles and counter strategies might create cycles in pairwise performance (A beats B, B beats C, C beats A). The information gained from a particular pairwise comparison against a particular opponent thereby only relates to part of an alternative's overall quality, leading to additional complications when attempting to optimize the sampling process.

1.1 Aims and Contributions

This thesis focuses on pairwise active sampling methods for the problem of top- k selection. Much of the past work on pairwise learning methods only seek to address the less general top-1 case, or are only applicable to binary pairwise sample outcomes, or are dependent on strong assumptions on the preference structure between

alternatives. Our primary research aim in this work was to address these restrictions by proposing pairwise sample selection methods that could perform well for any subset size, with either preference-based (binary) or quantitative sampling results and be resilient to inconsistency or intransitivities in the pairwise relationships between alternatives. Our specific contributions are as follows:

1. We propose two novel active sample allocation methods for top- k subset selection, adapted for pairwise sampling problems from the well-known Optimal Computing Budget Allocation and Knowledge Gradient frameworks.
2. We prove that the proposed methods are asymptotically correct under certain conditions.
3. We empirically investigate the performance of the proposed sampling methods in various settings, and compare against current state-of-the-art pairwise subset selection methods.
4. We integrate the best performing of these methods into the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), a state-of-the-art Evolutionary Strategy to improve fitness evaluation in pairwise cases where fitness evaluation is affected by noise.
5. We demonstrate empirically that using our sampling method to improve the quality of selection in CMA-ES can lead to better evolved solutions, including the evolution of artificial Poker players.
6. We propose a simple parametric model for pairwise ranking with quantitative sample outcomes and an adapted version of our sampling method that can exploit transitivity between using the parametric model.

1.2 Thesis Outline

This thesis proceeds as follows:

- In Chapter 2, we discuss the related work from the general ranking and selection literature, in particular active learning methods and methods for top-1 and top- k selection. We then discuss pairwise methods, especially the closely related work on Dueling Bandits and give an overview of several important concepts in pairwise ranking and selection. This chapter includes an introduction to two popular Bayesian sampling frameworks from the Simulation Optimization literature – The Knowledge Gradient method and the Optimal Computing Budget Allocation method, upon which much of the work in this thesis is based.
- In Chapter 3 we provide a formal definition of the pairwise top- k sampling problem. We adapt both the KG and OCBA frameworks for pairwise sampling, proposing two new active top- k selection methods, PKG and POCBAm. We provide a proof of asymptotic correctness for POCBAm, and proofs that PKG is asymptotically correct when the pairwise sample outcomes are unbounded, but due to an interesting peculiarity of pairwise sampling, it can fail with bounded sample results. We then test the performance of both methods on a range of synthetic top- k selection problems, showing that POCBAm significantly outperforms previously published methods and uniform sampling.
- In Chapter 4, we consider a natural application for our pairwise top- k selection methods in improving selection in Evolutionary Algorithms (EAs) where fitness evaluation is affected by noise. Prior work has suggested using efficient fitness evaluation methods like OCBA for noisy, non-pairwise sampling, but pairwise cases, which may commonly arise in the evolution of game players or other co-evolutionary applications have not yet been considered. We give

an overview of the popular Covariance Matrix Adaptation Evolution Strategy (CMA-ES), a state-of-the-art evolutionary method, and show how POCBAM can easily be integrated into CMA-ES to improve selection in pairwise problems. We show that POCBAM-based fitness evaluation can improve the quality of the evolved solutions on both a range of synthetic test functions and on an interesting co-evolutionary task of evolving Artificial Neural Network players for two-player Texas Hold'em Poker.

- Chapter 5 considers how to improve the performance of the POCBAM method in cases where pairwise sample outcomes are highly transitive across sets of alternatives. Unlike the POCBAM method proposed in Chapter 3, many Dueling Bandit methods assume parametric models for alternative fitness that can be exploited for more efficient sample choices. Concentrating on the case of quantitative, unbounded sampling outcomes, we adapt a simple parametric ranking model, and propose a version of POCBAM that improves selection accuracy by fitting the model parameters using sampling data. We return to the poker player selection example from Chapter 4, and show that the new ML-POCBAM method outperforms standard POCBAM at selecting the top subsets from groups of randomly generated players.
- Finally, Chapter 6 summarizes the thesis, its contributions and discusses possible future work.

CHAPTER 2

Preliminaries

In this chapter we discuss methods related to subset selection and pairwise information collection. We start by considering approaches to the standard top-1 and top- k selection problems within the *Machine Learning* and *Simulation Optimization* literature, before giving an overview of pairwise ranking and selection methods and pairwise top- k methods in particular.

2.1 Ranking and Selection

In the Machine Learning community, top- k selection sits within the set of Multi-Armed Bandit (MAB) problems. MAB problems are an active area of research that consider the online learning of optimal decision alternatives from a given set, using information obtained from sampling. In the standard MAB problem, named after an imagined multi-armed bandit casino machine, a decision maker sequentially acts by selecting an alternative from an available set (pulls a particular arm) and receives a noisy reward signal in return. Depending on the values of the rewards received, and the objective of the decision maker, they then select the next alternative to sample and repeat the process. The MAB literature is broadly divided into three areas, each concerned with a different decision maker objective. The first and largest branch considers the *regret minimization* problem, whereby the sampler must minimize the total regret incurred by sampling non-optimal alternatives during the

sampling process, typically defined as the cumulative difference between the reward received from the selected alternative in each sample, and the best possible reward from an alternative. The decision maker must choose samples to learn about the fitness of different alternatives, thus leading to better future sample choices, whilst simultaneously preferentially sampling alternatives that appear better to keep total regret low. This trade off between exploration and exploitation is the key feature of the regret minimization case. For example, in personalized medicine, a physician may try a variety of different treatments in order to identify the most effective. In each trial, the physician must balance the gain of learning more about a different treatment's effects (and thus perhaps finding a cure), with the regret of applying sub-optimal (or even harmful) treatments. [7], for example uses a regret minimizing MAB framework for learning appropriate Warfarin dosages for different patients. Another regret minimization application could be in retail pricing, where a retailer may wish to set the price of a product to maximize profit, but want to avoid lost sales due to testing incorrect prices [72]. The regret minimization problem is typically only concerned with top-1 alternative identification as so long as the decision-maker's sample selection process is guaranteed eventually to identify the correct arm, finite asymptotic upper bounds on regret can be constructed.

The second branch aims to solve the *simple regret* or *pure exploration* problem. In this case, it is assumed that there is an initial period in which a fixed number (possibly unknown to the decision maker) of samples can be taken, after which the decision maker must recommend their estimate of the best (or set of k best) alternatives. The *simple regret* incurred is just the difference in quality between the true best and recommended alternatives. For example, in consumer product testing, testers may be shown a variety of product alternatives, with their feedback being used to identify the best alternative to release commercially. In this case, the regret during the testing phase is irrelevant compared to making the best possible choice for the final product.

The third branch is the *Probably Approximately Correct* MAB problem. This problem is closely related to pure exploration, but instead of a fixed limit on the sampling budget for the exploration phase, the aim is to minimize the number of samples required to identify an ϵ -optimal alternative (or subset of alternatives) to a pre-specified degree of confidence.

The *Simulation Optimization* community has developed several alternative approaches to ranking and selection, as discussed in [14]. In particular, the authors describe in detail two classes of Bayesian methods; expected Value of Information Procedures (VIP) and the Optimal Computing Budget Allocation (OCBA).

OCBA refers to a group of procedures first proposed in [24], and further developed in [27] and [28]. In [25], the authors adapt a version of OCBA for optimal subset selection. The problem considered in this work is the classic selection problem, where allocating a simulation run corresponds directly with sampling the score of an alternative. The OCBA framework is an active sampling framework that seeks to improve selection efficiency by allocating samples preferentially to critical alternatives, typically seeking to maximize the Probability of Correct Selection (PCS). Using sampling results collected for each alternative, we can construct distributions that estimate our uncertainty for each alternative's fitness or score. This in turn allows us to estimate the probability that the top- k alternatives we currently consider to be best truly have higher fitness than the rest, under the assumption that alternative scores are indeed distributed according to these empirical score distributions. That is, given current top- k subset estimate \mathcal{I} , and for each alternative a_i , we have a sample from the alternative's score distribution s_i , PCS is defined as:

$$PCS \equiv \bigcap \mathbb{P}\{s_i > s_j \text{ for } a_i \in \mathcal{I}, a_j \notin \mathcal{I}\} \quad (2.1)$$

By considering the expected effect of allocating additional samples to the alternative score distributions, OCBA recommends the next sample or set of sam-

ples that should be taken, either by deducing the optimal proportional allocation of the sampling budget that asymptotically maximizes PCS [27], or by myopically selecting the next sample to maximize the expected increase in PCS resulting from the sample [29]. OCBA is well suited to stochastic simulation optimization problems, as considering the distribution of sampling outcomes for alternatives allows it to efficiently identify which alternatives are critical for making correct selection decisions [26]. As such, it seems reasonable to expect OCBA could be well applied to the noisy pairwise problem. The OCBA framework can be seen as fitting within either the second or third branches of the MAB literature, it is purely exploratory in that it only bases sampling decisions on the expected information that can be obtained from the sample, rather than considering the regret incurred from the actual rewards gained during sampling.

A popular variant of the VIP approach is the Knowledge Gradient (KG) policy first proposed in [46] and developed in [40] and [32]. The KG policy sequentially samples alternatives based on myopically optimizing the expected value of information gained by performing a single additional sample. On the top- k selection problem, this assumption that the sampling process will terminate after a single additional sample means that the final sample only provides value if it changes the estimated top- k subset. KG therefore estimates the probability of receiving a sample outcome that changes the alternative score estimates enough to alter the top- k subset, and samples where this probability is maximized. [40] demonstrates that the KG policy is able to perform efficiently where sample measurements are normally distributed. [59] identifies potential limitations of the KG for discrete measurement cases, proposing adapted sample selection methods demonstrated to improve performance in the Bernoulli case. We have reported some preliminary investigation on using OCBA and KG in the context of pairwise comparisons in [43]. Independent of our work, [76] show empirically that KG works better than Equal allocation, but can get stuck in a pairwise comparison setting.

2.2 Pairwise Ranking and Selection

2.2.1 Static Sampling

There is a wide variety of research related to ranking problems based on pairwise information. A number of works [17, 74, 48] present approaches for generating a complete ranking on static problem cases. For game players in particular, there are also some well known methods for extracting player rankings from given sets of pairwise comparison outcomes like the ELO ranking [34] or TrueSkill ranking [45] which have been widely applied to provide player rankings for a variety of games including Baseball, Chess, Go, and Xbox™ gamers. While any method that produces a complete ranking can obviously also be used to identify only the top subset of alternatives, such methods are unlikely to be as effective as those specifically designed for this purpose. For top- k selection specifically, a major approach is the class of spectral ranking methods based on *Rank Centrality*, notably the *Spectral MLE* algorithm proposed in [31] and further analyzed in [55]. Both consider sets of alternatives with underlying (true) preferences based on the popular *Bradley-Terry-Luce* (BTL) model [13]. This model assumes the existence of an underlying (unknown) vector of weights, (w_1, \dots, w_K) that parametrizes the true preferences over the set of alternatives a_1, \dots, a_K . These weights are assumed to determine the probability of each outcome of a pairwise comparison between two alternatives: specifically, in a comparison between alternatives a_i and a_j , the probability that a_i wins is given by $\frac{w_i}{w_i + w_j}$. As this model considers only win/loss comparison probabilities, these methods are limited to considering the preference-based top- k selection case.

Recently, [91] further developed *Rank Centrality* and *Spectral MLE* methods for top- k selection under the BTL model to include “adversarial” settings, where a portion of sample results are deliberately falsified. This is designed to make the methods more robust to real world effects, for example, to the effect of spammers and manipulation of internet survey results.

Although popular, the strong parametric assumptions of the BTL model often fail in real-world applications (see, for example, [5]). These limitations are discussed in detail in [86]. In that work, the authors suggest the class of *Strongly Stochastically Transitive* (SST) models first defined by [37] to be more consistent with experimental data. This more general class of models is based on the assumption of the SST condition, stated here for binary comparison outcomes:

Definition 2.2.1. *Strong Stochastic Transitivity condition.* Given alternatives a_i , a_j and a_k with pairwise comparison means $\mu_{i,j}$, $\mu_{j,k}$ and $\mu_{i,k}$, then:

$$\mu_{i,j} \geq \frac{1}{2} \text{ and } \mu_{j,k} \geq \frac{1}{2} \implies \mu_{i,k} \geq \max\{\mu_{i,j}, \mu_{j,k}\}$$

The class of SST models includes the BTL model, as well as other well known parametric models such as the Thurstone model [93]. [87] explore methods both for complete ranking and top- k selection for SST models, proposing a simple and computationally efficient counting algorithm based on the Copeland score of each alternative. [30] also look at static top- k selection with the SST model, presenting a counting algorithm with adaptations to better account for the varying importance of different sample results.

2.2.2 Active Sampling

Where possible, it is often advantageous to actively choose which pairwise comparisons to perform, by sampling sequentially and taking account of previous sample outcomes to choose more relevant or informative pairs to compare. A range of active sampling methods exist that attempt to capture this benefit. For complete ranking, see [20] for Mallows models, or [71] which discusses *Rank Centrality* and *QuickSort* based approaches for the BTL model. Closely related is the so called *Dueling Bandits* problem described in [97] and [98] for finding the best single al-

ternative through active pairwise sampling, using an underlying sampling model based on the SST condition. [95] proposes the SAVAGE algorithm, a more general dueling bandit method for identifying the top element without the stochastic transitivity assumption. For a good overview of pairwise learning methods in the context of bandit algorithms, see [19]. There are two key differences between the Dueling Bandit methods and the work in this thesis: firstly, like regret minimizing MAB methods, most dueling bandit algorithms focus solely on top-1 identification and cannot readily be generalized to top- k selection. Furthermore, like the BTL-based Rank Centrality and Spectral MLE methods, dueling bandit methods concentrate on the preference-based sampling cases, where the outcomes of pairwise samples are binary, whereas we consider methods for both preference-based and quantitative samples.

2.2.3 Top- k Selection

One approach for active top- k selection allowing cyclical preferences is through the use of *Successive Elimination* or *Racing* algorithms, first introduced in [68] and [69] for top-1 selection and [53] for top subset selection. These iterative methods provide a framework for dealing with sampling uncertainty, designed to replicate a race. During the sampling process, as the quantity of information about each alternative increases, particularly well-performing alternatives can be allowed to “finish early” and are selected, while those that lag behind are eliminated, with further sampling focused solely on alternatives remaining in the race. Although the standard racing implementation includes the idea of a maximum budget, the proportion of this budget utilized by the race is usually variable, as the algorithm terminates upon reaching a solution set with desired size, having successively eliminated alternatives during the sampling process based on a probabilistic bound based on an accuracy parameter α . Fixed budget adaptations of the racing framework (see for example [15]) do exist, aiming to adaptively tune the accuracy parameter α to maximize

performance within a given budget constraint.

Racing has been applied to a variety of contexts including model selection and parameter tuning. [53] describes in detail the *selectRace* procedure for obtaining the best μ of λ alternatives, using both the Hoeffding and empirical Bernstein bounds. For pairwise selection problems, [21] presents a preference-based racing (PBR) method for active top- k selection. The objective of the PBR method is to identify the top- k subset with probability at least $1 - \delta$ while minimizing the number of samples taken. To do this, it maintains an active subset of alternative pairs that are all sampled at each iteration, maintaining estimates of the win probabilities of each pair based on sampling data. When a sufficiently high degree of confidence is obtained about the pairwise sampling means of an alternative (to know with high probability whether it is in the top- k or not), the alternative is eliminated from the active set and no more samples are allocated to it, thus reducing the sampling complexity. Like in [53], the degree of confidence for pairwise win rates is estimated by constructing confidence intervals based on the Hoeffding Bound [54]. In the paper, the authors present sampling strategies for 3 different ranking methods: the Copeland ranking, the random walk ranking and the sum of expectations (Borda score) ranking. While the former two are only applicable for binary comparison outcomes, the PBR sum of expectations sampling strategy is more general and can apply to any numerical sampling case. However, the authors note that in cases where the support of the distribution of pairwise sample outcome is substantially greater than the variance, using the Empirical Bernstein bound [3] to construct confidence intervals may provide better performance.

[52] also suggest a racing-like sampling method for both top- k selection and total ordering of alternatives by Borda score estimation. In contrast to the PBR method, their Active Ranking (AR) method compares each alternative still in the race to a single randomly chosen opponent at each iteration, using the sample results to update alternative score estimates. They construct bounds around these

estimates, again based on the Hoeffding bound, and use these to partition the alternatives into the top set and remainder. To achieve a full ranking, they use the same procedure, with $K - 1$ rather than a single partition. To our knowledge, [52] represents the current state of the art for subset selection using pairwise comparisons with sampling uncertainty and without parametric models like BTL or regularity assumptions such as Stochastic Transitivity.

However, some methods do seek to use such assumptions to improve selection efficiency. For example, [73] consider active top- k selection and top- k ranking (where the top- k subset must be returned in rank order) for preference-based dueling bandits using a 2-step sampling procedure. They first propose the SELECT algorithm for finding the top-1 alternative. SELECT uses a single-elimination tournament with repeated comparisons to counteract noise. They then generalize to the TOP algorithm for top- k ranking by dividing the entire set of alternatives into k sub-populations, and applying SELECT to find the best alternative in each. This shortlist of alternatives is then ranked and stored in order and the top alternative of the ranked shortlist is selected and removed from the list. To find the next best, only the sub-population to which the selected alternative originally belonged is re-sampled, and the new top alternative from this sub-population inserted into the ranked shortlist. This process is repeated until k alternatives have been selected. SELECT/TOP assumes a total ordering over alternatives, with every pairwise preference (i.e the sign of $\mu_{i,j}$) corresponding correctly to the true alternative ranking. The single elimination tournament SELECT phase exploits this assumption by allowing alternatives to be removed from consideration based on results against only a single peer. This results in good sampling budget scaling with the number of alternatives ($\mathcal{O}(n \log n)$), but means that even a single pairwise intransitivity could make it impossible for the method to find the exact top- k , even when the number of replications per comparison in the tournament phase tends to infinity. In the paper, the authors present bounds on the samples required by the TOP method, and quan-

tify the gain in terms of reduced sampling complexity of using active sampling over passive methods for different noise models. They compare the performance of SELECT/TOP against the AR method of [52] and show superior performance on both top-1 and top- k selection. One key difference between the SELECT/TOP method and the sampling procedures we propose in this thesis is that the former does not account for the sampling variance of each pair, instead using a fixed number of replications each time. As well as the difficulty for the user of setting an appropriate setting for this replication number parameter a priori, this is potentially inefficient, leading to over sampling of low variance pairs, and insufficient allocation where sample variance is high.

CHAPTER 3

Pairwise Top- k Selection

3.1 Introduction

In this chapter, we give a formal definition of the pairwise top- k selection problem, the main problem that we seek to address in this Thesis. We then discuss how both the Optimal Computing Budget Allocation (OCBA) and Knowledge Gradient (KG) sample selection methods may be adapted for pairwise sampling, proposing an efficient sample selection procedure for each framework. In Subsection 3.3.3, we consider the asymptotic properties of each method and prove the asymptotic correctness of Pairwise OCBA and, under certain conditions, Pairwise KG. We test the empirical performance of the two methods against various other pairwise top- k selection sampling methods.

3.2 Problem Definition

The problem we consider is a variation of the standard active “Top- k Selection” problem. Suppose we are presented with a finite set of K possible alternatives $\mathcal{A} = \{a_1, a_2, a_3, \dots, a_K\}$ and to each possible pair of alternatives (a_i, a_j) , there is associated a random variable $X_{i,j}$ with unknown finite mean $\mu_{i,j}$, representing the expected outcome of a “pairwise comparison” of alternative a_i against alternative a_j . The quality S_i of an alternative a_i is determined by the sum of the means of the

R.V.'s corresponding to pairwise comparisons of a_i against all other alternatives:

$$S_i = \sum_{j \neq i} \mu_{i,j}$$

This is commonly known as the *Borda Score* [12]. We assume that comparing an alternative a_i to a_j has the same effect as comparing a_j to a_i . Thus, the random variables $X_{i,j}$ are paired, with $X_{i,j} = -X_{j,i}$ for value-based sample results and $X_{i,j} = 1 - X_{j,i}$ for binary preference samples. As such, performing a pairwise comparison of two alternatives will affect the estimates of both of their scores.

This model of defining alternative fitness using the Borda scores of their pairwise comparisons does not explicitly assume the existence of any underlying latent value model for alternatives. However, under reasonable conditions, namely Stochastic Transitivity (Definition 2.2.1) and Pairwise Distinguishability, the Borda score ranking for alternatives will be identical to the underlying latent ranking, should one exist. For a proof of this property, see Appendix A. It is also important to note that the Borda Score only considers the expected performance of an alternative, not the variability of this performance.

The aim is to identify the index set $\mathcal{I} \subset [K]$ of given size k containing the highest scoring alternatives, i.e, the solution to the optimization problem:

$$\operatorname{argmax}_{\mathcal{I} \subset [K]: |\mathcal{I}|=k} \sum_{i \in \mathcal{I}} S_i$$

This can be done by iteratively selecting pairs of alternatives (a_i, a_j) and sampling $X_{i,j}$, thereby improving the quality of our estimates of the $\mu_{i,j}$'s that comprise the alternative's scores. In particular, we are interested in cases where the sampling process is deemed "expensive", either computationally, or due to the need for real-world interactions, and hence the number of samples we can take is limited. The problem becomes how to iteratively select the next pair to sample to maximize the probability of correctly identifying the optimal subset.

3.3 Algorithm Details

In Chapter 2, we identified two possible sampling policies that can be adapted to address the problem defined in Section 3.2. In this section, we discuss them in more detail, along with our modifications for pairwise sampling.

3.3.1 Pairwise Optimal Computing Budget Allocation (POCBAm)

Optimal Computing Budget Allocation (OCBA) refers to a class of sampling allocation policies based on a Bayesian framework. Since it was first proposed in [24], several different variants of OCBA have been developed [27, 28, 70]. The procedure was adapted in [25] for optimal subset selection, with the name OCBA_m to refer to the selection of multiple elements.

Here, we implement the variant first defined by [29], and later also evaluated by [14], which we adapt both for selecting a subset rather than a single alternative and to use pairwise comparisons, and thus refer to as POCBAm. At each stage of the sampling process POCBAm aims to maximize the estimated increase in the probability of correct selection (PCS) gained from the sample. To estimate PCS, we consider the information we have gained from our sampling process to far; we have an estimate $\tilde{\mu}_{i,j}$ for the mean of each sample outcome $\mu_{i,j}$, and the standard deviation $\tilde{\sigma}_{i,j}$ of the sampling results obtained so far. Using a Gaussian approximation, we model the uncertainty of our estimate $\tilde{\mu}_{i,j}$ using the standard error $\frac{\tilde{\sigma}_{i,j}}{\sqrt{n_{i,j}}}$, where $n_{i,j}$ denotes the number of samples performed of the pair (a_i, a_j) . We use these pairwise estimates to construct distributions $\tilde{\mathcal{S}}_p$ for the overall Borda scores of each alternative a_p :

$$\tilde{\mathcal{S}}_p \sim \mathcal{N} \left(\hat{\boldsymbol{\mu}}_p = \sum_{q,q \neq p} \tilde{\mu}_{p,q}, \hat{\boldsymbol{\sigma}}_p^2 = \sum_{q,q \neq p} \frac{\tilde{\sigma}_{p,q}^2}{n_{p,q}} \right) \quad (3.1)$$

With these Borda score distributions, our *expected PCS* (*EPCS*) would sim-

ply be the probability that each of the alternative scores does indeed fall in the correct set, i.e:

$$EPCS = \mathbb{P}\{\tilde{\mathcal{S}}_p > \tilde{\mathcal{S}}_q, \text{ for all } p \in \mathcal{I}, q \notin \mathcal{I}\}$$

As we only need the relative values of the *EPCS* for each pair, we use the lower bound *approximate expected probability of correct selection (AEPCS)* as described in [25] to simplify the calculation:

For a constant c :

$$EPCS \geq \mathbb{P} \left[\left(\bigcap_{p \in \mathcal{I}} \{\tilde{\mathcal{S}}_p > c\} \right) \cap \left(\bigcap_{q \notin \mathcal{I}} \{\tilde{\mathcal{S}}_q < c\} \right) \right] \equiv AEPCS \quad (3.2)$$

To obtain the best approximation of EPCS, we want to choose c in order to maximize AEPCS and make our lower bound as tight as possible. As suggested in [25], we use:

$$c = \frac{\hat{\sigma}_{k+1}\hat{\mu}_k + \hat{\sigma}_k\hat{\mu}_{k+1}}{\hat{\sigma}_k + \hat{\sigma}_{k+1}}$$

where $\hat{\mu}_k$, $\hat{\sigma}_k$ and $\hat{\mu}_{k+1}$, $\hat{\sigma}_{k+1}$ are the score means and standard errors of the alternatives currently ranked k^{th} and $(k+1)^{\text{th}}$ respectively.

However, calculating this probability directly is not straightforward. Unlike in [25], alternative scores are not independent as the sums used to calculate $\tilde{\mathcal{S}}_p$ and $\tilde{\mathcal{S}}_q$ include the mean estimates $\mu_{p,q}$ and $\mu_{q,p}$ of the paired random variables $X_{p,q}$ and $X_{q,p}$ respectively, as described above. Instead, this pairing ensures that the correlation $\rho_{p,q}$ is negative between any pair of alternatives. Thus using Slepian's Theorem, as described in [94] (Theorem 2.1.1 and Corollary 1), and assuming joint normality between alternatives, we can produce an upper bound for the parts of

$AEPCS$ from the sets of alternatives on each side of the threshold:

$$\begin{aligned} \mathbb{P}\left(\bigcap_{p \in \mathcal{I}} \{\tilde{\mathcal{S}}_p > c\}\right) &\leq \prod_{p \in \mathcal{I}} \mathbb{P}\{\tilde{\mathcal{S}}_p > c\} \\ \mathbb{P}\left(\bigcap_{q \notin \mathcal{I}} \{\tilde{\mathcal{S}}_q < c\}\right) &\leq \prod_{q \notin \mathcal{I}} \mathbb{P}\{\tilde{\mathcal{S}}_q < c\} \end{aligned} \quad (3.3)$$

Sketch proofs of inequalities 3.3 are given in Appendix A. Similarly, the negative correlations between alternative scores gives us an obvious lower bound for pairs of alternatives from either side of the threshold. For $p \in \mathcal{I}, q \notin \mathcal{I}$:

$$\mathbb{P}(\tilde{\mathcal{S}}_p > c) \mathbb{P}(\tilde{\mathcal{S}}_q < c) \leq \mathbb{P}(\{\tilde{\mathcal{S}}_p > c\} \cap \{\tilde{\mathcal{S}}_q < c\}) \quad (3.4)$$

Given these bounds, it seems reasonable to expect that the product over elements of Equation 3.2 will provide an acceptable and easy to calculate approximation of $AEPCS$, given again that only relative values are needed. Thus, in practice we approximate $AEPCS$ using:

$$AEPCS \approx \prod_{p \in \mathcal{I}} \mathbb{P}\{\tilde{\mathcal{S}}_p > c\} \prod_{q \notin \mathcal{I}} \mathbb{P}\{\tilde{\mathcal{S}}_q < c\} \quad (3.5)$$

To estimate the expected increase in $AEPCS$ due to allocating an additional sample, the POCBAM procedure considers the effect of allocating a single additional sample to a particular pairwise comparison and none to the others. The expectation is that, by collecting an additional sample from the random variable corresponding to that pair, the estimate of the sample mean and standard deviation will not change (as they are calculated using unbiased estimates), but the standard error of our estimate of the mean of the outcome from that pairwise comparison will decrease. We model this effect, for a sample allocated to the pair (a_i, a_j) , by scaling the distribution of the scores $\tilde{\mathcal{S}}_p$ of the alternatives a_p to our expected post-sample distributions $\tilde{\mathcal{S}}_p^{i,j}$:

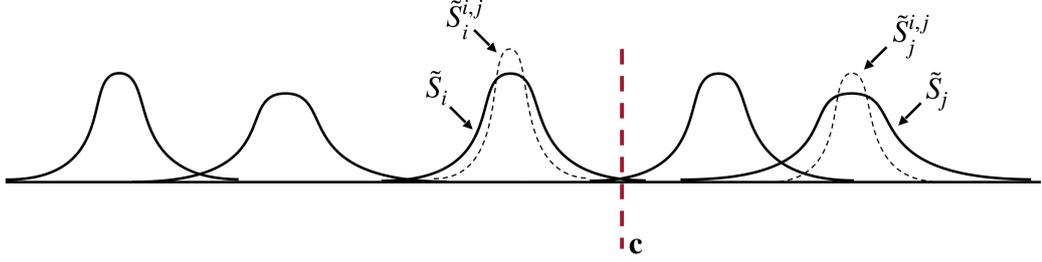


Figure 3.1: Illustration of the expected effect on the approximated posterior score distributions due to allocating a sample to the pair (a_i, a_j) . The expected post-sample distributions $\tilde{S}_i^{i,j}$ and $\tilde{S}_j^{i,j}$ are narrower, increasing the probability mass lying on the “correct” side of the threshold c and thereby increasing AEPCS.

$$\tilde{S}_p^{i,j} \sim \mathcal{N} \left(\hat{\mu}_p^{i,j} = \sum_{q,q \neq p} \tilde{\mu}_{p,q}, (\hat{\sigma}_p^{i,j})^2 = \sum_{q,q \neq p} \frac{\tilde{\sigma}_{p,q}^2}{n_{p,q} + \mathbb{1}\{p, q = i, j\}} \right) \quad (3.6)$$

where $\mathbb{1}\{p, q = i, j\}$ is the indicator function that returns 1 if either $p = i$ and $q = j$, or $q = i$ and $p = j$. Figure 3.1 gives an illustration of the expected effect of a sample on the score distributions relative to the threshold value c .

Calculating $\tilde{S}_p^{i,j}$ for all alternatives allows us to obtain a prediction for AEPCS after having performed the additional comparison of (a_i, a_j) :

$$\text{For } c = \frac{\hat{\sigma}_{k+1} \hat{\mu}_k + \hat{\sigma}_k \hat{\mu}_{k+1}}{\hat{\sigma}_k + \hat{\sigma}_{k+1}},$$

$$\begin{aligned} AEPCS^{i,j} &= \prod_{p \in \mathcal{I}} \mathbb{P}\{\tilde{S}_p^{i,j} > c\} \prod_{q \notin \mathcal{I}} \mathbb{P}\{\tilde{S}_q^{i,j} < c\} \\ &= \prod_{p \in \mathcal{I}} \left(1 - \Phi \left(\frac{c - \hat{\mu}_p^{i,j}}{\hat{\sigma}_p^{i,j}} \right) \right) \prod_{q \notin \mathcal{I}} \left(\Phi \left(\frac{c - \hat{\mu}_q^{i,j}}{\hat{\sigma}_q^{i,j}} \right) \right) \end{aligned} \quad (3.7)$$

where Φ is the cumulative distribution function of the standard normal distribution.

At each step, POCBam selects and performs a single sample of the pair that maximizes $AEPCS^{i,j}$, before recalculating, repeating until the pre-sample

$AEPCS > (1 - \alpha)$ for a pre-specified accuracy parameter $\alpha \in [0, 1]$. Initial values for pairwise sample mean and variance estimates $\tilde{\mu}_{i,j}$ and $\tilde{\sigma}_{i,j}^2$ are obtained by performing an initial warm-up phase where each alternative pair is sampled n_0 times. Despite the cost of this initial sampling of all $\frac{1}{2}(K^2 - K)$ alternative pairs, obtaining reasonable starting estimates for pairwise sample means and variances is important to ensure that the estimated alternative Borda Score distributions are accurate enough to usefully inform sample selection. With discrete sample values, we apply add-one Laplace smoothing [65] to our initial estimates to ensure $\tilde{\sigma}_{i,j}^2 > 0$. This is necessary to guarantee asymptotic correctness, discussed further in Section 3.3.3 below.

3.3.2 Pairwise Knowledge Gradient (PKG)

Pairwise Knowledge Gradient (PKG) is a one-step Bayesian look-ahead policy that aims to maximize the expected value gained by collecting one additional sample under the assumption that the sampling process will terminate immediately afterwards. In the context of optimal subset selection, given an index set \mathcal{I} , its value is typically determined by the zero-one loss function:

$$U(\mathcal{I}) = \begin{cases} 1, & \text{if } \mathcal{I} \text{ is correct} \\ 0, & \text{otherwise} \end{cases}$$

For applications where the requirement to return *exactly* the correct top-k alternatives is less strict, the opportunity cost loss function (difference between the cumulative fitness of the true best and the selected alternatives) is also commonly used. Suppose during the sampling process, we currently consider the index set \mathcal{I} to contain the k best alternatives and denote the (as yet unknown) best index set we would obtain after sampling a pair (a_i, a_j) by $\mathcal{I}^{i,j}$. The expected value gain (for

Table 3.1: The Pairwise Optimal Computing Budget Allocation for Subset Selection Procedure

INPUT:	Set of K alternatives $\{a_1, \dots, a_K\}$, Required selection size k , Accuracy parameter α .
INITIALIZE:	Perform n_0 samples of each pair of alternatives; $n_{p,q} = n_0$ for all p, q , Sample means $\tilde{\mu}_{p,q} = \frac{1}{n_{p,q}} \sum X_{p,q}$, and: Standard dev. $\tilde{\sigma}_{p,q} = \sqrt{\frac{1}{n_{p,q}-1} \sum (X_{p,q} - \tilde{\mu}_{p,q})^2}$, For all $p = 1, \dots, K$: alternative scores $S_p = \sum_{q, q \neq p} \tilde{\mu}_{p,q}$, Index set \mathcal{I} of best k alternatives.
WHILE $AEPCS < (1 - \alpha)$ DO:	
FOR ALL PAIRS (a_i, a_j) :	
UPDATE:	For all $p = 1, \dots, K$: Alternative score means $\hat{\mu}_p^{i,j} := S_p$, Alternative score std. devs. $\hat{\sigma}_p^{i,j} := \sum_{q, q \neq p} \frac{\tilde{\sigma}_{p,q}^2}{n_{p,q} + \mathbb{1}_{\{p,q=i,j\}}}$, Boundary value $c = \frac{\hat{\sigma}_{k+1} \hat{\mu}_k + \hat{\sigma}_k \hat{\mu}_{k+1}}{\hat{\sigma}_k + \hat{\sigma}_{k+1}}$, $AEPCS^{i,j} = \prod_{p \in \mathcal{I}} \left(1 - \Phi \left(\frac{c - \hat{\mu}_p^{i,j}}{\hat{\sigma}_p^{i,j}} \right) \right) \prod_{q \notin \mathcal{I}} \left(\Phi \left(\frac{c - \hat{\mu}_q^{i,j}}{\hat{\sigma}_q^{i,j}} \right) \right)$.
END FOR	
SAMPLE:	Select pair (a_i, a_j) that maximizes $AEPCS^{i,j}$, Perform sample of (a_i, a_j) , $n_{i,j} \leftarrow n_{i,j} + 1$, UPDATE: $\tilde{\mu}_{i,j}, \tilde{\sigma}_{i,j}, S_i, S_j$,
UPDATE:	\mathcal{I} .
END WHILE	
RETURN	\mathcal{I}

the zero-one loss function) of such a sample is simply:

$$V^{i,j} = \mathbb{P}\{U(\mathcal{I}^{i,j}) = 1|U(\mathcal{I}) = 0\} - \mathbb{P}\{U(\mathcal{I}^{i,j}) = 0|U(\mathcal{I}) = 1\}$$

However, as we do not know the value of $U(\mathcal{I})$ during our sampling process, $V^{i,j}$ cannot be computed. To allow us to approximate it, we make the assumption that the information gained by further sampling should improve our ability to identify the correct index set and thus will not cause us to discard a correct index set \mathcal{I} , i.e that $\mathbb{P}\{U(\mathcal{I}^{i,j}) = 0|U(\mathcal{I}) = 1\} = 0$. Under this assumption, and the assumption that the next sample will be the last, the expected value of information gained from performing a sample is simply the probability that the sample will change our estimate of the index set. Thus, we define the approximate value gain $AV^{i,j}$ of sampling the pair (a_i, a_j) :

$$AV^{i,j} := \mathbb{P}\{\mathcal{I}^{i,j} \neq \mathcal{I}\} \tag{3.8}$$

For the sample to change our current index set \mathcal{I} , these score changes must be sufficiently large to move one of S_i, S_j either into, or out of the current k best score estimates. The sample may either (i) increase the score estimate S_i of a_i , and thus decrease S_j by an equal amount, or (ii) decrease S_i and so increase S_j by a corresponding amount. For case (i), we denote the required increase in S_i to change \mathcal{I} by $\delta_i^{i,j}$. Similarly, for case (ii), we denote the required increase in S_j to change \mathcal{I} by $\delta_j^{i,j}$. There are several cases for calculating $\delta_i^{i,j}$ and $\delta_j^{i,j}$, dependent on whether

a_i and a_j are present in the current estimated index set \mathcal{I} :

$$\delta_i^{i,j} = \begin{cases} S_k - S_i & \text{if } a_i, a_j \notin \mathcal{I} \\ S_j - S_{k+1} & \text{if } a_i, a_j \in \mathcal{I} \\ \min\{\frac{S_j - S_i}{2}, S_k - S_i, S_j - S_{k+1}\} & \text{if } a_j \in \mathcal{I}, a_i \notin \mathcal{I} \\ \infty & \text{if } a_i \in \mathcal{I}, a_j \notin \mathcal{I} \end{cases} \quad (3.9)$$

and vice versa for $\delta_j^{i,j}$. If both a_i and a_j are outside the current index set, the increase in S_i must be sufficiently large to make S_i exceed S_k , the lowest score for alternatives currently in \mathcal{I} . Similarly, if a_i and a_j are both currently in \mathcal{I} , the increase in S_i must be large enough that the corresponding decrease in S_j is enough to reduce S_j to below S_{k+1} , the highest score for alternatives not in \mathcal{I} . Alternatively, if $a_i \notin \mathcal{I}$, $a_j \in \mathcal{I}$, S_i must either increase enough to exceed S_k , or cause a decrease in S_j sufficiently large to reduce it below S_{k+1} , or both increase S_i and decrease S_j enough to make $S_i > S_j$. Finally, if $a_i \in \mathcal{I}$, $a_j \notin \mathcal{I}$, no increase in S_i can change \mathcal{I} so $\delta_i^{i,j}$ is infinite. However, in this case, $\delta_j^{i,j} = \min\{\frac{S_i - S_j}{2}, S_k - S_j, S_i - S_{k+1}\}$, so at least one of $\delta_i^{i,j}$ and $\delta_j^{i,j}$ will always be finite. The sampling outcome required to change our estimate of $\mu_{i,j}$ from $\tilde{\mu}_{i,j}$ after $n_{i,j}$ samples, to $\tilde{\mu}_{i,j} + \delta_i^{i,j}$ after $n_{i,j} + 1$ samples, and thereby increase S_i by $\delta_i^{i,j}$, is then simply:

$$\Delta_i^{i,j} = \delta_i^{i,j} (n_{i,j} + 1) + \tilde{\mu}_{i,j} \quad (3.10)$$

So a sample result from the pair (a_i, a_j) of at least $\Delta_i^{i,j}$, or at least $\Delta_j^{i,j}$ will cause our index set \mathcal{I} to change. Thus, the expected value of information $AV^{i,j}$ from this sample under our knowledge gradient assumptions is just the probability of either of the required sampling outcomes. With Gaussian sampling noise, this is given by:

$$AV^{i,j} = 2 - \left[\Phi \left(\frac{|\Delta_i^{i,j} - \tilde{\mu}_{i,j}|}{\tilde{\sigma}_{i,j}} \right) + \Phi \left(\frac{|\Delta_j^{i,j} - \tilde{\mu}_{i,j}|}{\tilde{\sigma}_{i,j}} \right) \right] \quad (3.11)$$

where Φ is the cumulative distribution function of the standard normal distribution. At each step after our initial warm-up phase, we choose to perform the sample that maximizes AV .

Throughout our sampling process, we maintain estimates of sampling mean $\tilde{\mu}_{i,j}$ and variance $\tilde{\sigma}_{i,j}^2$ for each pair, which we can use to construct an estimate for the probability of correct selection given our sampling results so far. As with the POCBAm method, we use the *AEPCS* approximation given in Section 4.1 to simplify the calculation, differing only in that we do not scale the variance of our alternative score distributions \tilde{S}_p to predict future sample effects as we do with POCBAm, instead using only the actual sampling results obtained so far. We use *AEPCS* as a stopping criterion, halting our sampling process when $AEPCS > (1 - \alpha)$.

3.3.3 Asymptotic Correctness

A desirable property for sampling methods is that of *Asymptotic Correctness*; the guarantee of convergence to the best possible solution given an infinite sampling budget. For instance, the simple policy of uniform sample allocation is asymptotically correct. Under this policy, with an infinite sampling budget, infinitely many samples will be allocated to each possible pair and so each pairwise mean estimate and therefore all Borda Score estimates for alternatives will converge to the true value. This idea is important when discussing asymptotic correctness, as so long as we can guarantee our sampling method will eventually allocate infinitely many samples to each pair, the Central Limit Theorem (CLT) justifies that eventually our Borda Score estimates for alternatives will be sufficiently accurate to guarantee we select the correct subset of high scoring alternatives.

Table 3.2: The Pairwise Knowledge Gradient Procedure

INPUT:	Set of K alternatives $\{a_1, \dots, a_K\}$, Required selection size k , Accuracy parameter α .
INITIALIZE:	Perform n_0 samples of each pair of alternatives; $n_{p,q} = n_0$ for all p, q , and: Sample means $\tilde{\mu}_{p,q} = \frac{1}{n_{p,q}} \sum X_{p,q}$, Standard dev. $\tilde{\sigma}_{p,q} = \sqrt{\frac{1}{n_{p,q}-1} \sum (X_{p,q} - \tilde{\mu}_{p,q})^2}$, For all $p = 1, \dots, K$: alternative scores $S_p = \sum_{q,q \neq p} \tilde{\mu}_{p,q}$, Index set \mathcal{I} of best k alternatives.
WHILE $AEPCS < (1 - \alpha)$ DO:	
FOR ALL PAIRS (a_i, a_j) :	
UPDATE:	Required score changes $\delta_i^{i,j}$ and $\delta_j^{i,j}$, $\Delta_i^{i,j} = \delta_i^{i,j} (n_{i,j} + 1) + \tilde{\mu}_{i,j}$, $\Delta_j^{i,j} = \delta_j^{i,j} (n_{j,i} + 1) + \tilde{\mu}_{j,i}$, $AV^{i,j} = 2 - \left[\Phi \left(\frac{ \Delta_i^{i,j} - \tilde{\mu}_{i,j} }{\tilde{\sigma}_{i,j}} \right) + \Phi \left(\frac{ \Delta_j^{i,j} - \tilde{\mu}_{j,i} }{\tilde{\sigma}_{j,i}} \right) \right]$.
END FOR	
SAMPLE:	Select pair (a_i, a_j) that maximizes $AV^{i,j}$, If $\max_{i,j} \{AV^{i,j}\} = 0$, select sample uniformly at random, Perform sample of (a_i, a_j) , $n_{i,j} \leftarrow n_{i,j} + 1$,
EST. PCS:	UPDATE: $\tilde{\mu}_{i,j}, \tilde{\sigma}_{i,j}, S_i, S_j$,
For all $p = 1, \dots, K$:	
Alternative score means $\hat{\mu}_p := S_p$,	
Alternative score std. devs. $\hat{\sigma}_p := \sum_{q,q \neq p} \tilde{\sigma}_{p,q}^2$	
Boundary value $c = \frac{\tilde{\sigma}_{k+1} \hat{\mu}_k + \tilde{\sigma}_k \hat{\mu}_{k+1}}{\hat{\sigma}_k + \hat{\sigma}_{k+1}}$,	
$AEPCS = \prod_{p \in \mathcal{I}} \left(1 - \Phi \left(\frac{\hat{\mu}_p - c}{\hat{\sigma}_p} \right) \right) \prod_{q \notin \mathcal{I}} \left(\Phi \left(\frac{c - \hat{\mu}_q}{\hat{\sigma}_q} \right) \right)$.	
UPDATE: \mathcal{I} .	
END WHILE	
RETURN \mathcal{I}	

The asymptotic correctness of OCBA methods on standard (i.e non-pairwise) ranking and selection problems is well established, with proofs of the property given for different formulations of the method in [38] and [26].

Theorem 3.3.1. *Pairwise OCBA for top-k selection (POCBAm) is asymptotically correct.*

Proof. After our n_0 warm-up samples of each pair we have $\tilde{\sigma}_{i,j} > 0$ for all pairs a_i, a_j and thus $AEPCS^{i,j} > 0$. Now, when we sample a pair, we only affect the pairwise mean and score estimates of the two alternatives directly involved in the comparison, leaving most of the terms in $AEPCS$ unchanged. Thus, we can write the expected increase in $AEPCS$ due to sampling (a_i, a_j) (which we denote $\Delta AEPCS^{i,j}$) by:

$$\Delta AEPCS^{i,j} = C_{i,j} \left[\Phi \left(\frac{(|c - \tilde{\mu}_{i,j}|)(n_{i,j} + 1)}{\tilde{\sigma}_{i,j}} \right) - \Phi \left(\frac{(|c - \tilde{\mu}_{i,j}|)n_{i,j}}{\tilde{\sigma}_{i,j}} \right) \right]$$

Where Φ is the cumulative distribution function of the standard normal distribution. With $0 < C_{i,j} < 1$ being the product of all terms in $AEPCS^{i,j}$ that are unaffected by sampling (a_i, a_j) . Now:

$$\lim_{n_{i,j} \rightarrow \infty} \Delta AEPCS^{i,j} = C_{i,j}(\Phi(+\infty) - \Phi(+\infty)) = C_{i,j}(1 - 1) = 0$$

Therefore, as our total number of samples allocated $N = \sum_{i,j} n_{i,j} \rightarrow \infty$, for at least some pairs (a_i, a_j) , we must have $n_{i,j} \rightarrow \infty$ and therefore $\Delta AEPCS^{i,j} \rightarrow 0$. If this is the case for all pairs, then we are done. Let F denote the set of pairs for which $n_{i,j}$ remains finite, then eventually we must reach a state where we allocate no further samples to F . But if $n_{i,j} \rightarrow \infty$ for all pairs not in F , then for any $\epsilon > 0$ there exists some number of samples N' such that, once at least N' samples have been taken we have $\max_{(a_i, a_j) \notin F} \Delta AEPCS^{i,j} < \epsilon$. If we choose $\epsilon < \min_{(a_i, a_j) \in F} \Delta AEPCS^{i,j}$ then for some N^* we have $\max_{(a_i, a_j) \notin F} \Delta AEPCS^{i,j} < \min_{(a_i, a_j) \in F} \Delta AEPCS^{i,j}$ after N^* samples, and so POCBAm will allocate our next sample to F . Thus, by

contradiction, F is empty. \square

The question of asymptotic correctness for pairwise knowledge gradient is less straightforward. With unbounded sample responses of finite variance, KG is asymptotically correct [38] and we show in Theorem 3.3.2 that this remains true for PKG. However, with bounded sample outcomes, this does not hold, as we discuss below. In Theorem 3.3.3, we give a specific example of how the asymptotic correctness of PKG may break with binary sampling.

Theorem 3.3.2. *Pairwise Knowledge Gradient (PKG) is asymptotically correct for sampling models with unbounded sample results of finite variance.*

Proof. For every pair i, j at least one of $\delta_i^{i,j}$ and $\delta_j^{i,j}$ must be finite. Thus, at least one of $\Delta_i^{i,j}$ and $\Delta_j^{i,j}$ will be finite and, as sample results are unbounded, either $\mathbb{P}[X_{i,j} > \Delta_i^{i,j}] > 0$ or $\mathbb{P}[X_{j,i} > \Delta_j^{i,j}] > 0$. Hence,

$$AV^{i,j} > 0$$

for every pair. Now, as $\lim_{n_{i,j} \rightarrow \infty} \Delta_i^{i,j} = \infty$,

$$\lim_{n_{i,j} \rightarrow \infty} AV^{i,j} = 2 - \left[\Phi\left(\frac{|\infty - \mu_{i,j}|}{\sigma_{i,j}}\right) + \Phi\left(\frac{|\infty - \mu_{i,j}|}{\sigma_{i,j}}\right) \right] = 2 - (1 + 1) = 0$$

Thus, suppose that as $N = \sum_{\forall i,j} n_{i,j} \rightarrow \infty$ there are some pairs sampled only finitely many times, and denote these by F , but again, if $n_{i,j} \rightarrow \infty$ for all pairs not in F , then for any $\epsilon > 0$ there is a number of samples N' after which $\max_{(a_i, a_j) \notin F} AV^{i,j} < \epsilon$. If we choose $\epsilon < \min_{(a_i, a_j) \in F} AV^{i,j}$ then after some N^* samples, we have $\max_{(a_i, a_j) \notin F} AV^{i,j} < \min_{(a_i, a_j) \in F} AV^{i,j}$ and thus PKG will allocate our next sample to F . \square

However, when sample outcomes are bounded, even the standard formulation of KG can encounter states where it is unable to allocate a sample. At each

step of this algorithm and for each pair (a_i, a_j) , we calculate an estimate $AV^{i,j}$ of the probability that collecting a single additional sample will change the scores of alternatives a_i and a_j sufficiently to alter our top-rated subset. However, restricting the range from which sample results are drawn limits the change that sampling can make to the alternatives' scores. Specifically, using the example of binary sample outcomes; suppose we are in some knowledge state θ , with an estimate of $\tilde{\mu}_{i,j}^\theta$ for $X_{i,j}$ (and thus $\tilde{\mu}_{j,i}^\theta = 1 - \tilde{\mu}_{i,j}^\theta$ for $X_{j,i}$) and we perform a single additional sample of $X_{i,j}$. Then we will have:

$$\tilde{\mu}_{i,j}^{\theta+1} = \begin{cases} \tilde{\mu}_{i,j}^\theta + \frac{1}{n_{i,j}+1}(1 - \tilde{\mu}_{i,j}^\theta), & \text{if } X_{i,j}^{\theta+1} = 1 \\ \tilde{\mu}_{i,j}^\theta - \frac{\tilde{\mu}_{i,j}^\theta}{n_{i,j}+1}, & \text{if } X_{i,j}^{\theta+1} = 0 \end{cases}$$

and vice versa for $\tilde{\mu}_{j,i}^\theta$. If the required difference in estimated score exceeds all these possible change amounts, then no single pairwise sample will be able to alter the current top set. This means that our knowledge gradient values will be:

$$AV^{i,j} = 0, \forall(a_i, a_j)$$

and our Knowledge Gradient policy will be unable to select a sample. This potential problem with the KG policy was hinted at in [75], and discussed in detail in [59].

It becomes necessary to consider multiple samples in order to find sampling sequences with non-zero change probabilities. [39] proposes the adapted KG(*) policy, which considers sequences of samples, selecting to perform the sample at the start of the shortest sequence required to change the ranking. They show that this policy performs well, but can be computationally very intensive, as the state space of sampling sequences grows rapidly as the sequences lengthen. In the case of pairwise sampling, with $\frac{1}{2}(K^2 - K)$ possible sample pairs at each stage, this method rapidly becomes computationally intractable for even modest values of K .

To solve this, [59] suggests an alternative method for formulating $AV^{i,j}$, lead-

ing to an adapted policy KG(min), that allocates based on minimizing the number of consecutive repeated samples of a single alternative needed to change the selection. For the standard subset selection problem, where simulation directly estimates the score of alternatives, this is sufficient to prevent the policy from failing and restore asymptotic correctness, as for any possible alternative score value S and accuracy ϵ , there is a finite string of sampling outcomes that can move the score estimate of an alternative to within ϵ of S , with non-zero probability. However, this is not necessarily true in the pairwise problem, which we show here for binary sample outcomes. In this example, pairwise outcomes are modelled with Bernoulli random variables $X_{i,j}$, paired such that $X_{i,j} = 1 - X_{j,i}$. Let $r_{i,j}$ denote the minimum number of consecutive samples of the pair (a_i, a_j) required to change the selected subset.

Theorem 3.3.3. *For any $K > 4$ and with binary pairwise sampling, it is possible that $r_{i,j} = \infty$ for all (a_i, a_j) regardless of the selected subset size k*

Proof. To show this, we aim to construct an example sampling situation whereby $r_{i,j} < \infty \implies K \leq 4$. Suppose at some point in our sampling process we have:

$$\tilde{\mu}_{i,j} = \begin{cases} 0.5 & \text{if } i, j \in \mathcal{I} \\ 0.5 & \text{if } i, j \notin \mathcal{I} \\ 1 & \text{if } i \in \mathcal{I}, j \notin \mathcal{I} \\ 0 & \text{if } i \notin \mathcal{I}, j \in \mathcal{I} \end{cases}$$

then the estimated difference in score of the k^{th} and $(k+1)^{\text{th}}$ best alternatives will be:

$$\begin{aligned} S_k - S_{k+1} &= \sum_{j,j \neq k} \tilde{\mu}_{k,j} - \sum_{j,j \neq k+1} \tilde{\mu}_{k+1,j} \\ &= (0.5(k-1) + K - k) - 0.5(K - k - 1) \\ &= 0.5K \end{aligned}$$

Now, $\min_{i,j} \{\delta_i^{i,j}\} = \delta_{k+1}^{k+1,k} = \frac{1}{2}(S_k - S_{k+1}) = \frac{K}{4}$, so finitely many samples must be able to alter $\tilde{\mu}_{k+1,k}$ by at least $\frac{K}{4}$ for $r_{k,k+1}$ to be finite. As $\tilde{\mu}_{i,j} \in [0, 1]$ for all a_i, a_j , the maximum change in $\tilde{\mu}_{k+1,k}$ we can obtain is 1. Hence, we require:

$$1 \geq \frac{K}{4}$$

□

Theorem 3.3.3 means that, when our sampling process returns binary preference feedback, and we have more than 4 alternatives to choose from, it can be the case that infinitely many samples of a particular pair cannot change the ranking. This means that a pairwise adaptation of KG(min) procedure proposed in [59] can fail, even in the asymptotic limit, which explains the observations reported in [43] and [76]. To prevent this asymptotic failure for our PKG policy, we adapt PKG to allow random sample selection in the case that $AV^{i,j} = 0$ for all pairs (a_i, a_j) .

3.4 Empirical Testing

In this section, we empirically evaluate the performance of our algorithms against other sampling methods. For comparison, we choose the *Active Ranking (AR)* method from [52] and the *Hoeffding Racing (H-Race)* method from [18], using the PBR objective. To the best of our knowledge, these methods represent the current state of the art for top- k selection for models without systematic regularity assumptions such as SST. We also include the performance of uniform sample allocation as an additional benchmark. We test their performance on a range of standard scoring models (BTL, SST, Unstructured), with both binary preference and value-based sample results.

3.4.1 Top- k Selection (2 from 5)

Here we simulate the problem of selecting the top 2 alternatives from a set of 5. Pairwise outcomes are binary, i.e for alternatives (a_i, a_j) , $X_{i,j}$ is Bernoulli distributed. We consider three different scoring models:

- BTL model: Here the underlying “true” quality of our alternatives is parametrized by a score vector $T = (t_{a_1}, \dots, t_{a_5})$. T fully determines the matrix of pairwise comparison outcome probabilities with $\mu_{i,j} = \frac{t_i}{t_i+t_j}$. We set $T = (0.9, 0.7, 0.5, 0.3, 0.1)$.
- SST model: We generate the pairwise comparison probability matrix according to the “Independent Bands” SST model from [86]. As they describe, the class of SST Bernoulli scoring models is characterized up to permutation of elements by the set of matrices whose upper-triangular entries lie in $[0.5, 1.0]$, increase along rows and decrease down columns. Thus, we generate the matrix of true comparison means M first by selecting the entry $M_{0,1} = \mu_{0,1}$ uniformly at random from $[\frac{1}{2}, 1]$, before populating the remainder of the upper triangle of the matrix row-wise, at each stage selecting values uniformly from the allowable interval, i.e bounded above either by 1 or the entry above, and bounded below either by $\frac{1}{2}$ or the entry to the left.
- Unstructured model: In this model pairwise comparison means are uncorrelated. Each entry in the upper triangle of the comparison matrix M is sampled independently and uniformly at random from $[0, 1]$. The “true” ranking of the alternatives is then determined by their Borda score.

The POCBA_m, PKG and AR methods contain an accuracy parameter α related to stopping time, which we vary to obtain a range of values. For the H-Race method, there are two parameters that affect the width of the confidence interval used to eliminate alternatives from the race, and therefore stopping time: α and n_{max} , the maximum number of samples allowed of each particular pair. Specifically,

the interval is defined for each alternative pair as:

$$\tilde{\mu}_{i,j} \pm \sqrt{\frac{1}{2n_{i,j}} \log \left(\frac{2K^2 n_{max}}{\alpha} \right)}$$

Thus for the H-Race method we obtain a range of different stopping times by varying n_{max} for three different values of α : 1.0, 0.1 and 0.01, and display the best performance from the three. The parameter values used for each method are given in the figure caption for each scoring model. We also included a fixed maximum budget total constraint of 10,000 samples per run for each method to ensure timely completion. We use correct selection success rate as our performance metric for each task, defined as the proportion of correctly identified top- k subsets over a large number of independent replications of the experiment, with different alternative populations and pairwise variances:

$$SuccessRate = \frac{\{\mathcal{I} = \mathcal{I}^*\}_\mathbf{1}}{N_{replications}} \quad (3.12)$$

We use $N_{replications} = 10,000$ for all the experiments in this chapter. Figure 3.2a shows the performance of each method at selecting the top 2 of 5 alternatives for the BTL model scenario. POCBAm is the best performer, with both PKG and POCBAm outperforming the comparison methods, achieving the same success rate using fewer samples. Both racing methods, particularly the H-Race, struggled due to the loose width of the bounds used to construct their confidence intervals. Without being able to successfully eliminate alternatives from the race before reaching n_{max} samples of each pair, H-Race performs essentially the same samples as simple uniform allocation.

Figure 3.2b shows the results for the SST scoring model. Overall the SST scoring model produces easier top- k selection problems than the BTL model used in the first scenario. The method for generating the underlying comparison matrix for the SST scoring model will, on average, produce pairwise means further from

0.5 than for the BTL model. For example, the expectation of the mean $\mu_{2,3}$ of the alternatives on the boundary between the top subset and the discarded subset will be:

$$\mathbb{E}[\mu_{2,3}] = \frac{0.5 + \mathbb{E}[\mu_{0,2}]}{2} = \frac{0.5 + \frac{\mathbb{E}[\mu_{0,1}] + 1}{2}}{2} = \frac{0.5 + \frac{0.5 + 1}{2} + 1}{2} = 0.6875,$$

compared to $\frac{0.7}{0.7+0.5} = 0.58\dot{3}$ in the BTL model experiment. Consequently, we see higher success rates at each given budget when compared to Figure 3.2a, and a much clearer improvement over uniform sampling for the AR method. The larger the differences in Borda score between alternatives, the easier it is for methods to identify which sampling pairs are irrelevant and thus to sample more efficiently than uniformly. However, we see that the confidence bound used by the H-Race method is again too loose to reliably eliminate alternatives before reaching n_{max} samples of each pair, and thus does not make any improvement over uniform allocation.

In contrast to the SST scenario, the unstructured scoring model shown in Figure 3.2c is much harder, with very little average distance between alternative’s Borda scores. Both POCBAm and PKG perform well on this problem, with POCBAm achieving the highest success rate as sampling budget increases. To accurately estimate the alternative scores with unstructured preferences, we have to learn far more about the underlying matrix M . This is particularly difficult for the AR method as, although this method chooses one alternative for the sampling pair directly, the other is selected at random, meaning we would require far more samples to be performed before being sure we have learned about every entry in M . This is reflected in low initial performance of AR. The H-Race also struggled in this problem, due to the closeness of the alternative’s total scores and the loose confidence bound used, and again fails to improve over the uniform benchmark.

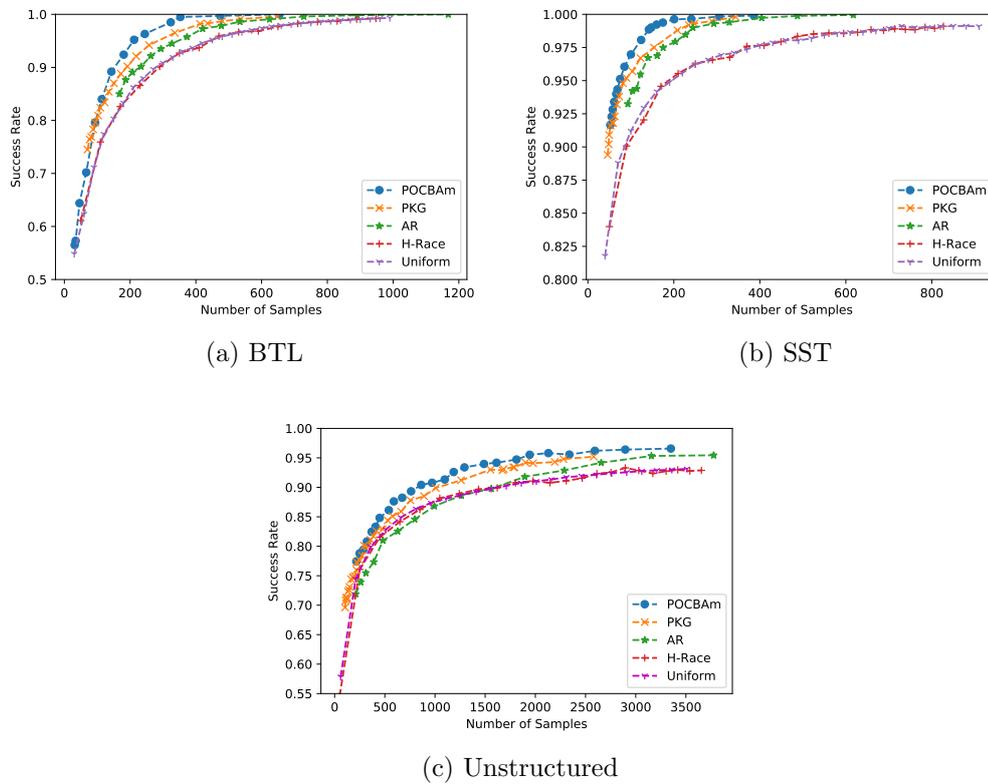


Figure 3.2: Performance of POCBAm, PKG, AR and H-Race algorithms against random allocation at best 2 of 5 selection for the BTL, SST and unstructured models. For POCBAm, we vary α between 0.5 and 0.01, for PKG between 0.3 and 0.001 and for AR between 0.15 and 0.01. For the H-Race, n_{max} ranges between 5 and 100 for the BTL and SST models, and between 5 and 450 for the Unstructured model, with $\alpha = 0.01$ for all three.

3.4.2 Other System Sizes

Here we examine effect of the number of alternatives and of the top subset size on the algorithm performance on the SST scoring model used in the previous section. Specifically, we test top 1 of 5 selection (finding the best single element) and top 4 of 10, with the results shown in Figures 3.3a and 3.3b respectively. POCBA is again the best performing method in both cases, reaching perfect accuracy on both problems with substantially fewer samples, particularly on the larger problem. When only needing to identify the single best alternative, both the AR and H-Race methods are able to exclude poorly performing alternatives much earlier as they need only be confident that they are beaten by a single competitor. As sampling budget increases, they are therefore able to allocate the last portion of their sampling budgets more effectively between fewer remaining pairs. This is reflected in their performance, with AR matching PKG and H-Race improving over uniform allocation. Figure 3.3c shows the performance on a much larger subset selection problem, choosing the top 40 of 100 alternatives. For this problem two other changes were made. Firstly the maximum budget constraint for the variable stopping time methods was increased to 200,000 samples to compensate for the increased number of alternative pairs. Secondly, the PKG method was adapted to use a fixed sampling budget, rather than a variable stopping point based on EPCS. As the number of alternatives increases, but the range of values for each pairwise comparison mean $\mu_{i,j}$ remains fixed and bounded, the relative effect that changing each sample mean may have on an alternative's score decreases. This makes it more likely that $AV^{i,j}$ will fall to zero for some or possibly all alternative pairs, as discussed in Section 4.3. If there are only a few pairs with non-zero $AV^{i,j}$ values, PKG will only select samples from amongst these, which can prevent EPCS from reaching $(1 - \alpha)$ even asymptotically. The fixed sampling budget allows PKG to terminate in these cases. If $AV^{i,j} = 0$ for all pairs, PKG has to resort to random sample allocation. Here

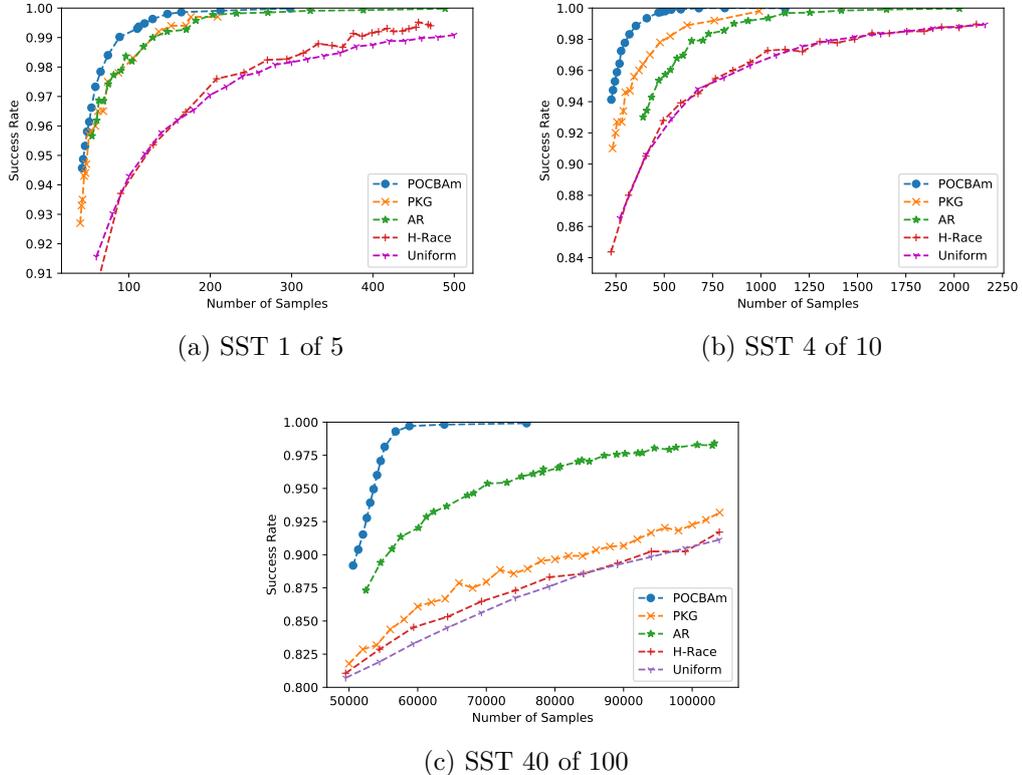


Figure 3.3: Performance of POCBAm, PKG, AR and H-Race algorithms against random allocation at best 1 of 5 selection (a), best 4 of 10 (b), and best 40 of 100 selection (c), for the SST scoring model. For POCBAm, we vary α between 0.5 and 0.01, for PKG between 0.3 and 0.001 and for AR between 0.2 and 0.01. For the H-Race, (a) uses $\alpha = 1.0$, n_{max} between 5 and 100, (b) uses $\alpha = 0.01$, n_{max} between 5 and 50, and (c) uses $\alpha = 0.01$, n_{max} between 10 and 25.

we see that these changes limit the effectiveness of PKG, reducing its improvement over uniform allocation compared to the smaller selection problems. Overall, we see that POCBAm is still the best performer. The AR method also performs well, substantially improving over uniform sampling.

3.4.3 Value-based Scoring Models

The final part of this section examines empirical performance of the sampling methods on top 2 of 5 selection on models where pairwise comparison feedback is con-

tinuous valued and unbounded. This value-based feedback increases the amount of information received from sampling; instead of simply receiving a 0 – 1 win/loss result as in our previous testing, we now gain a measure of the magnitude of an alternative’s win or loss.

We test performance on value-based SST and unstructured problem models, the BTL model used in the previous section being suitable only for binary preference-based sampling. For both models, we assume that sample outcomes are normally distributed, and choose the underlying variance for each pair uniformly at random from $[0, 1]$. The matrix of pairwise comparison means for each model is then generated as described below:

- Value-based SST model: As before, the upper triangle of M should increase along rows and decrease down columns. Thus we populate the upper triangle of M using the same procedure as for the binary SST model, except using $[0, 1]$ instead of $[0.5, 1]$ as the allowable interval. The lower triangle is then filled according to $\mu_{j,i} = -\mu_{i,j}$. Note that value-based SST comparison matrices are skew-symmetric.
- Value-based Unstructured model. Entries in the upper triangle are chosen independently and uniformly at random from $[0, 1]$, and the lower triangle filled according to $\mu_{j,i} = -\mu_{i,j}$.

Figure 3.4a shows the performance for the value-based SST model. With normally distributed sample results, PKG will be asymptotically correct (Theorem 3.3.2), so should no longer encounter states where $AV^{i,j} = 0$ and should no longer have to resort to random samples allocation. As such, performance of PKG and POCBA_m seems to be very similar and substantially better than the comparison methods.

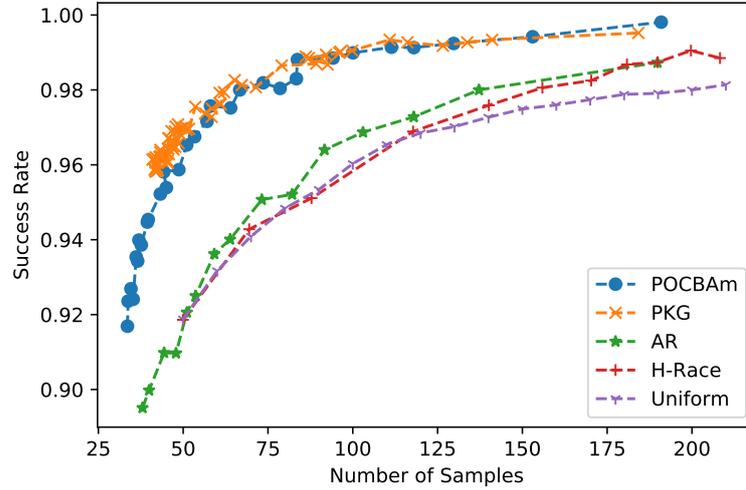
We again see that the unstructured model in Figure 3.4b is much more difficult, with all methods requiring far more samples to reach their stopping points. As

Table 3.3: Percentage reduction in sampling budget to match performance of uniform sample allocation for each sampling method. Best values shown in bold.

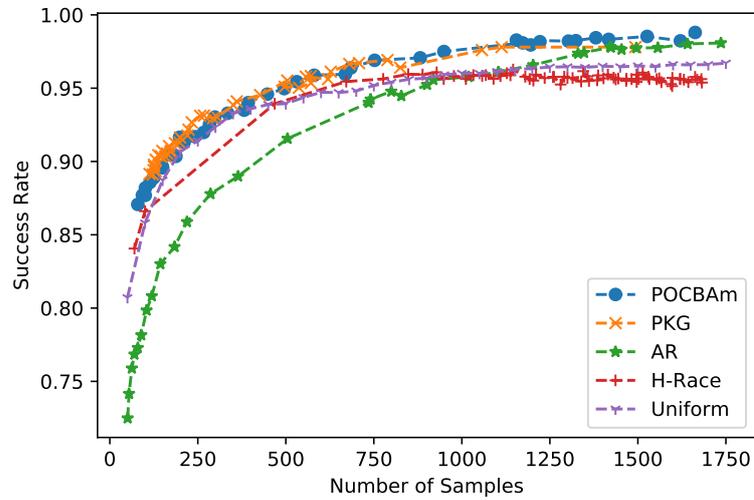
Scoring Model		Uniform		Sampling Budget Reduction			
		Samples	Succ. Rate	POCBAM	PKG	AR	H-Race
2 of 5	BTL	1000	0.993	64.7%	35.2%	27.1%	3.34%
	SST	800	0.991	78.0%	70.7%	63.4%	0.512%
	Unstr.	3500	0.931	63.0%	51.9%	24.1%	0.151%
	1 of 5 (SST)	500	0.991	81.7%	71.1%	70.6%	24.8%
	4 of 10 (SST)	2000	0.988	81.0%	66.6%	56.9%	-1.25%
	40 of 100 (SST)	100,000	0.906	49.4%	13.4%	42.5%	-1.48%
Value-Based	SST	200	0.980	63.2%	67.4%	31.4%	25.9%
	Unstr.	2000	0.970	62.4%	60.6%	32.7%	N\A

with the binary unstructured model from Figure 3.2c, we see that the AR method initially performs poorly, as its random sample selection is unable to ensure sufficient information collection from all pairs without taking a large number of samples. Interestingly, as the number of samples taken increases, the success rate of the H-Race method falls behind uniform sampling. The bounds for the alternative’s Borda score confidence intervals used by the H-Race are the arithmetic means of the bounds for the individual pairwise confidence bounds, calculated using only the alternatives still included in the race. When pairwise means are uncorrelated, as in the unstructured model, these become progressively poorer estimates of alternatives Borda scores whenever alternatives are removed, leading to incorrect classification of the remaining alternatives.

Table 3.3 provides a summary of our empirical results, showing the percentage reduction in samples required to achieve the same success rate as uniform sampling for each method on each scoring model.



(a) Value-based SST



(b) Value-based Unstructured

Figure 3.4: Performance of POCBAm, PKG, AR and H-Race algorithms against random allocation at best 2 of 5 selection with normally distributed sample results for SST (a) and unstructured (b) models. Sub-figure (a) uses α between 0.5 and 0.01 for POCBAm, between 0.05 and 10^{-5} for PKG and between 0.15 and 0.01 for AR with n_{max} between 5 and 80, and $\alpha = 1.0$. Sub-figure (b) uses α between 0.3 and 0.001 for POCBAm, between 0.03 and 10^{-5} for PKG, and between 0.15 and 0.001 for AR with n_{max} between 5 and 2500, and $\alpha = 0.01$.

3.5 Summary

In this chapter, we have presented two new pairwise subset selection methods, adapted from well known sampling algorithms from the Simulation Optimization community, as well as theoretical guarantees of asymptotically correct performance under certain conditions. Additionally, we identify an interesting idiosyncrasy of Knowledge Gradient policies with bounded pairwise sampling, where even n -step sampling methods can fail.

In our empirical testing, we see that both PKG and POCBAM offer improvements over current state-of-the-art top- k sampling procedures for scoring models without dependence on structural assumptions like the SST property. POCBAM in particular performed well across all the test scenarios, both with binary and unbounded value-based sample feedback, and with both structured and unstructured underlying models.

A possible future development would be to consider correlations between pairwise sample distributions, as they would occur if there was an underlying (unknown) quality of each solution that would influence the outcome. In such cases, it may be possible to further improve our sampling method by correctly learning and accounting for this dependence between alternative scores.

There are other forms of the standard OCBA method such as the one described in [25]. These are based on evaluating the asymptotically optimal proportional sample allocation based on current information and then recommending sampling proportionally. As POCBAM was generally the best performing method, it might be interesting to investigate pairwise adaptations of these other OCBA algorithms.

CHAPTER 4

Applications to Evolutionary Strategies

4.1 Introduction

In this chapter, we consider a natural application of our proposed POCBA_m sampling method – integrating efficient sampling into fitness evaluation for Evolutionary Algorithms (EAs).

EAs are a class of stochastic, derivative-free techniques for black-box function optimization. As the name suggests, they are loosely inspired by the biological process of evolution and the concept of “survival of the fittest”. As derivative-free methods, they are commonly used for problems where gradient information is unavailable, but have also successfully been applied to a wide range of non-linear and non-convex optimization problems. They generally make fewer assumptions about the shape of the optimization function and are more robust to rugged fitness landscapes and local optima than gradient-based approaches. A plethora of different EA methods exist, but methods generally follow the same iterative framework. Each generation of the EA takes a population of alternatives (candidate solutions), measures the fitness of these alternatives and creates a set of alternatives to form the next generation by applying a set of stochastic operations with some form of bias

towards high-fitness alternatives. A common method for doing this would be to select a top- k subset of individuals by fitness value, then use this top- k subset to generate the next generation through reproduction and/or random mutation. In this section, we focus on one particular class of EAs known as Evolutionary Strategies (ES). ES use real valued representations and apply mutation using normally distributed random modifications [4]. Candidate solutions are generally referred to as “individuals” and the set of all individuals as a population. To maintain consistency with the previous chapter, we continue referring to them as “alternatives”. The best known and current state-of-the art ES is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [49]. CMA-ES represents the evolutionary population as a multivariate Gaussian distribution in the domain of the function to be optimized. Each generation, the optimizer performs mutation and recombination by taking a set of samples from this distribution, evaluates the fitness of these alternatives, then uses the highest fitness alternatives to determine the parameters of the distribution of the next generation. In Section 4.3, we give more detail on the CMA-ES optimizer and how our proposed POCBA method can be integrated to improve fitness evaluation in pairwise cases.

4.2 Tackling Noise in Fitness Evaluation

Many real-world optimization problems are noisy, e.g. because a stochastic simulation is used for fitness evaluation, because evaluation is done by physical experiments and there is measurement noise, or because the fitness function depends on uncertain data. Noisy fitness functions are a challenge for evolutionary algorithms (EAs), because they impact an EA’s ability of selection, i.e., its ability of correctly identifying the better individuals. This can have a detrimental effect on the performance of the EA, leading to slower convergence to poor solutions [9, 16]. Various researchers have proposed different methods to improve the performance of EAs in noisy envi-

ronments, for surveys see e.g. [56, 77]. The simplest option is to reduce the effect of noise by evaluating each solution multiple times, and using the average fitness value for selection. However, this is obviously computationally expensive. To reduce the computational cost, one option is to use techniques from ranking and selection to allocate evaluations to individuals, and allocating evaluations in a way that maximally informs the selection process. [85] was the first paper to integrate Optimal Computing Budget Allocation into evolutionary algorithms, and proposed a general framework. Other examples include [96, 53]. Ranking and selection techniques have also been combined with multi-objective EAs [61, 92] and other metaheuristics such as particle swarm optimization [6, 99].

However, this previous work integrating efficient sampling methods into EAs have been for standard noisy selection case where individual alternative fitness samples can be independently obtained. Evolutionary methods have also been applied to a variety of optimization problems with pairwise fitness evaluations, for example the evolving of Artificial Neural Network players for Checkers [22, 23], or Poker [63, 64]. Pairwise fitness evaluations also frequently occur in Co-evolution, a subset of Evolutionary Algorithms where the fitness of alternatives is defined only relative to their peers, for example in the evolution of dueling robots that try to overpower each other [90], or compete for a limited resource like catching a virtual ball [88]. In this chapter, we discuss the integration of the efficient pairwise sampling method into the state-of-the-art Evolutionary Strategy CMA-ES, by applying the POCBAm from Chapter 3.

4.3 CMA-ES

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [51] is one of the most popular Evolutionary Strategies, with state-of-the art performance on a range of derivative-free optimization tasks. Many different forms of CMA-ES exist, each

adapted to improve performance on a particular class of objective functions, for example Sep-CMA-ES [80], with diagonalized covariance matrices for separable problems, LM-CMA-ES [67] for higher dimensional optimization, or a version for discrete feature spaces [8]. In this chapter, we utilize only the standard form of CMA-ES as described in [49], with full covariance matrix and both evolution path (cumulation) and full rank updates. Here we give a brief overview of the selection and update steps of the optimizer. A full description and justification, can be found in [49].

Each generation, CMA-ES generates a new population of K alternatives (a_1, \dots, a_K) by sampling from a multivariate Gaussian distribution. i.e at the g^{th} time-step:

$$a_i^{(g)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)})$$

where $\mathbf{m}^{(g)}$ is the mean vector, $\mathbf{C}^{(g)}$ the covariance matrix, and $\sigma^{(g)}$ and overall step-size parameter. Next, the fitness of the alternatives (a_1, \dots, a_K) is measured, and high fitness alternatives are recombined to determine the distribution of alternatives at the next time step. This distribution update is purely rank-based, i.e the actual fitness function values of the alternatives are not used, only their relative values. Additionally, only a subset of alternatives with the k highest fitness values are typically selected and used with equal weighting, with the other $(K - k)$ low fitness alternatives discarded. Therefore the challenge of the fitness evaluation step is not to find the most accurate estimates of the alternative fitnesses, or to provide the most accurate total ordering on alternatives, but purely to identify the best top- k subset, with highest confidence. Wherever the fitness estimated are affected by noise and must be obtained through pairwise sampling, we can use POCBA to efficiently allocate our available samples to obtain the best solution to this top- k selection problem.

Given a top- k subset (a_1, \dots, a_k) , the standard CMA-ES distribution update uses Equations 6,22,27,28 and 34 from [49]:

Mean update:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^k a_i^{(g)} \quad (4.1)$$

The covariance matrix update is a combination of the rank-1 evolution path update, which utilizes correlations between generations based only the shift in the mean of the distribution and the rank-k update, which uses the empirical covariance of the selected top- k subset in each generation. Evolution path update:

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_v(2 - c_c)} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$$

Covariance update:

$$\mathbf{C}^{(g+1)} = (1 - c_1 - c_\mu)\mathbf{C}^{(g)} + c_1 \underbrace{\mathbf{p}_c^{(g+1)}\mathbf{p}_c^{(g+1)T}}_{\text{rank 1 update}} + c_\mu \underbrace{\sum_{i=1}^k \frac{1}{\sigma^{(g)^2} \mathbf{x}_i^{(g)} - \mathbf{m}^{(g)}} \left((\mathbf{x}_i^{(g)} - \mathbf{m}^{(g)}) \right)^T}_{\text{rank k update}} \quad (4.2)$$

CMA-ES constructs another evolution path measurement to control step size, called the conjugate path. When the movement of the distribution (length of the conjugate path) is long relative to its expected length under random selection, it indicates that the individual steps made by the optimizer are correlated. Given that the steps are similar, increasing the step size thereby allows the optimizer to shift its distribution the same amount in fewer generations. Similarly, if the conjugate path is short, individual steps are anti-correlated and the step size should be reduced. Conjugate path used to control step size:

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)} \mathbf{C}^{(g)^{-\frac{1}{2}}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$$

Step size update:

$$\sigma^{(g+1)} = \sigma^{(g)} e^{\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\sqrt{n}} - 1 \right) \right)}$$

Recommended values for parameters $c_1, c_\sigma, c_\mu, d_\sigma$ can be found in Table 1 of [49].

4.4 Experiments on Synthetic Test Functions

In this section, we test the performance of CMA-ES with noisy fitness evaluation, with samples selected using POCSAM and uniform sample allocation on a range of synthetic test functions commonly used to test evolutionary optimization algorithms, adapted here for pairwise sampling.

4.4.1 Test functions

We consider four different test functions. Implementations of the functions along with many other alternative optimization test functions can be found in the Black-Box Optimization Benchmarking suite [50]. The functions we used are:

- The Sphere function:

$$f_{shp}(x_1, \dots, x_n) = \sum_{x=1}^n x_i^2$$

The global minimum is located at $[0, \dots, 0]$. This is the easiest of the test functions: it is smooth, convex, has no local optima and is separable – the global minimum can be located by optimizing along each dimension independently. A visualization of the 2-dimensional Sphere function can be found in Figure 4.1.

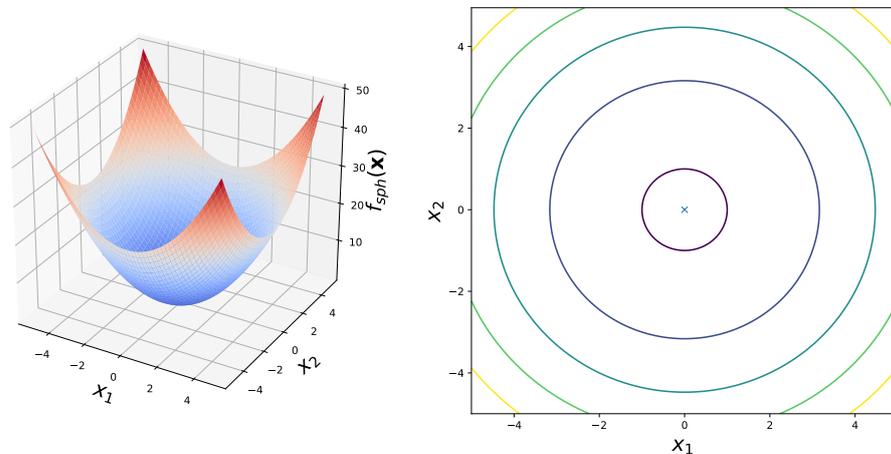


Figure 4.1: The Sphere function. Global minimum at $[0, \dots, 0]$ marked by cross.

- Ackley function [1]:

$$f_{ack}(x_1, \dots, x_n) = -20e^{-0.2\sqrt{0.5\sum_{i=1}^n x_i^2}} - e^{0.5(\sum_{i=1}^n \cos(2\pi x_i))}$$

Global minimum located at $[0, \dots, 0]$. This is a much more challenging test function for optimizers. Away from the global minimum, the overall shape of the function surface is relatively flat, with many local optima. This makes finding the global minimum very difficult for search methods that start in these flat regions. A visualization of the 2-dimensional Ackley function can be found in Figure 4.2.

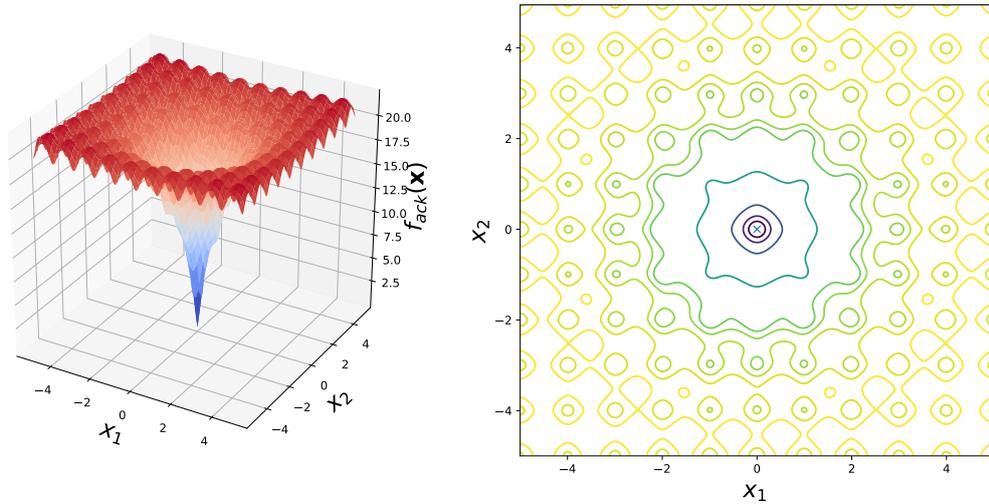


Figure 4.2: The Ackley function

- Rosenbrock function [81]:

$$f_{ros}(x_1, \dots, x_n) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2) + (1 - x_i)^2)$$

The Rosenbrock function has a single global minimum at $[1, \dots, 1]$. It is non-separable and optimization must be performed jointly over all dimensions. In contrast to the Ackley function, the minimum of the Rosenbrock function is

located along a relatively flat valley, with the function rapidly increasing away in one direction. A visualization of the Rosenbrock function can be found in Figure 4.3.

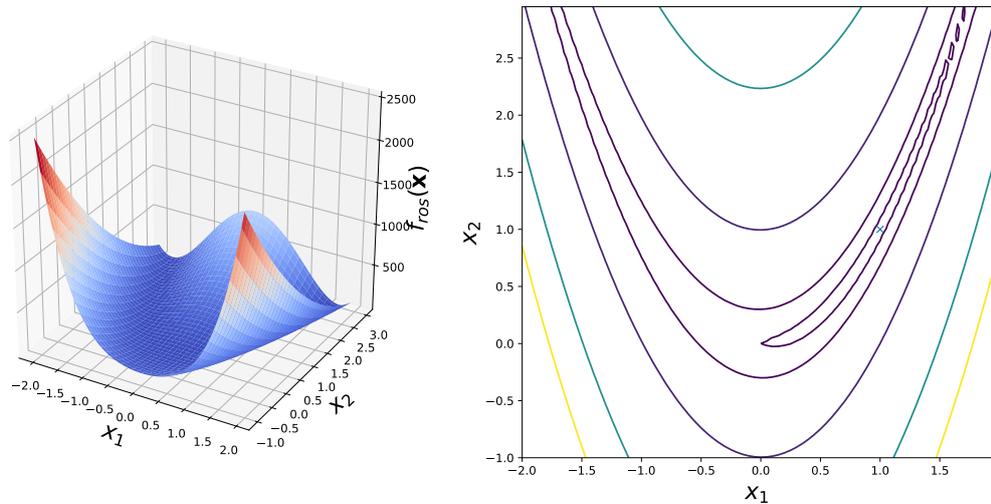


Figure 4.3: The Rosenbrock function

- Rastrigin function [78]:

$$f_{rast}(x_1, \dots, x_n) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i))$$

The Rastrigin function has a global minimum at $[0, \dots, 0]$. Like the Ackley function, the function surface undulates with a large number of local optima. However, unlike the Ackley function, the relative difference in function value at the global minimum and the nearby local minima is relatively small, making this challenging to optimize even when starting close to the global optimum. A visualization of the Rastrigin function can be found in Figure 4.4.

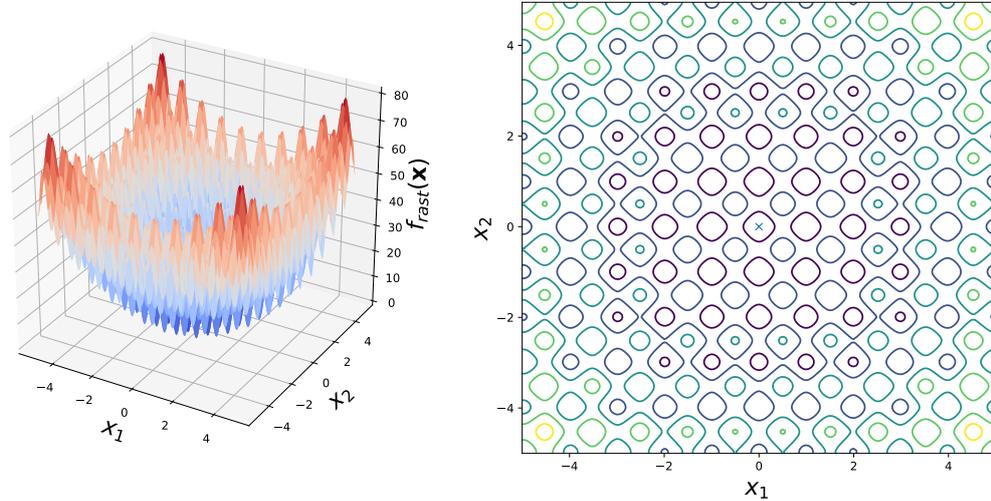


Figure 4.4: The Rastrigin function

To make the fitness evaluations pairwise, we define the pairwise analogs of each of these functions simply by returning the difference of the non-pairwise function values of the alternatives, perturbed by noise. I.e, for a test function $f(\mathbf{x})$ and an alternative pair (a_i, a_j) parameterized by \mathbf{x}_i and \mathbf{x}_j , the pairwise sample function is:

$$p(a_i, a_j) = f(\mathbf{x}_i) - f(\mathbf{x}_j) + \epsilon_{i,j}$$

Where $\epsilon_{i,j}$ is the Gaussian sampling noise, $\epsilon_{i,j} \sim \mathcal{N}[0, \sigma_{i,j}^2]$ for the pair (a_i, a_j) . Clearly, the Borda score estimates for alternatives (if sufficiently accurate) reproduce the correct ordering of the actual test function values for the alternatives.

4.4.2 Results for Single Generation Selection

Figure 4.5 shows the performance of POCBAM and uniform sampling at selecting the top-3 of 6 alternatives for each of the synthetic test functions. Population size and top subset size were chosen using the recommended CMA-ES parameter settings from [49]. Initial alternatives were generated randomly, with each alternative's parameters sampled from a Gaussian distribution with mean 1 and variance 1. The

means of the pairwise sampling distributions were equal to the true value of the pairwise test function $p(a_i, a_j)$, with the variances ($\sigma_{i,j}^2$) of the sampling noise ($\epsilon_{i,j}$) selected uniformly at random for each pair, with ranges corresponding approximately to the scale of each of the test functions in the vicinity of the initial distribution. For the pairwise Sphere and Rastrigin functions, $\sigma_{i,j}^2 \sim \mathcal{U}[0, 25]$, for the Ackley function $\sigma_{i,j}^2 \sim \mathcal{U}[0, 5]$ and for the Rosenbrock function $\sigma_{i,j}^2 \sim \mathcal{U}[0, 500]$. Results in the plots are averaged over 10,000 different replications with different initial populations and noise distributions, common across the two methods, and performance measured by correct selection success rate (Equation 3.12). We observe that POCBAM performs significantly better than uniform allocation across all the test functions after the initial warm-up period. P-values for the final differences in success rate between the methods were less than 0.001 in all four cases.

4.4.3 CMA-ES Performance Across Multiple Generations

In the next experiment, we investigate how the improved selection of POCBAM can improve the performance of the CMA-ES optimizer over multiple generations. During each generation of CMA-ES, the sampling method must select the subset of alternatives used to determine the distribution to generate the next generation. The hope is that improving selection will reduce the cost to the optimizer of sampling noise, thereby leading to faster convergence.

We start by visualizing some example runs of the CMA-ES optimizer on each of the test problems, using POCBAM sample selection, with 500 samples per generation. Figures 4.6 and 4.8 show examples of the convergence of the CMA-ES with initial distribution $\mathcal{N}(1, 1)$ to the global minima of the Sphere and Rastrigin functions respectively. In both cases, the population distribution approximately centers on the global optimum after only a few generations, before the population variance shrinks and the optimizer converges. Likewise, Figure 4.7 shows an exam-

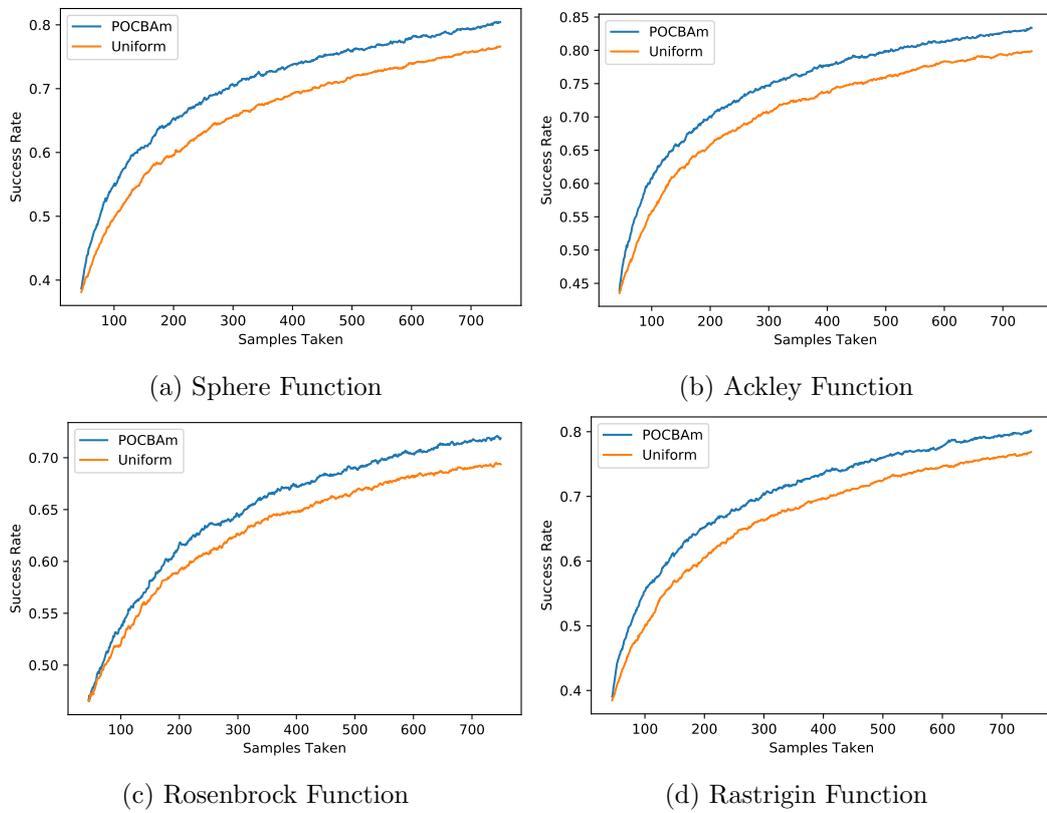


Figure 4.5: Performance of POCBAm and uniform sample allocation at pairwise top-3 of 6 selection of the initial population randomly generated for CMA-ES for different 2D test functions.

ple run of CMA-ES with POCSAm sampling on the Rosenbrock function, in this cases plotting every 3rd generation. We see that the optimizer is able to identify the near-optimal trough relatively quickly, before gradually crawling along the trough toward the global optimum. Figure 4.9 shows an example of how incorrect selection in a single generation can cause CMA-ES to fail to converge to the global optimum. In this example, one of the points selected in the fifth generation is incorrect. As the three selected points are very close together, this significantly reduced the variance of the recombination distribution in subsequent generations, meaning that the optimizer is ultimately unable to escape the nearby local minimum.

Even with noiseless fitness evaluation, it is possible for the optimizer to encounter similar problems. The recombination population is randomly generated, and as such, the alternatives in the top- k set may be arbitrarily close to one another with non-zero probability. If they are too close, the distribution of subsequent generations may retain insufficient variability to locate the global optimum. The form of the covariance update in CMA-ES (Equation 4.2) helps to mitigate this risk by retaining a dependence on all prior distributions.

Without sufficiently good selection the optimizer cannot hope to converge correctly: Figure 4.10 shows the performance of CMA-ES on the Sphere function where the top- k subset is selected at random with initial population distribution $\mathcal{N}[5, 5^2]$, averaged over 10,000 replications. The optimization error shown on the y-axis of the figure is the difference in value of the objective function at the mean of the CMA-ES population distribution for each generation, and the global optimum. Although the expected sample mean of the randomly selected top- k subset is equal to the mean of the original distribution, the cumulation or evolution path component of the CMA-ES update has the effect of giving momentum to the empirical movements of the recombination mean. Over time this causes the distribution to drift away from

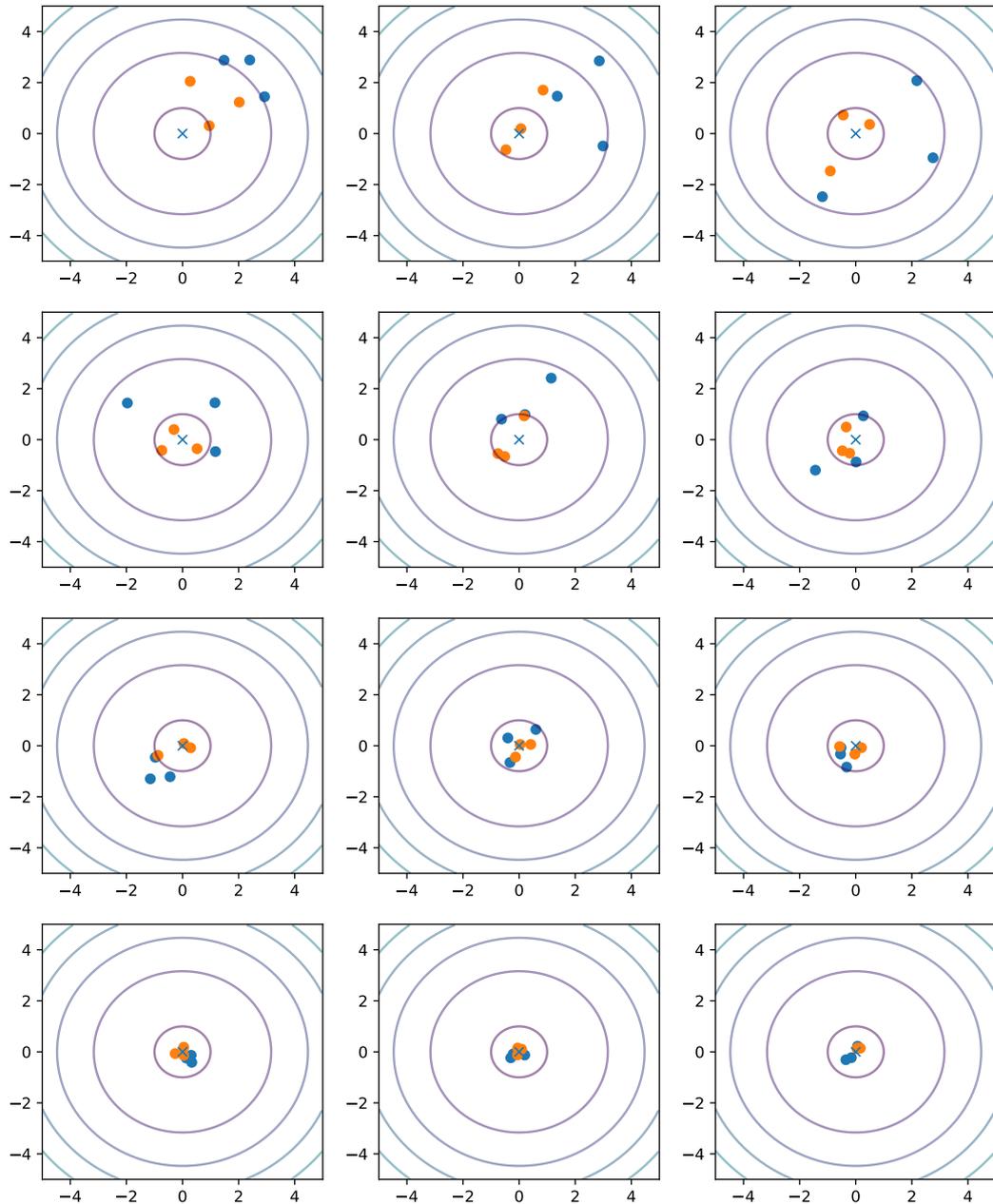


Figure 4.6: Example of the convergence of the CMA-ES optimizer on the 2D Sphere function with noisy pairwise sampling. Samples chosen using POCBAM, with the selected top- k individuals for each generation highlighted in orange. Global optimum highlighted by x .

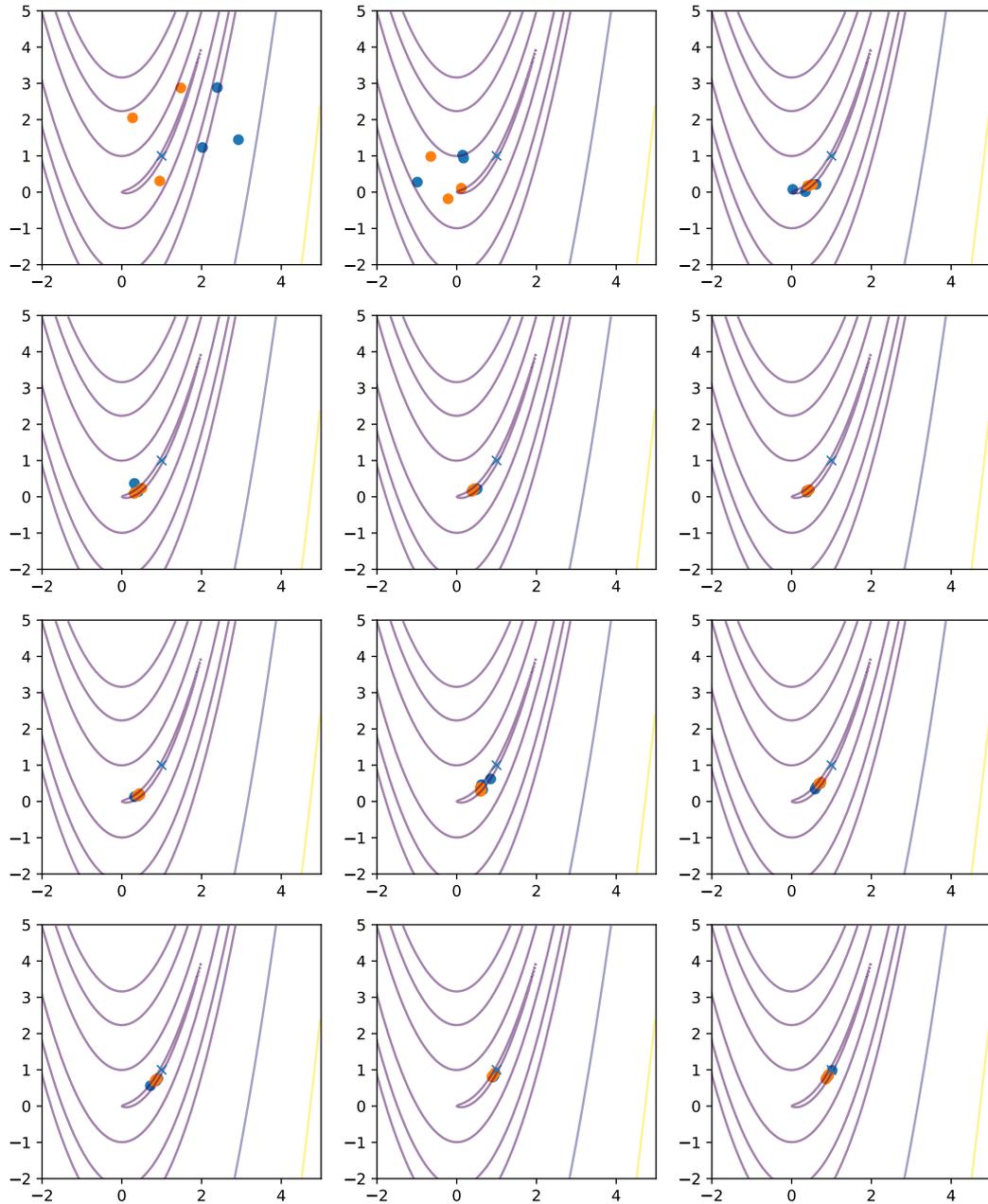


Figure 4.7: Convergence of CMA-ES on the 2D Rosenbrock function with noisy pairwise samples selected using POCBAM. We see that the optimizer is able to identify the near-optimal trough, before steadily traversing towards the global optimum. In this figure every only 3rd generation is shown.

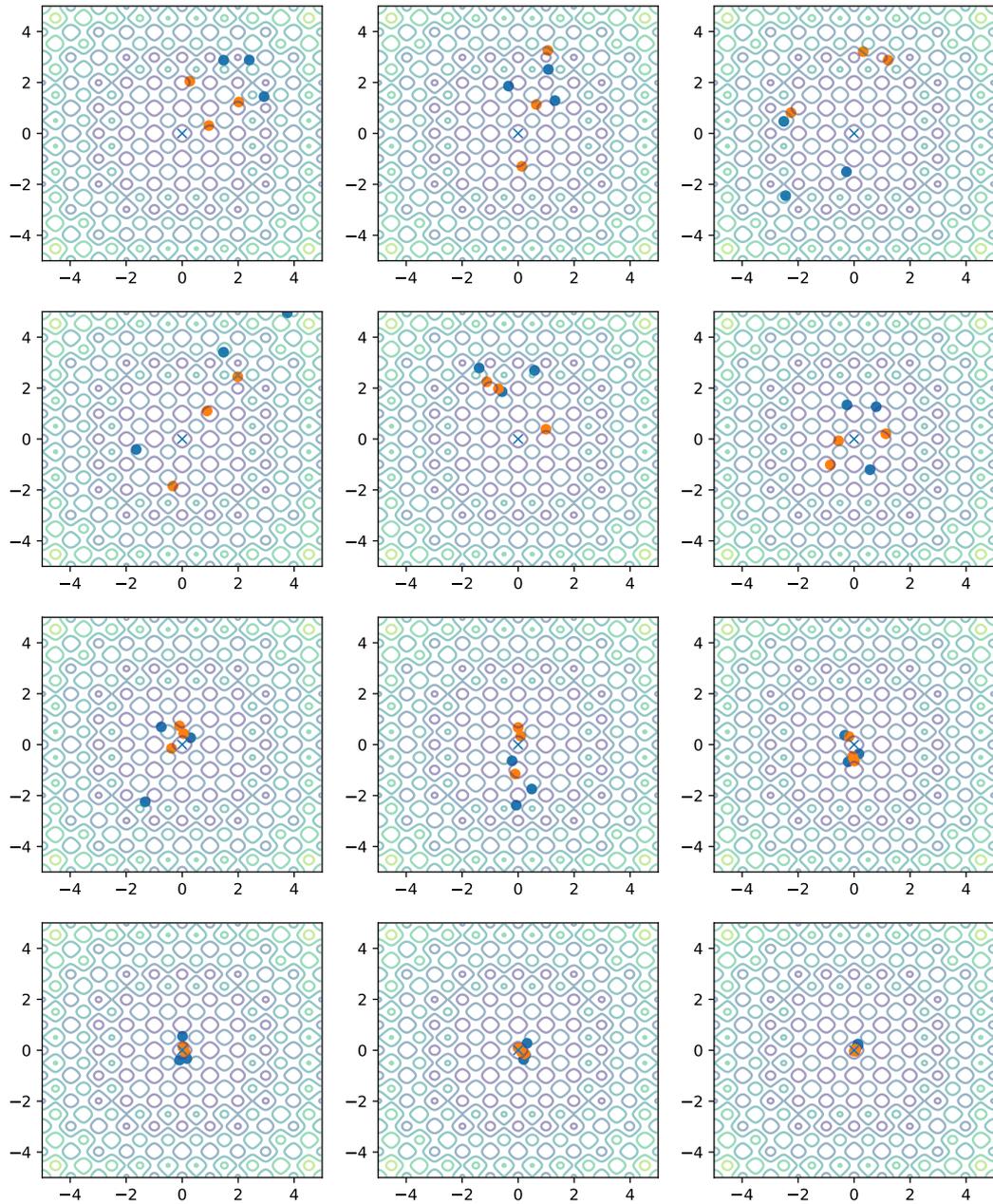


Figure 4.8: Convergence of CMA-ES on the 2D Rastrigin function with noisy pairwise samples selected using POCBAM.

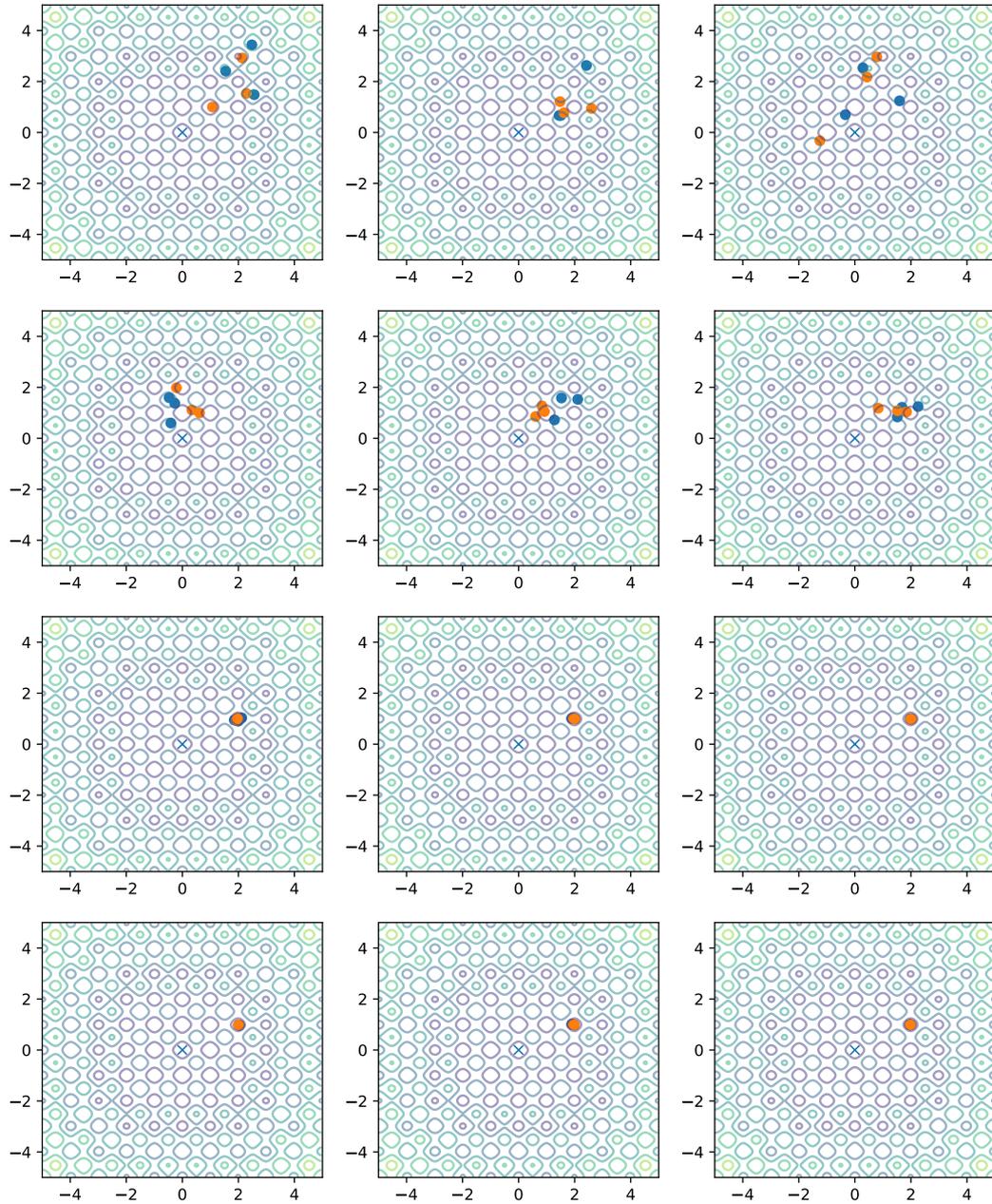


Figure 4.9: Generally stochastic optimization methods like CMA-ES should be more resilient to local optima than gradient-based methods. In this figure, we see an example of a case when CMA-ES becomes trapped in a local minimum. In generation 5 POCSAM selects 3 points that are very close to each other, greatly shrinking the variance of the recombination distribution. Without sufficient variability in the subsequent generations, CMA-ES was unable to escape the pull of a nearby local minimum.

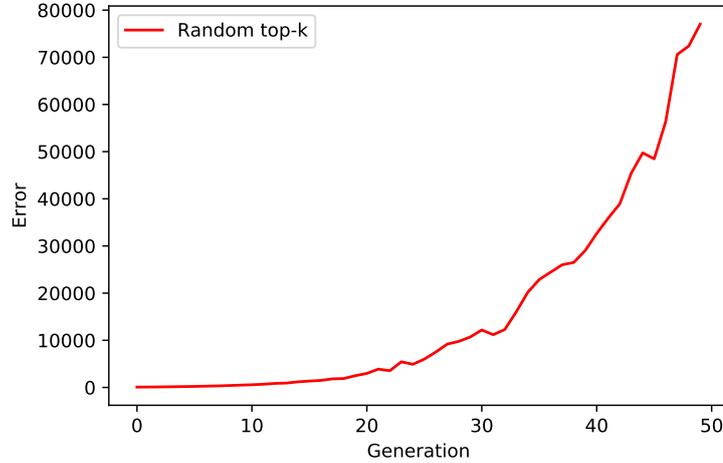


Figure 4.10: Performance of CMA-ES on the Sphere function with random top- k selection. On the y-axis, Error is the difference between objective function value at the mean of the optimizer’s distribution and the global optimum.

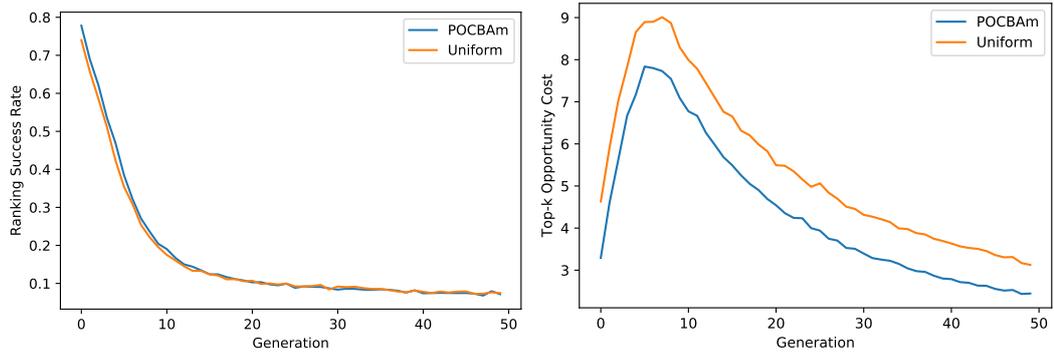
the global optimum.

To investigate the effect of noisy fitness evaluation on performance of CMA-ES on the pairwise test functions, and to evaluate the performance of CMA-ES with POCSAm against uniform sampling, we again tested the convergence of the optimizer on the four test functions, averaging over a large number of replications and comparing against CMA-ES with noiseless fitness evaluation (“Oracle” method). Each generation we record the success rate of each method at top- k selection, the top- k “opportunity cost” and the optimization error. The top- k opportunity cost is defined as the cumulative difference in objective function value between the top- k subset selected by the sampling method, and the true top- k subset for each generation. This gives us a measure of the magnitude of the mis-selection errors that each of the sampling methods make due to noise. Each sampling method was tested using a fixed budget of 300 samples per generation (equivalent to 20 samples per pair if uniformly allocated). The CMA-ES optimizer was allowed to run for 50 generations, with the results averaged over 10,000 replications. Initial populations in each replication were sampled from $\mathcal{N}[5, 5^2]$, and the sampling noise variances for

each test function were the same as in the single generation experiments (Subsection 4.4.2) for the uniform and POCBAm methods, and zero for the Oracle method.

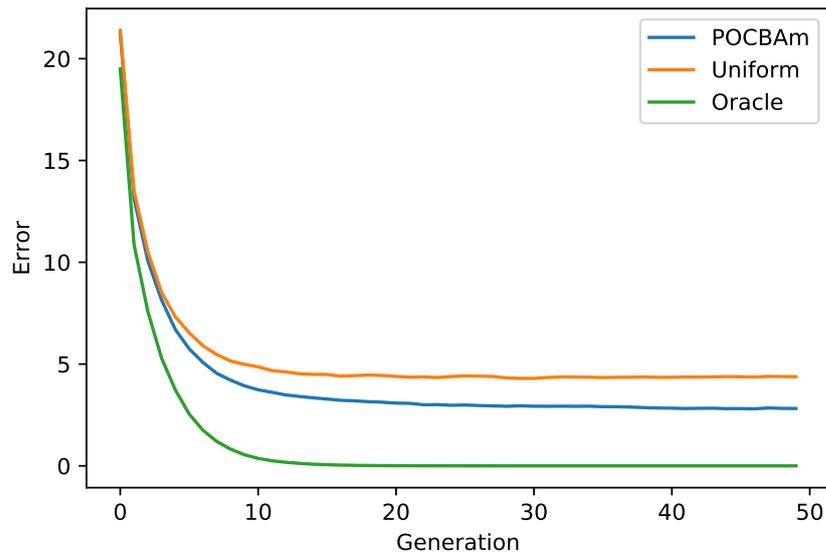
Figure 4.11 shows the results for the pairwise Sphere test function. Interestingly, the difference in selection success rate between the two methods was small, particularly after the first few generations. As the CMA-ES distribution approaches the optimum, the fitness differences for alternatives become very small relative to the noise, resulting in decreasing selection success over time. However, the difference in top- k opportunity cost remained relatively constant after the first 5 generations, peaking for both methods around the elbow of the convergence error plot, with POCBAm achieving lower opportunity cost throughout. This suggests that although both methods make errors with approximately the same frequency (after a few initial generations), the selection errors made by POCBAm are generally smaller than those made by uniform sampling. The effect of this was that optimization error for CMA-ES with the POCBAm method was significantly lower ($P < 0.001$) than with uniform sampling, achieving lower error on average after 8 generations than the uniform sampling method achieved after 50, an 84% reduction in sampling cost. However, the error was still substantially higher than CMA-ES with noiseless fitness evaluation. With perfect fitness evaluation, the Oracle method was able to converge very fast, generally reaching the global optimum after around 12 generations. Clearly, although the improved fitness evaluation of POCBAm was of some benefit in mitigating sampling noise, the effect on the convergence of CMA-ES is still large.

Figure 4.12a shows the performance of the sampling methods on the pairwise Ackley test function. This is clearly a more difficult test function than the Sphere to optimize, with the Oracle method taking much longer to converge than on the previous plot. When the optimization error is plotted on a log axis, we can see an indication of the different phases of the optimizer: away from the global



(a) Selection Accuracy

(b) Selection Opportunity Cost



(c) CMA-ES performance

Figure 4.11: Performance of CMA-ES with POCBAm and uniform sample selection over 50 generations on the pairwise 2D sphere function. Sub-figures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Subfigure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum. The “Oracle” method in sub-figure (c) shows CMA-ES performance with perfect, noise-less fitness evaluation.

optimum, the surface of the function is relatively flat, albeit with many small local minima. Consequently, the initial convergence rate of CMA-ES is quite slow, before accelerating when the population distribution finally locates the deep pit around $[0, \dots, 0]$. The difference in selection success rate between POCBAM and uniform is again relatively small, but remains visible for more generations. Again the difference in opportunity cost between the methods remains fairly constant, peaking around the elbow of the convergence error curves. CMA-ES convergence with POCBAM is again better than with uniform ($P < 0.001$), but by a smaller margin than on the Sphere function, here taking on average 27 generations to surpass the performance of uniform after 50, or a sampling reduction of 46%.

Figure 4.13 shows the result for the pairwise Rosenbrock test function. Initial performance of the two sampling methods appears very similar, while the population distribution is in very steep regions of the Rosenbrock function away from the optimum. As the CMA-ES distributions move into the flatter region of the test function, the selection success rates for both methods remains similar, but the top- k opportunity cost and convergence error of the CMA-ES with POCBAM sampling is significantly lower ($P < 0.01$), reaching the same error as CMA-ES with uniform sampling after 66% fewer samples.

Finally, Figure 4.14 shows the results for the pairwise Rastrigin function. On this function, there is a clear success rate difference between the two methods across all 50 generations, with a corresponding difference in top- k opportunity cost and a growing difference between the convergence error. The Rastrigin function appears to be challenging for the optimizer, with the average performance of even the noiseless CMA-ES still failing to reach the optimum after 50 generations. However, POCBAM is still significantly better than uniform ($P < 0.01$), and reaches the same error rate as uniform after 50 generations with 22% fewer samples.

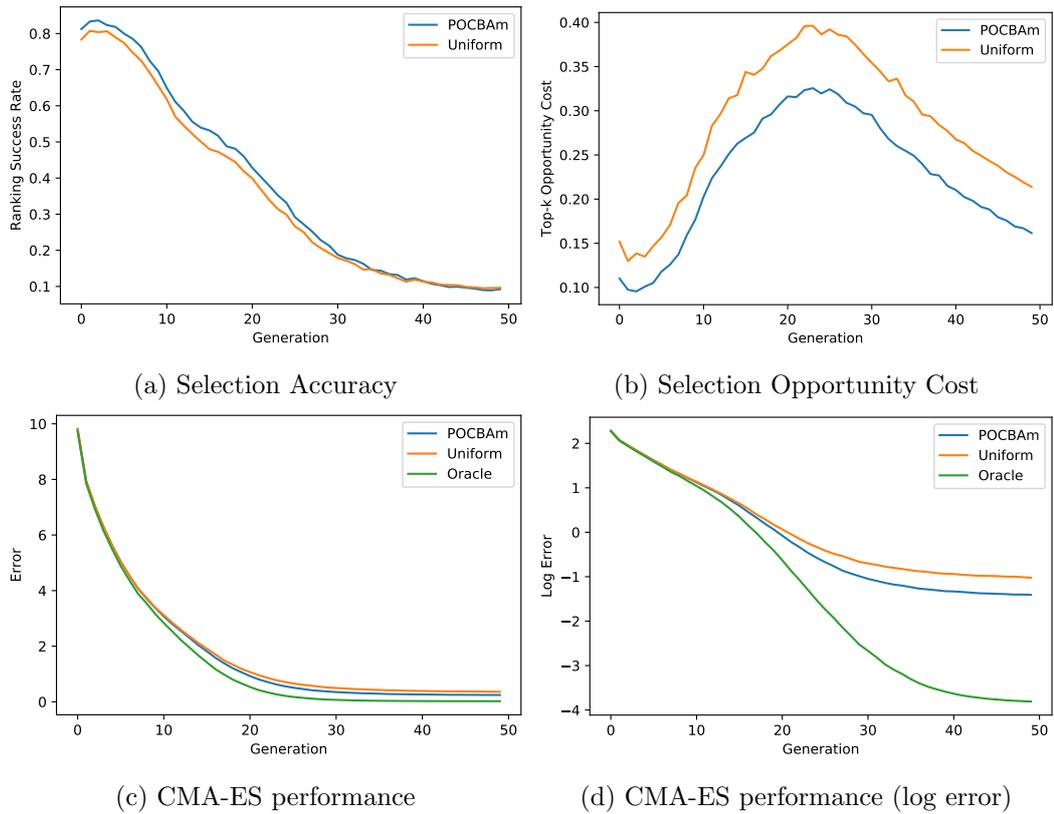
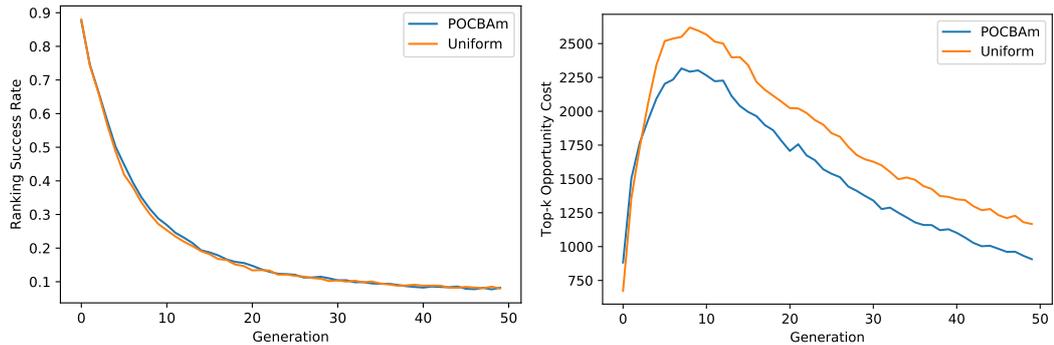
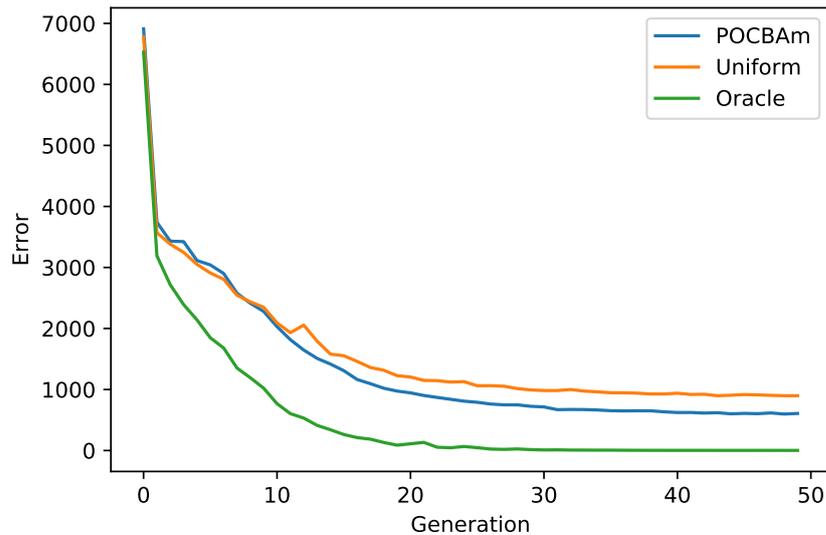


Figure 4.12: Performance of CMA-ES with POCBAm and uniform sample selection on the pairwise 2D Ackley function. Sub-figures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Subfigure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum, with subfigure (d) showing the same result plotted on a log scale. As before, the “Oracle” method shows CMA-ES performance with perfect, noise-less fitness evaluation.



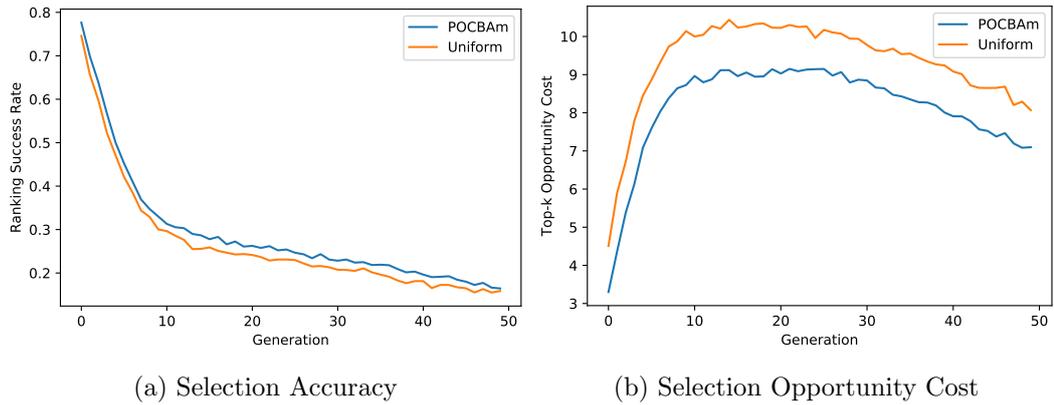
(a) Selection Accuracy

(b) Selection Opportunity Cost



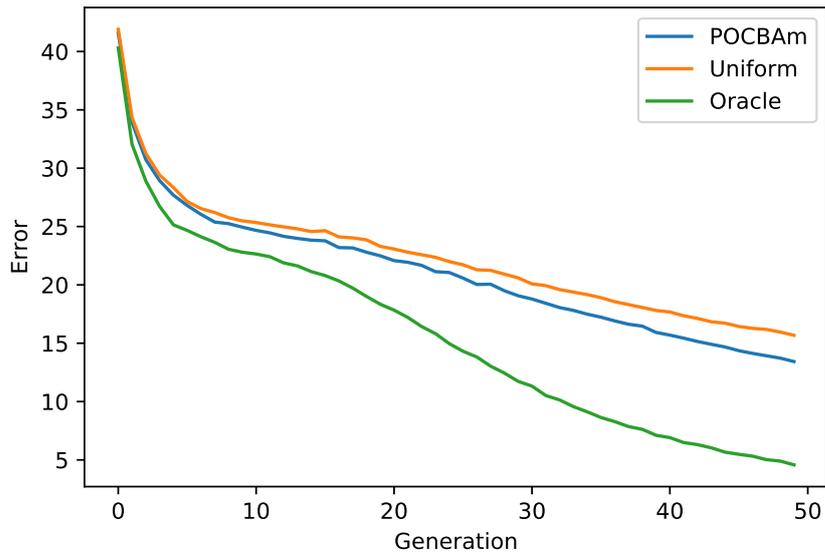
(c) CMA-ES performance

Figure 4.13: Performance of CMA-ES with POCBAm and uniform sample selection on the pairwise 2D Rosenbrock function. Sub-figures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Subfigure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum. The “Oracle” method in sub-figure (c) shows CMA-ES performance with perfect, noise-less fitness evaluation.



(a) Selection Accuracy

(b) Selection Opportunity Cost



(c) CMA-ES performance

Figure 4.14: Performance of CMA-ES with POCBAm and uniform sample selection on the pairwise 2D Rastrigin function. Sub-figures (a) and (b) show the top- k ranking accuracy (success rate) and opportunity cost (difference between selected and ideal function values) respectively. Subfigure (c) shows the difference between the objective function value at the CMA-ES distribution mean for each generation, and at the global optimum. The “Oracle” method in sub-figure (c) shows CMA-ES performance with perfect, noise-less fitness evaluation.

4.5 Evolving Poker Playing Agents

In this section, we consider a more realistic test optimization problem that might be tackled using CMA-ES. Recent work [63, 64] has considered the development of highly skilled automated Texas Hold'em playing agents, using an evolutionary strategy to optimize the weights of a recurrent neural network based player model. Their ES approach is slightly different to CMA-ES, but still uses top- k selection. Each generation, a top performing portion of the population of playing agents is identified, which are then used to produce parameters for a new population of players for the next generation. The key difference to their method is the inclusion of elitism, where a small percentage of the top players from each generation is preserved to the next.

In contrast to the simple test functions in our earlier experiments, simulating poker games between players is computationally expensive as many hands need to be played between players to overcome the stochastic effects of the card shuffle. Improving the accuracy of the top player subset or reaching the same level of accuracy with fewer samples can both be valuable, either by reducing the computational cost of the evolutionary optimizer per generation, or increasing the speed of convergence, thereby reducing the number of simulations required.

4.5.1 Texas Hold'em poker

No limit Texas Hold'em (NLTH) is a challenging game with a vast state space of approximately 7×10^{75} states [57]. It has been extensively studied by researchers and is considered a useful non-trivial example for imperfect information, stochastic games [10]. Many different approaches have been taken for developing computer poker players, see [83] for an overview. A lively annual computer poker competition is held each year at the AAAI conference [66].

For our pairwise player experiments, we consider only 2-player NLTH, com-

monly known as “Heads-up” NLTH. NLTH is played with a standard 52 card deck. Play is structured into rounds, known as hands, during which players receive cards, place bets and either win or lose chips (tokens typically used to represent money). Play typically continues either for a fixed number of hands, or until only one player has chips remaining. At the start of a hand, the cards are shuffled into random order, and two cards are dealt to each player face-down, so that only the player may see her own cards. Players must place an initial bet, known as the blinds, with one player placing a smaller fixed size bet (the small blind) and the other a larger fixed sized bet (the big blind) into a collective pot. The play then proceeds in turns beginning with the small blind player. On each turn, a player may choose either to discard her cards (fold), ending the hand and returning all bets currently in the pot to her opponent, or to match the amount currently bet by her opponent (call, or check, if no additional money is needed to match the opponent’s bet), or finally to increase the bet by at least a minimum amount (raise). In Heads-up NLTH, a betting round continues until a player chooses not to raise. If neither player folded during the initial betting round, three “community” cards from the deck are dealt face-up (flop). These may be used by any player along with their hidden cards to form one of several ranked combinations (also called hands). This is followed by another betting round, whereupon, so long as neither player folded, another community card is revealed (turn). After which another betting round takes place. If neither player has folded, a final community card is revealed (river) and a final betting round occurs, and, if neither player has folded, the players reveal their hidden cards (showdown) and the player with the highest ranked combination of cards wins the pot. The players then swap roles (big and small blind) and proceed with the next hand. More detail on hand rankings, rule variations and strategies can be found in many popular published works, for example [89].

4.5.2 Poker player model

We generate poker players using the player model described in [64]. This represents, to the best of our knowledge, the current state-of-the-art player model for evolved no limit Texas Hold'em agents. Each player consists of three main components, a pattern recognition tree (PRT) that records the frequency of betting patterns observed during the game, along with opponent fold frequency and showdown win frequency for each betting pattern. To restrict the growth of the tree, opponent bets are discretized into seven different size buckets, and the agent's own bets are restricted to 5 different sizes (0.5x pot, 1x pot, 1.5x pot, 2x pot and all in). The data recorded in the PRT allows the agent to identify patterns in the opponent's play (like for example the tendency to fold after the agent makes a large bet) and is used to provide information for the other components of the player model: At each decision point during a hand, the agent looks up the statistics from the PRT for the betting sequence that would result from each possible action and forwards them to the other key player model components – the opponent fold rate estimator (OFRE) network, and the showdown win rate estimator (SWRE) network. The OFRE and SWRE components are each formed of an initial layer of recurrent LSTM blocks [42] whose outputs feed into a smaller, fully connected feed forward head [60], with single output node, whose activation is taken to represent the estimated probability. These probability estimates are then used to determine the player's action by a simple decision rule that attempts to maximize the player's expected utility for the hand, as described in [64]. The hidden states of the LSTM blocks for the estimator networks are reset after each hand. Each takes a set of input features from the PRT and the observable game state, shown in Table 4.1. This architecture is designed to allow the player to identify long-term trends in opponent behavior through the information stored in the PRT, with the LSTM blocks retaining short-term sequential information for better decision making within each hand. The task

Table 4.1: Input features used in the OFRE and SWRE networks (as used in [64]).

Feature	Description	OFRE	SWRE
Normalized state frequency	Measure of how often the game state has been observed	✓	✓
Opp. fold rate for state	What proportion of hands from this game state did the opponent fold.	✓	✓
Showdown rate	What proportion of hands from the game state reached showdown.		✓
Expected opp. hand strength	Average strength of opponent’s hand in showdowns from this game state.		✓
Flush and straight draw	Probability of a random hand making a flush or straight given the current communal cards. Estimated through Monte Carlo (MC) simulation	✓	
Pair	Number of paired communal cards.	✓	
Betting round	Which betting stage the game state is (pre-flop, flop, turn, river).	✓	✓
Hand strength	Probability of the player’s hand beating a random hand given the communal cards. Estimated through MC simulation.		✓
Opp. bet	Total amount bet by opponent this hand.	✓	✓
Player bet	Total bet by the player this hand.	✓	✓

of the evolutionary strategy is to optimize the weights of the OFRE and SWRE networks, to find a configuration that is able to accurately estimate and therefore exploit opponent action probabilities.

4.5.3 Experiments

We test the effect on evolved player quality of different sample selection methods for CMA-ES population fitness evaluation. Each generation, the sample selection

methods were allowed to allocate a budget of 2000 short Head-up NLTH games between chosen pairs of players, with each game consisting of only 20 poker hands. To reduce variance, the deck shuffles were mirrored for each position, i.e. the two players have the same 10 sets of hands as each other in each position (small or big blind). The player's chips stacks were configured according to the AAAI annual poker competition rules (<http://www.computerpokercompetition.org>). Given the large state space and high variance of NLTH poker games, this sampling budget is very small relative to the level of noise of the games, making the top- k selection task of the sampling methods very challenging. All CMA-ES parameters were set using the default recommended values in [49]. The population size used was 22, with the top- k subset selected each generation having size 13.

Due to the high computational cost of the evolutionary runs, we could perform a total of only 20 replications of runs of the CMA-ES optimizer, 10 using POCBAm sample selection, and 10 using uniform sampling. Despite this small number of replications, this required simulating a total of 80,000,000 hands of poker, with multiple quantities estimated via MC simulation required for each player several times during each hand.

Figure 4.15 shows the average performance of the players in each CMA-ES population for each sampling method, compared against the final players from the 2000th generations of the evolving runs that used POCBAm sampling. Players from the early generations for both sampling methods lose heavily against the final POCBAm evolved players, with similar losses for the first 400 generations. After this, the POCBAm runs appear to improve faster, with significantly better performance after about 1750 generations. However, due to the small number of replications, the result appears very noisy.

We also tested the performance of the evolved players against a fixed set of benchmark opponents. We defined five different benchmarks, each of which plays a

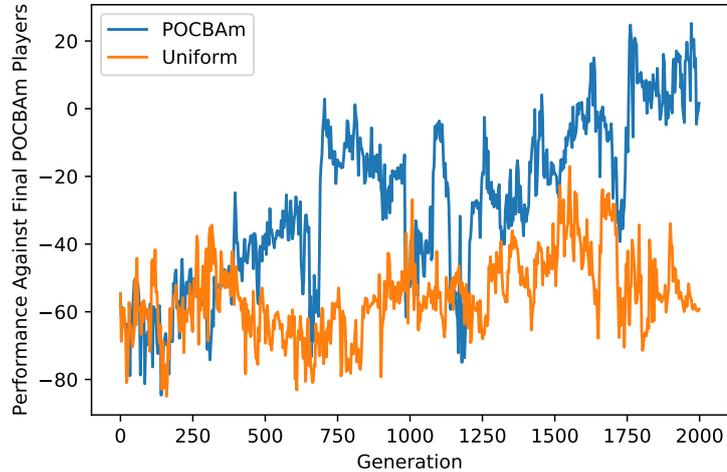


Figure 4.15: Performance of the evolved players from each of the 20 CMA-ES runs against the final 10 players from the 2000th generation of the CMA-ES runs that used POCBAM sampling. Y-axis scale is in average Big Blinds won per Hand (BB/H), averaged over 1000 hands per pair.

strategy based only on their own hand. All five of these benchmark players should be highly exploitable, but each provides different signals that can be learned, and a different complexity of strategy. This means that for the evolved players to successfully exploit all of them, they must be able to learn and identify a variety of different patterns. A summary of each of the benchmark players can be found in Table 4.2.

Figure 4.16 shows the performance of players evolved using each sampling method against the different benchmark players. In general, we were not able to reproduce the strong performance of the evolved players using the sample player architecture shown in [64]. In that work, their players were able to significantly beat all players in a similar benchmark set, but here the only benchmark consistently beaten by the evolved players was the trivial Always Fold fixed player. The performance for the evolved players for both sampling methods against Benchmarks 2 and 3 was not significantly different from zero, although the POCBAM evolved players generally seems to be decreasing over time. Benchmarks 4 and 5 both beat

Table 4.2: Description of the fixed benchmark opponents

Benchmark	Description
1: Fold Only	The most trivial benchmark opponent. Maximizing the exploitability of this benchmark only requires the evolved player to learn not to fold on their turn to act first.
2: Fold/Call	This player simply calls any bet or folds based on estimating the strength of their hand against a random hand, given the table cards. Hand strength is estimated through MC sampling.
3: All-in post flop	This player is similar to the second Benchmark, folding weak hands or calling pre-flop, then post flop either betting all-in or check/fold according to whether their hand strength against a random opponent hand exceeds a threshold level.
4: Statistical Player	A slightly more complicated strategy, this benchmark player attempts to size their bets proportionally to their estimated hand strength, folding weak hands and making progressively larger bets with strong ones. This is still highly exploitable if the evolved player can learn to connect the opponent's bet to their hand strength.
5: Statistical bluffer	Similar to Benchmark 4, but with potential to "buff", occasionally and randomly over or under betting weak and strong hands respectively.

both sets of evolved players by significant margins. Overall, the evolved players appear to be very weak. However, when comparing the relative performance of the evolved players against the benchmarks, there appears to be some evidence that the set using POCBAm do improve more over time, particularly against Benchmarks 1,4 and 5. Statistical tests for the performance differences are shown in Table 4.3.

Evolving neural network weights is a difficult task for evolutionary optimizers. Such networks generally have a high number of inter-related weight parameters, resulting in high-dimensional, non-separable optimization problems. The total number of network parameters (1052) in each of our evolved players is tiny relative to the size of networks often trained using modern, gradient-based optimization methods [60], but would be considered very large for the standard form of CMA-ES. Therefore, it is perhaps unsurprising that the evolved players were not able to achieve high performance. However, reducing the size of the OFRE and SWRE network components reduces the capacity of the networks to learn and recognize different patterns and strategies. [64], who propose the player model that we have used, and who have successfully evolved strong players, do not include the exact specifications for the network architectures they use.

Table 4.3: Average performance difference between the final POCBAm and uniform evolved players against each of the five fixed benchmark players. P values are the result of a two-sided t-test.

Benchmark	Performance Difference (Std Err.)	P
1	0.1210 (0.01294)	<0.0001
2	-3.672 (7.911)	0.643
3	0.3080 (6.182)	0.960
4	8.0217 (5.494)	0.144
5	14.30 (4.915)	0.00362

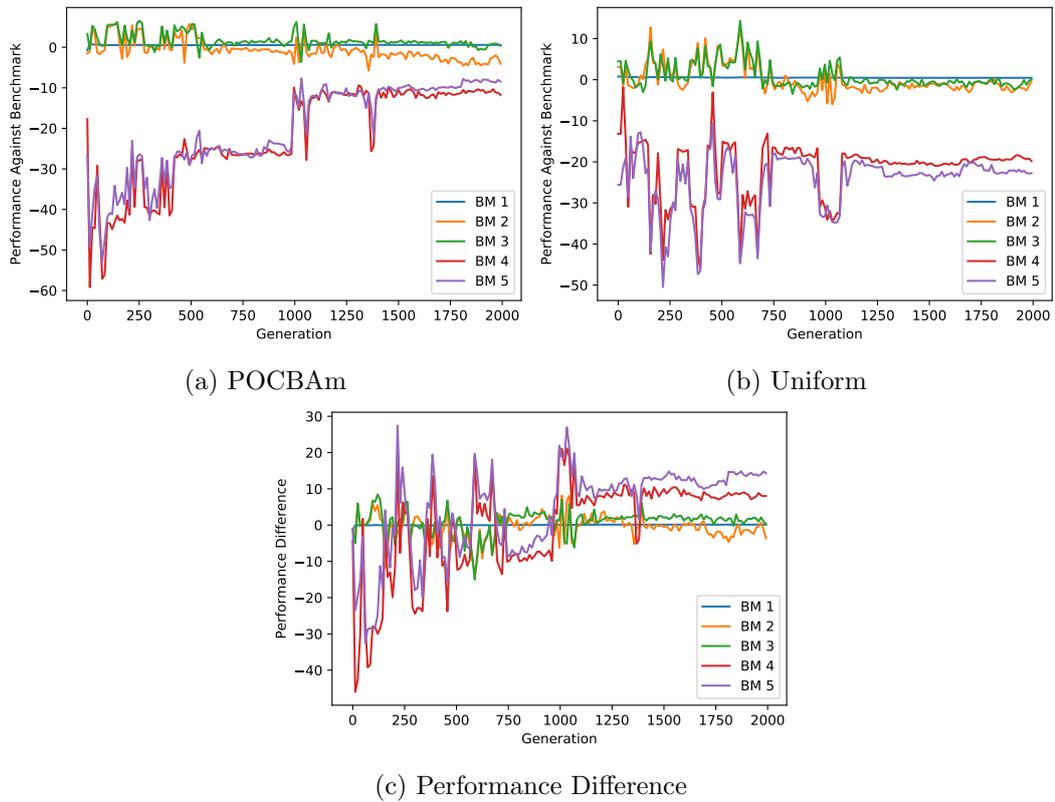


Figure 4.16: Performance of the evolved poker players for each sampling method against each of the 5 benchmark players, average over the 10 replications for each method, with each combination of benchmark and evolved player playing each other for 1000 hands. Subfigure (a) shows the results for POCBAM and subfigure (b) shows the results for uniform sampling. Subfigure (c) shows the difference in performance for the POCBA and uniform evolved players. Scale is in Big Blinds won per Hand (BB/H).

4.6 POCBAM With Different Objectives

The version of POCBAM proposed in Chapter 3 and used in the experiments in the current chapter selects the sampling pair that myopically optimizes an estimate of the expected increase in probability of correct selection (PCS) at each step. The objective of this method is to maximize the probability that the recommended top- k subset matches exactly with the true top- k set. However, not all errors in top- k selection are of equal severity – a recommended set that contains all but one of the true top- k should be preferable to one that is entirely wrong, but both would be equally penalized by the Correct Selection metric. As we observe in Figures 4.11 and 4.14, the difference in the PCS of POCBAM and uniform is negligible after the first few generations, but the top- k opportunity cost of POCBAM remains significantly lower and the performance gap between the two optimizers increases over time. This suggests that reducing the Pairwise Opportunity Cost (OppC) may be a more relevant objective for the sampler to optimize when selecting samples. Adapting POCBAM to use expected opportunity cost is simple. Using the same OCBA framework as in Chapter 3, we can estimate the expected opportunity cost of mis-selecting an alternative a_p using the same expected post sample alternative score distributions $\tilde{\mathcal{S}}_p^{i,j}$ as the expectation of the component of the post sample score distribution that lies on the wrong side of the threshold. I.e:

$$AEOppCS_p^{i,j} = \begin{cases} \int_{-\infty}^c x \mathbb{P}\{\tilde{\mathcal{S}}_p^{i,j} < x\} dx & \text{for } p \in \mathcal{I} \\ \int_c^{+\infty} x \mathbb{P}\{\tilde{\mathcal{S}}_p^{i,j} > x\} dx & \text{for } p \notin \mathcal{I} \end{cases} \quad (4.3)$$

Thus, instead of sampling where Equation 3.7 is maximized, we could select the pair that minimizes

$$AEOppCS^{i,j} = \sum_{p \in \mathcal{I}} \int_{-\infty}^c x \mathbb{P}\{\tilde{\mathcal{S}}_p^{i,j} < x\} dx + \sum_{q \notin \mathcal{I}} \int_c^{+\infty} x \mathbb{P}\{\tilde{\mathcal{S}}_q^{i,j} > x\} dx \quad (4.4)$$

Using this as an acquisition function for POCBAM gives a version that myopically maximizes reduction of the expected magnitude of selection error in terms of Borda score of the top- k subset. As well as PCS and OppC, there are a range of other objectives that may be relevant. For example, when using POCBAM for sample selection in CMA-ES, we may be interested in the effect of mis-selecting the top- k subset on the subsequent recombination distribution of the next generation. This updated distribution, (Equations 4.1, 4.2), depends only on the previous distributions and the parameter values of the top- k alternatives, not their actual fitness values. Small selection errors in terms of OppC, may in fact lead to large changes in the recombination distribution if the mis-selected alternatives are far from the correct ones in alternative parameter space. Likewise, if the fitness function surface is rugged, alternatives with very high and very low fitness values may be close to each other in alternative parameter space. Mis-selecting one for the other may incur a large OppC, but actually have a negligible effect on the evolutionary strategy as the recombination distribution would be almost unchanged. It may therefore be beneficial to try and quantify the severity of selection errors in terms of this distribution. In earlier work, we applied this idea to evolutionary optimization with non-pairwise fitness evaluation, proposing an adapted version of the KG sampler that aims to maximize the expected difference in pre and post-sample recombination distributions in terms of Kullback-Leibler (KL) divergence, with the results published in [44]. Excerpts from this paper describing the KL-KG method and showing empirical results using a simplified version of CMA-ES can be found in Appendix B. Applying a similar principle to POCBAM, swapping a particular alternative out of out index set for another has a fixed effect on the resulting recombination distribution, and, as the distribution depends only on the features of the alternatives in the selected top- k subset, we can calculate both sets of distribution parameters explicitly. Furthermore, as CMA-ES uses a multivariate Gaussian for the recombination distribution, the KL divergence between the two distributions has

an easy to evaluate closed-form expression [33]. Thus, we can account for the importance of potential selection errors to the recombination distribution by weighting the probability of mis-selection (the mass of the estimated alternative score distribution that lies on the wrong side of the threshold c) by the KL divergence between the recombination distribution currently assumed to be correct (based on the current index set) and the distribution that would be used if the alternative score fell on the other side of the threshold. Specifically, POCBAm would then select the sample that minimizes:

$$AEKLD^{i,j} = \sum_{p \in \mathcal{I}} \mathbb{P}\{\mathcal{S}_p^{i,j} < c\} D_{KL}(\mathcal{I}_{p:k+1} || \mathcal{I}) + \sum_{q \notin \mathcal{I}} \mathbb{P}\{\mathcal{S}_q^{i,j} > c\} D_{KL}(\mathcal{I}_{p:k+1} || \mathcal{I}) \quad (4.5)$$

Where $\mathcal{I}_{p:k+1}$ represents the top- k index set with alternative p removed and replaced by the current $(k+1)$ th best and $D_{KL}(\mathcal{I}_{p:k+1} || \mathcal{I})$ is the KL divergence between the CMA-ES recombination distributions arising from $\mathcal{I}_{p:k+1}$ and \mathcal{I} , calculated using [33]:

$$D_{KL}(\mathcal{I}_{p:k+1} || \mathcal{I}) = \frac{1}{2} \log \left(\frac{\det \Sigma_2}{\det \Sigma_1} - d + \text{tr}((\Sigma_2)^{-1} \Sigma_1) \right. \\ \left. + (\mathbf{m}_2 - \mathbf{m}_1)^T (\Sigma_2)^{-1} (\mathbf{m}_2 - \mathbf{m}_1) \right)$$

Where (\mathbf{m}_1, Σ_1) and (\mathbf{m}_2, Σ_2) are the mean vectors and covariance matrices of the CME-ES distributions of $\mathcal{I}_{p:k+1}$ and \mathcal{I} respectively. Minimizing AEKLD encourages POCBAm to allocate more samples to pairs where the uncertainty of the alternative score estimates are high and the divergence from mis-selecting one of the alternatives is also large. This is equivalent to maximizing the expected reduction in divergence from the noiseless recombination distribution under the standard OCBA assumptions.

4.7 Summary

In this chapter, we have explored the application of POCBAm as a sample selection method to improve fitness evaluation in pairwise cases for the CMA-ES evolution strategy. We have shown on a range of empirical test functions that POCBAm can improve the performance of the CMA-ES optimizer when fitness evaluations are noisy. We also investigated how better sampling might improve the quality of NLTH poker players evolved using CMA-ES. Unfortunately, we were not able to train any strong players using this method, but our results suggest that the players evolved using POCBAm sample selection were somewhat better than those evolved using uniform sampling. Finally, we discussed and defined different acquisition functions for POCBAm that could potentially further improve the performance of the evolutionary strategy by either seeking to minimize the magnitude of selection errors in terms of selected alternative fitness, or by considering the effect of selection errors on the recombination distribution. Unfortunately, due to time constraints, we have not yet tested how changing the POCBAm objective affects the top- k selection performance or the performance of the CMA-ES optimizer.

CHAPTER 5

Exploiting Transitivity in Pairwise Preference Models

5.1 Introduction

In this chapter, we consider how to improve the top- k selection performance of the POCBAm method by taking advantage of transitivity between pairwise alternative preferences when present. We focus on the case of selecting the top- $k \geq 1$ alternatives from an available set, where pairwise sampling results represent quantitative information about the degree of preference for one alternative over the other. Just as many preference-based of the dueling bandit methods discussed in Chapter 2 utilize parametric fitness models to encode reasonable assumptions on transitive relations between comparison outcomes and alternative rankings, our proposed method uses a similar approach, adapting the commonly used Thurstone model [93] for the quantitative case. The chapter is organized as follows: Section 5.2 defines the parametric ranking model and describes the proposed ML-POCBAm sampling method in detail. In Section 5.3 we compare the performance of ML-POCBAm against alternative methods on cases where the assumed preference model accurately describes the underlying alternative ranking. In Section 5.4, we investigate the effect of noise and inconsistency in the underlying ranking, and propose a simple adaptation to

ML-POCBAM to improve its robustness in such cases. Our final experiments in Section 5.5 return to the poker player selection problem of Chapter 4, considering the level of transitivity in populations of such players, and whether our proposed method ML-POCBAM can exploit this to improve selection accuracy.

5.2 Maximum Likelihood Pairwise Optimal Computing Budget Allocation

As discussed in Chapter 2, the majority of current literature on ranking and selection using pairwise comparisons considers the case where the $X_{i,j}$ are Bernoulli R.Vs and the pairwise means $\mu_{i,j}$ correspond to the probability that alternative a_i “wins” a comparison against alternative a_j . This is a useful case with many applications, for example in ranking game players when only win/loss game feedback is available, or in online advertising and search engine optimization where a user may be presented with a number of alternatives (web links), and provides binary feedback (clicks on one link). In the Bernoulli case, the pairwise random variables are generally linked by the *shifted skew-symmetry* condition [86], i.e $X_{i,j} = 1 - X_{j,i}$. However, in many applications the pairwise comparison result may yield more information than a simple win/loss. For example in many pairwise games, comparisons are scored, with the score indicating the magnitude of the advantage or preference for the winning alternative. In particular, we concentrate on the common case of scored zero-sum pairwise outcomes, where the score or amount won by an alternative from a comparison is equal to the amount lost by the other (for example, 2-player Texas Hold'em poker or any other even-odds gambling game). In this scenario, the $X_{i,j}$'s are continuous, unbounded, and paired *skew-symmetrically* such that $X_{i,j} = -X_{j,i}$ (zero-sum condition).

The general version of POCBAM proposed in Chapter 3 treats the results of pairwise comparisons of each distinct pair as independent. It produces a top- k

ranking of alternatives by Borda score, which is equivalent to ranking by expected pairwise comparison outcome against a randomly chosen opponent. The Borda score ranking is general in that it doesn't assume stochastic transitivity conditions on pairwise comparison means. However, in many real-world pairwise ranking cases, (stochastic) transitivity assumptions are reasonable, and it may be beneficial to adapt the fitness estimate to take advantage of the additional information gain from exploiting transitivity.

With this aim in mind, we propose modifying POCBAm by adding an additional assumption of a latent variable model for the pairwise outcome distributions. In particular, for the pairwise top- k selection problem described in Chapter 3, with continuous, skew-symmetric and unbounded pairwise comparison outcomes, we assume a Thurstonian style [93] model in which the fitness of each alternative a_i is wholly determined by the value of some underlying "quality" parameter γ_i . We model each pairwise comparison distribution $P(X_{i,j})$ with a Gaussian, with the mean specified by the difference $(\gamma_i - \gamma_j)$ of the quality parameters of the alternatives being compared, and variance $\sigma_{i,j}^2$. Similar Thurstonian models have been applied to preference-based Dueling Bandits, for example in [2] or [86]. It is important to note that the assumption of this model imposes a rigid stochastic transitivity relationship on alternative pairwise means, specifically that:

$$\mu_{i,j} > 0 \text{ and } \mu_{j,k} > 0 \implies \mu_{i,k} = \mu_{i,j} + \mu_{j,k} > 0$$

This assumption is stronger than the typical (strong) stochastic transitivity assumption ($\mu_{i,j} > 0$ and $\mu_{j,k} > 0 \implies \mu_{i,k} \geq \max\{\mu_{i,j}, \mu_{j,k}\}$). However, the rigidity of the equality allows us to express the likelihood of observed sample data explicitly under the assumed model, which we utilize below. We hope to demonstrate in Section 5.4, that this can be of practical benefit, even when the rigid transitivity assumption is broken to a moderate degree.

Given a set of sampling results $\{r_{i,j}^{(t)}\}_{t=1}^T$ of alternative pairs (a_i, a_j) , the likelihood of a set of model parameter estimates $\Gamma = (\tilde{\gamma}_1, \dots, \tilde{\gamma}_K)$ (underlying quality values) and $\Sigma = (\tilde{\sigma}_{i,j}^2)$ (pairwise variances) is given by:

$$L(\Gamma, \Sigma) = \prod_{t=1}^T \frac{1}{\tilde{\sigma}_{i,j}^{(t)} \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{-r_{i,j}^{(t)} - \tilde{\gamma}_i^{(t)} + \tilde{\gamma}_j^{(t)}}{\tilde{\sigma}_{i,j}^{(t)}} \right)^2} \quad (5.1)$$

$$ll(\Gamma, \Sigma) = -\frac{T}{2} \log(2\pi) - \sum_{t=1}^T \log(\tilde{\sigma}_{i,j}^{(t)}) - \frac{1}{2} \sum_{t=1}^T \left(\frac{r_{i,j}^{(t)} - \tilde{\gamma}_i^{(t)} + \tilde{\gamma}_j^{(t)}}{\tilde{\sigma}_{i,j}^{(t)}} \right)^2 \quad (5.2)$$

Taking partial derivatives w.r.t the parameters:

$$\begin{aligned} \frac{\partial ll(\Gamma, \Sigma)}{\partial \gamma_k} &= - \sum_{t=1}^T \{i^{(t)} = k\}_{\mathbf{I}} \frac{-r_{i,j}^{(t)} + \tilde{\gamma}_i^{(t)} - \tilde{\gamma}_j^{(t)}}{\tilde{\sigma}_{i,j}^{(t)2}} - \sum_{t=1}^T \{j^{(t)} = k\}_{\mathbf{I}} \frac{r_{i,j}^{(t)} - \tilde{\gamma}_i^{(t)} + \tilde{\gamma}_j^{(t)}}{\tilde{\sigma}_{i,j}^{(t)2}} \\ \frac{\partial ll(\Gamma, \Sigma)}{\partial \tilde{\sigma}_{k,l}} &= - \sum_{t=1}^T \frac{\{i^{(t)}, j^{(t)} = k, l\}_{\mathbf{I}}}{\tilde{\sigma}_{i,j}^{(t)}} + \sum_{t=1}^T \{i^{(t)}, j^{(t)} = k, l\}_{\mathbf{I}} \frac{\left(r_{i,j}^{(t)} - \tilde{\gamma}_i^{(t)} + \tilde{\gamma}_j^{(t)} \right)^2}{\tilde{\sigma}_{i,j}^{(t)3}} \end{aligned} \quad (5.3)$$

Where $\{i^{(t)} = k\}_{\mathbf{I}}$ is the 0/1 indicator function returning 1 when the first alternative in the pair sampled in the t^{th} sample is a_k , $\{j^{(t)} = k\}_{\mathbf{I}}$ the analogous function for the second alternative, and $\{i^{(t)}, j^{(t)} = k, l\}_{\mathbf{I}}$ the indicator function that returns 1 when the pair of alternatives in the t^{th} sample were a_k and a_l in either order. Calculating these partial derivatives, we can then maximize the (log) likelihood of the model parameters given our collected sampling data using a quasi-newton method.

We initialize the model parameter vector Γ with the alternative Borda Score estimates, centered (arbitrarily) about $\tilde{\gamma}_0$ and divided by population size, i.e:

$$\tilde{\gamma}_i = \frac{1}{K} \left(\sum_{j \neq i} \tilde{\mu}_{i,j} - \sum_{j \neq 0} \tilde{\mu}_{0,j} \right)$$

and the pairwise variance parameters Σ with the sample standard deviations:

$$\tilde{\sigma}_{p,q}^2 = \hat{\sigma}_{p,q}^2 = \frac{1}{n_{p,q} - 1} \sum_{t=1}^T (r^{(t)} - \tilde{\mu}_{p,q})^2 \mathbf{1}_{\{i^{(t)}, j^{(t)} = p, q\}}$$

These values represent a reasonable initial guess at the model parameters without taking account of the inter-dependence of pairwise sampling results. We then perform gradient based optimization of the parameter log likelihood (Equation 5.3) to converge to a local maximum, providing a better set of parameter estimates from the whole data, whilst exploiting the structure of the proposed underlying preference model. To ensure that the starting parameter estimates are reasonable when first beginning the sampling process, we allocate a portion of the sampling budget uniformly across all alternative pairs, taking n_0 samples of each.

After this initial “warm-up” sampling phase, we use this sampling information collected to inform our active sample selection procedure. Just as with standard POCBAm, we sequentially allocate additional samples to myopically maximize the expected increase in our confidence that the data we have collected allows us to correctly identify the top- k subset of alternatives. To quantify our level of confidence in our current top- k subset, we again estimate the Probability of Correct Selection (PCS), this time using alternative score distributions obtained from our fitted model parameters rather than our empirical estimates.

Under the assumption of normality of pairwise sample mean estimates, we can construct approximated posterior distribution estimates for the alternative Borda scores using the fitted model parameter estimates $\Gamma^* = [\gamma_1^*, \dots, \gamma_K^*]$, $\Sigma^* = (\sigma_{i,j}^{*2})$ obtained from the optimization step above:

$$\hat{S}_i \sim \mathcal{N}[\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i^2] = \mathcal{N}\left[\sum_{j \neq i} (\gamma_i^* - \gamma_j^*), \sum_{j \neq i} \frac{\sigma_{i,j}^{*2}}{n_{i,j}}\right]$$

As in Chapter 3, we simplify the calculation of PCS with two approximations. Firstly, we use a threshold value to separate the score distributions of the current

top- k alternatives from the rest of the population. For a constant c ,

$$PCS \geq \mathbb{P} \left[\left(\bigcap_{p \in \mathcal{I}} \{\hat{S}_p > c\} \right) \cap \left(\bigcap_{q \notin \mathcal{I}} \{\hat{S}_q < c\} \right) \right] \equiv APCS$$

Ideally we want to select c carefully to make this lower bound as accurate as possible. [25] show that APCS is asymptotically maximized for

$$c = \frac{\hat{\sigma}_{k+1}\hat{\mu}_k + \hat{\sigma}_k\hat{\mu}_{k+1}}{\hat{\sigma}_k + \hat{\sigma}_{k+1}}$$

Where $\hat{\mu}_k$ and $\hat{\mu}_{k+1}$ are the means and $\hat{\sigma}_k$ and $\hat{\sigma}_{k+1}$ are the standard deviations of the current k^{th} and $(k+1)^{\text{th}}$ best alternatives respectively. Secondly, evaluating the intersections in Equation 5.2 directly is difficult as the alternative score distributions are not independent. Instead we use the following approximation:

$$APCS \approx \prod_{p \in \mathcal{I}} \mathbb{P}\{\hat{S}_p > c\} \prod_{q \notin \mathcal{I}} \mathbb{P}\{\hat{S}_q < c\}$$

The justification for this approximation is discussed in detail in Section 3.3.1 of Chapter 3 (Inequalities 3.3 and 3.4). Using these approximations reduces the complexity of calculating PCS from $\mathcal{O}(K^2)$ to $\mathcal{O}(K)$, which is useful as our active sample selection procedure calculates predicted post-sample AEPCS for each of the $(K^2 - K)/2$ possible alternative pairs.

To inform our sample selection, we predict the approximate alternative score distributions if we were to allocate an additional sample to a particular pair. The expected effect of this sample would be a reduction in the uncertainty of the estimate of the pairwise mean $\mu_{i,j}$, thereby increasing our APCS estimate. It is important to note that, unlike the standard version of POcBAM described in Chapter 3, the score distribution parameters obtained from likelihood maximization are no longer unbiased estimates, and increasing the confidence in the sample estimate of a single pairwise mean would also affect the likelihood of the entire set of fitted

model parameters. To avoid repeating the costly optimization step for every possible alternative pair before selecting each sample, we make a further approximation by restricting the effect to only the score distributions of the two alternatives in the sampling pair. The potential error introduced by this approximation should be mitigated in part by the fact that we only require relative APCS values and by the performance benefit from exploiting preference transitivity. In Section 5.3, we demonstrate that despite this simplification, ML-POCBAm is still able to improve performance on a range of empirical tests.

Under our earlier assumption of normality, for a sample allocated to the pair (a_p, a_q) , our expected score distributions are:

$$\hat{S}_i^{p,q} \sim \mathcal{N} \left[\sum_{j \neq i} (\gamma_i^* - \gamma_j^*), \sum_{j \neq i} \frac{\sigma_{i,j}^{*2}}{n_{i,j} + \{i, j = p, q\} \mathbf{I}} \right]$$

To choose which sample to take, our active sampling method calculates an expected value for *APCS* for each potentially sampled pair (a_i, a_j) , denoted by $AEPCS^{i,j}$, and samples wherever this quantity is maximized:

$$\begin{aligned} AEPCS^{i,j} &= \prod_{p \in \mathcal{I}} \mathbb{P}\{\hat{S}_p^{i,j} > c\} \prod_{q \notin \mathcal{I}} \mathbb{P}\{\hat{S}_q^{i,j} < c\} \\ &= \prod_{p \in \mathcal{I}} \left(1 - \Phi \left(\frac{c - \hat{\mu}_p^{i,j}}{\hat{\sigma}_p^{i,j}} \right) \right) \prod_{q \notin \mathcal{I}} \left(\Phi \left(\frac{c - \hat{\mu}_q^{i,j}}{\hat{\sigma}_q^{i,j}} \right) \right) \end{aligned} \quad (5.4)$$

Where Φ is the cumulative distribution function for the standard normal distribution and using c as defined in Equation 5.2. A step-by-step description of the method is shown in Table 5.1.

5.3 Empirical Testing

In this section we evaluate the performance of the *ML-POCBAm* method on several test scenarios generated with by varying the underlying pairwise preference model

Table 5.1: The Maximum Likelihood POCBA_m Procedure

INPUT:	Set of K alternatives $\{a_1, \dots, a_K\}$, Required selection size k , Sampling budget N .
INITIALIZE:	Perform n_0 samples of each pair of alternatives; $n_{p,q} = n_0$ for all p, q , Sample means $\tilde{\mu}_{p,q} = \frac{1}{n_{p,q}} \sum X_{p,q}$, and: Standard dev. $\tilde{\sigma}_{p,q} = \sqrt{\frac{1}{n_{p,q}-1} \sum (X_{p,q} - \tilde{\mu}_{p,q})^2}$, Index set \mathcal{I} of best k alternatives.
WHILE $\sum n_{i,j} < N$ DO:	
FOR ALL PAIRS (a_i, a_j) :	
UPDATE:	
Initial model parameter estimates:	
$\tilde{\gamma}_p = \frac{1}{K} \left(\sum_{q \neq p} \tilde{\mu}_{p,q} - \sum_{q \neq 0} \tilde{\mu}_{0,q} \right)$,	
$\tilde{\sigma}_{p,q}^2 = \tilde{\sigma}_{p,q}^2 = \frac{1}{n_{p,q}-1} \sum_{t=1}^T (r^{(t)} - \tilde{\mu}_{p,q})^2 \{i^{(t)}, j^{(t)} = p, q\} \mathbf{I}$,	
Fit parameter estimates:	
$(\Gamma^*, \Sigma^*) = (\gamma_0^*, \dots, \gamma_K^*), (\sigma_{i,j}^*) = \operatorname{argmax}[\ell(\Gamma, \Sigma)]$	
For all $p = 1, \dots, K$:	
$\hat{S}_i \sim \mathcal{N}[\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i^2] = \mathcal{N}\left[\sum_{j \neq i} (\gamma_i^* - \gamma_j^*), \sum_{j \neq i} \frac{\sigma_{i,j}^{*2}}{n_{i,j}}\right]$	
Alternative score means $\hat{\boldsymbol{\mu}}_p^{i,j} := \hat{\boldsymbol{\mu}}_p$,	
Alternative score variances $(\hat{\boldsymbol{\sigma}}_p^{i,j})^2 := \sum_{q, q \neq p} \frac{\sigma_{p,q}^{*2}}{n_{p,q} + \mathbb{1}\{p, q = i, j\}}$,	
Boundary value $c = \frac{\hat{\sigma}_{k+1} \hat{\boldsymbol{\mu}}_k + \hat{\sigma}_k \hat{\boldsymbol{\mu}}_{k+1}}{\hat{\sigma}_k + \hat{\sigma}_{k+1}}$,	
$AEPCS^{i,j} = \prod_{p \in \mathcal{I}} \left(1 - \Phi\left(\frac{c - \hat{\boldsymbol{\mu}}_p^{i,j}}{\hat{\sigma}_p^{i,j}}\right) \right) \prod_{q \notin \mathcal{I}} \left(\Phi\left(\frac{c - \hat{\boldsymbol{\mu}}_q^{i,j}}{\hat{\sigma}_q^{i,j}}\right) \right)$.	
END FOR	
SAMPLE:	Select pair (a_i, a_j) that maximizes $AEPCS^{i,j}$, Perform sample of (a_i, a_j) , $n_{i,j} \leftarrow n_{i,j} + 1$,
UPDATE:	$\tilde{\mu}_{i,j}, \tilde{\sigma}_{i,j}, \hat{S}_i, \hat{S}_j$,
UPDATE:	\mathcal{I} .
END WHILE	
RETURN	\mathcal{I}

and on a realistic poker player top- k selection task for identifying the best performing subset of No Limit Texas Hold'em playing agents. We compare the method performance against the *SELECT/TOP* method proposed in [73], standard POCBAm and against uniform sample allocation. Unlike the racing methods we compared against in earlier experiments, *SELECT/TOP* specifically utilizes its total ordering assumption to reduce sampling complexity, making it a suitable competitor for ML-POCBAm, whenever this total ordering assumption holds. We use correct selection Success Rate as our performance metric for each task, defined as in Equation 3.12. For the following experiments, unless otherwise stated, results are averaged over 10,000 independent replications and the initial random seeds used to generate the underlying alternative parameters in each replication are common to each method (i.e the true alternative Borda scores for the n^{th} replication of the ML-OCBAm method are the same as for the n^{th} replication of the *SELECT/TOP* method).

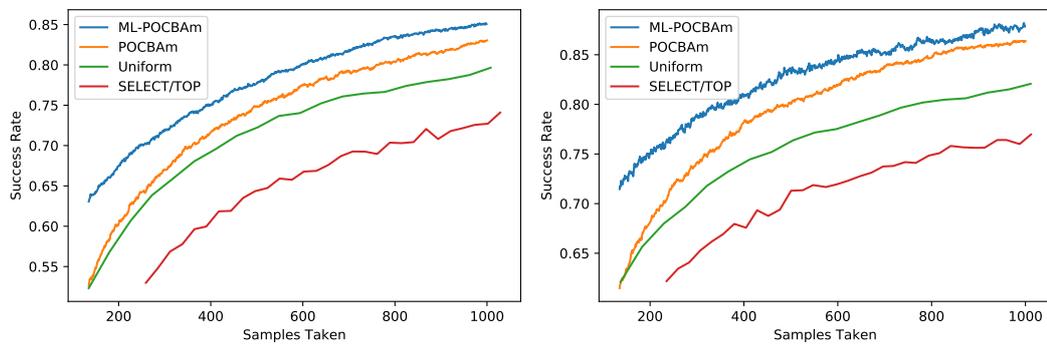
5.3.1 Value-based Thurstone model

The first test scenario we consider is where the pairwise preference distributions are generated according to the value-based Thurstonian model described in Section 5.2, i.e. the underlying model assumed by ML-POCBAm is correct. Note that in this case, the total ordering assumption of the *SELECT/TOP* method also holds. In each replication of the experiment, the true parameter vector Γ for our population of alternatives is generated with each entry γ_i chosen uniformly at random from $[0, 1]$. The pairwise variance parameters $\sigma_{i,j}^2$ are also selected independently and uniformly from $[0, 1]$ for each pair of alternatives. These parameter values determine the true top- k ranking, and the pairwise sampling distributions. At each step t of the experiment, each of the sampling methods may select a pair of alternatives (a_i, a_j) and receive a sample result drawn from the distribution $\mathcal{N}(\gamma_i - \gamma_j, \sigma_{i,j}^2)$. The ML-POCBAm and POCBAm methods were allowed a maximum of 1000 samples when selecting from 10 alternatives, and 30,000 when selecting from 100, with the first

3 samples for each pair allocated uniformly by each method to make the initial pairwise mean estimates. For the SELECT/TOP method, the number of samples taken by the method is variable, and is dependent on a pre-chosen parameter ν that specifies the number of repetitions of each comparison to make during the knockout tournaments and heap construction phases. As such, we vary ν to produce a range of different average sampling budgets, without limiting the maximum sampling budget used by this method. Figure 5.1 shows the method performance on both top 4 of 10, top 1 of 10 selection and larger scale top 40 of 100 selection. In all three cases, we see that ML-POCBAm achieves highest success rate throughout the 1000 samples, with regular POCBAm performing second best. The benefit of exploiting the transitivity of the model is clear, as fitting the most likely model parameters given only the initial warm-up sampling data significantly improves success rate. Interestingly, SELECT/TOP performs worse than uniform sampling in each of the cases, which is surprising given its strong performance in [73] and the required assumption of a total ordering holds. We discuss the cause for this poor performance in the next subsection.

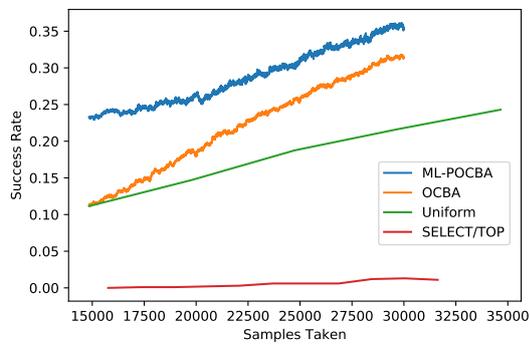
5.3.2 Analyzing SELECT/TOP

As [73] note, the probability of correct selection of the SELECT tournament stage is highly dependent on the difference in quality of the top alternative and the remainder. If any of the best alternatives are knocked out of a sub tournament it is much more difficult SELECT/TOP to return them to consideration than for ML-POCBAm to produce a correct ranking given a poor estimate of a single pairwise mean. It is therefore critical to the performance of SELECT/TOP that the best alternatives are highly unlikely to ever lose in a comparison to alternatives not in the correct top set. Taking the above top 1 of 10 experiment as an example, we generated the alternative score parameters (γ_i 's) uniformly from $[0, 1]$. This means



(a) 4 of 10

(b) 1 of 10



(c) 40 of 100

Figure 5.1: Performance of ML-POCBAm, POCBAm, SELECT/TOP and uniform sample allocation on top 4 of 10, top 1 of 10 and top 40 of 100 selection with a Thurstonian latent preference model.

that the expected comparison mean between the top and 2nd best alternative will be the difference between the 9th and 10th order statistic of 10 uniform samples ([41], page 63): $1/11 = 0.0909$. Using the expected value of the standard deviation parameter for this pair (0.5), the probability of the top alternative winning over n comparisons against the second best is given by:

$$1 - \Phi\left(\frac{n/11}{\sqrt{0.25n}}\right)$$

Which is only about 0.572 when $n = 1$ and 0.717 when $n = 10$. The probability of the top alternative losing in one of the several tournament rounds is therefore significant.

However, one notable advantage of SELECT/TOP over both ML-POCBAm and POCBAm is its lower sampling complexity. To generate initial estimates for alternative score distributions, both OCBA-based methods use a small number of warm-up samples of each pair, which, like uniform allocation, scales $O(K^2)$ with the number of alternatives. In contrast, SELECT/TOP has complexity $O(K \log(K))$. As the number of alternatives grows large, the cost of the warm-up phase of ML-POCBAm will become dominant and can restrict performance. Therefore, to ensure SELECT/TOP has a fair chance to display its merits, we compare it to ML-POCBAm and uniform sampling using the larger population size (100) used above, selecting the top-1 alternative, where the gap in underlying quality value between the best alternative and the rest is relatively large. To do this, we set $\gamma_0 = 1.2$ and sample $\gamma_{1:99}$ and pairwise variances $\sigma_{0:99,0:99}^2$ from $U[0, 1]$. Results are shown in Figure 5.2. We observe that the large gap between the best alternative and the remainder improves the success rates for all three methods, as the selection problem is now significantly easier. SELECT/TOP in particular performs much better here, reaching the same success rate as ML-OCBAm achieves immediately after its warm-up phase with approximately 12% fewer samples. However ML-POCBAm is

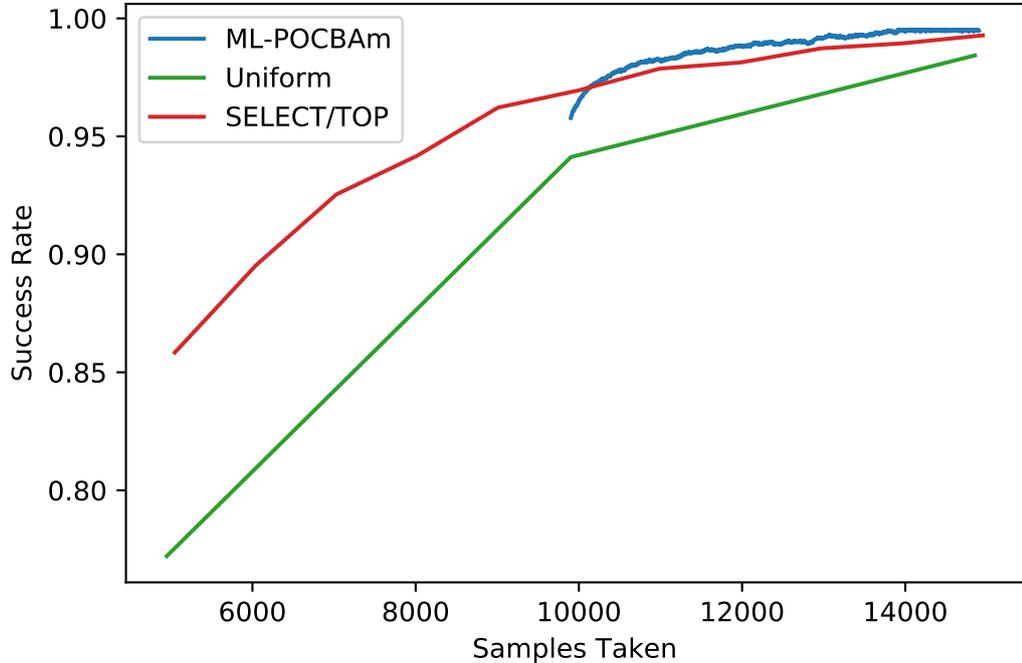


Figure 5.2: Performance of ML-POCBAm, uniform sampling and SELECT/TOP on top 1 of 100 selection with a Thurstonian latent preference model and a large separation between the quality value of the top alternative and the rest of the population. Here $\gamma_0 = 1.2$, with γ_i selected uniformly at random from $[0, 1]$ for all other alternatives.

still able to reach perfect accuracy earlier. After the initial warm-up phase ML-POCBAm is able to refit its model, recalculate AEPCS and thus allocate every sample to the myopically optimal pair. This leads to substantially more efficient sample acquisition than using fixed rules for the number of sample repetitions as in SELECT/TOP and uniform allocation.

5.4 Noise Perturbed Thurstone model

In the experiments above, the ground-truth mechanism for generating pairwise sample outcomes corresponds exactly with the model used by ML-POCBAm. Under

these conditions ML-POCBAM performs very well, but having sole access to the true model gives the method a potentially unrealistic advantage over its competitors. In a real-world scenario, the underlying model is generally unknown, and the rigid assumptions of the Thurstonian model are unlikely to hold exactly, even if they are broadly correct. In this section, we investigate the effect on algorithm performance of random perturbations to the pairwise means of the underlying Thurstonian model, and propose an adaptation to the method to make it more resilient to model inaccuracy. Here the model parameters are generated in the same way as before, but the means of the sampling distributions are each perturbed by an additive noise component, i.e, for each pair (a_i, a_j) samples are drawn from the distribution $\mathcal{N}(\gamma_i - \gamma_j + \epsilon_{i,j}, \sigma_{i,j}^2)$. The $\epsilon_{i,j}$'s are chosen independently at random from the distribution $\mathcal{N}(0, d^2)$. An example of the effect of these perturbations on the pairwise transitivity of the alternatives for $d = \{0.1, 0.2, 0.3, 0.4\}$ is shown in Figure 5.3. As d increases the number of alternative pairs that violate the total ordering assumption (any negative mean in the upper triangle of the ordered pairwise mean matrix) or the weak stochastic transitivity condition as defined in [35] (alternative triplets (a_i, a_j, a_k) where $\mu_{i,j} > 0, \mu_{j,k} > 0$, but $\mu_{i,k} < 0$) increases considerably. Figure 5.4 shows the performance of ML-POCBAM and the best performing competing method from the previous section (POCBAM) on similarly generated cases. We see that the performance of ML-POCBAM is significantly reduced as d increases, and the parametric model used by ML-OCBAM becomes a poorer representation of the population. In contrast, POCBAM is resilient to the perturbations, as each pairwise mean is treated independently. Figure 5.4c highlights the difference in performance between the two methods. For low d , fitting the Thurstonian model to the data is still beneficial to selection accuracy, although as sampling budget increases, this advantage lessens, as POCBAM will asymptotically converge to the correct subset. For high d values, POCBAM performs better than ML-OCBAM, with the performance gap widening as more samples are taken. This suggests that when the model

used by ML-POCBAM is sufficiently inaccurate, fitting this model to the sampling data harms not only prediction accuracy, but also sample acquisition.

Clearly it would be beneficial to use a sampling method that could exploit the benefit of fitting the parametric model when appropriate, but can retain the resilience to pairwise intransitivity of the POCBAM method. However, without knowing the true top- k subset, it is difficult to know what the effect of modeling inaccuracy is on correct selection. [84] discusses different methods for estimating the degree of intransitivity when ranking pairwise systems. They suggest comparing the symmetrized Kullback Leibler (KL) divergence between the observed (sample) distributions and the predicted pairwise distributions according to the model. Using this idea, we define the empirical Intransitivity Index (**II**) of our population of alternatives as:

$$\mathbf{II} = 1 - e^{-\frac{1}{2K} \sum_i (\mathbb{D}_{KL}(\hat{S}_i || S_i^*) + \mathbb{D}_{KL}(S_i^* || \hat{S}_i))} \quad (5.5)$$

Where $\mathbb{D}_{KL}(P||Q)$ is the KL divergence of the distributions of continuous random variables P and Q , defined as:

$$\mathbb{D}_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \quad (5.6)$$

For Gaussian distributions $P \sim \mathcal{N}[\mu_1, \sigma_1^2]$ and $Q \sim \mathcal{N}[\mu_2, \sigma_2^2]$, Equation 5.6 becomes [79]:

$$\mathbb{D}_{KL}(P||Q) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

Using this, we can easily calculate an empirical estimate of the degree of pairwise intransitivity between alternatives. Note that **II** = 0 if the predicted and observed distributions are identical, i.e. the observed sampling data corresponds

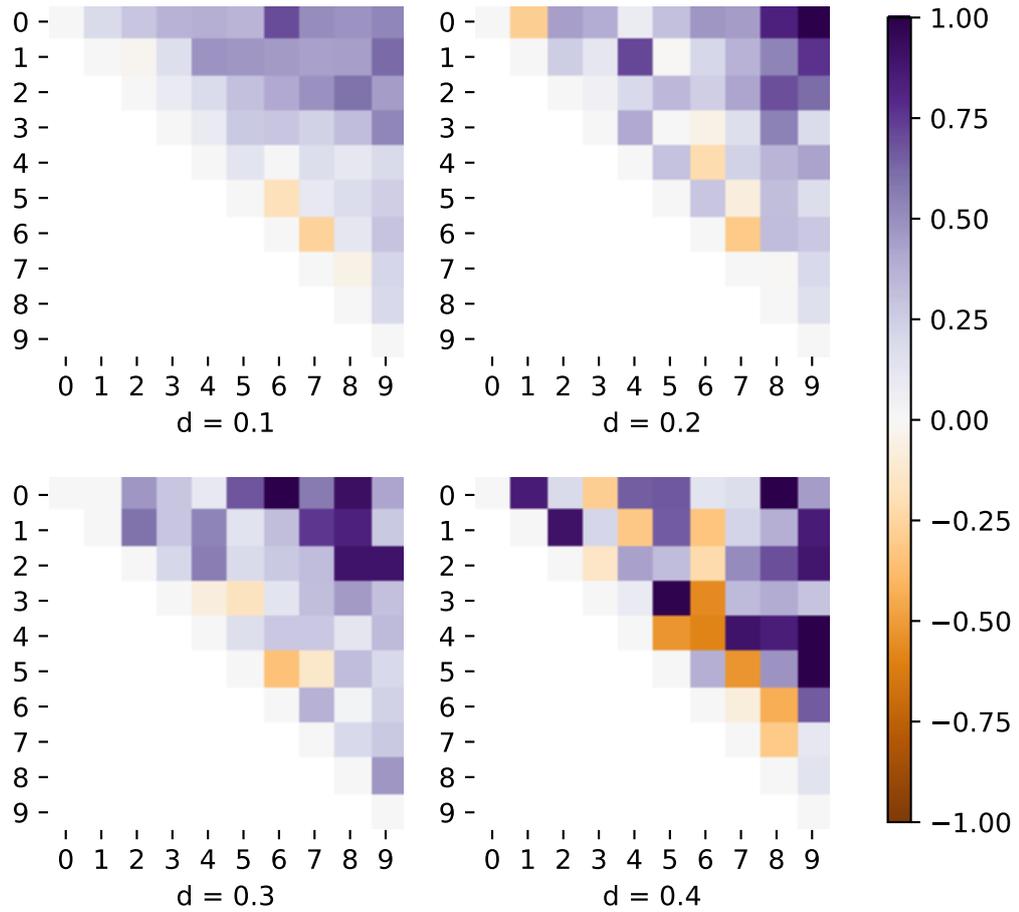
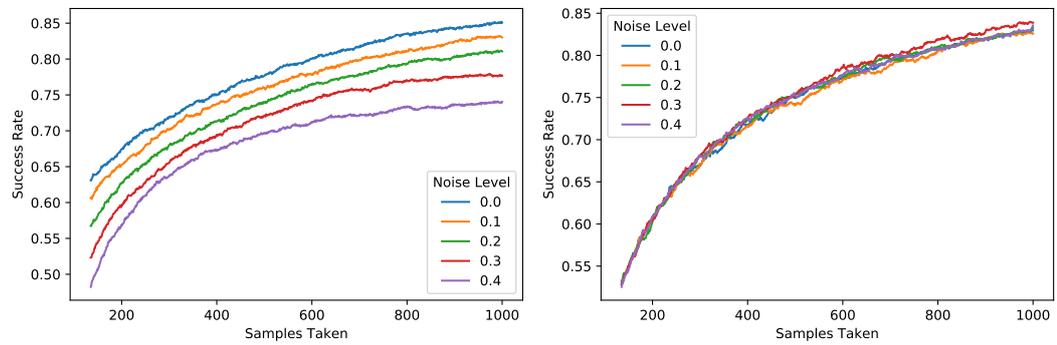
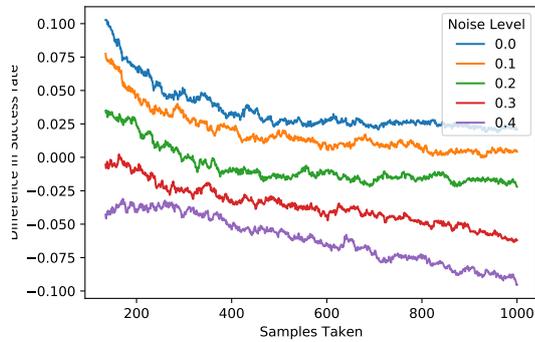


Figure 5.3: Sample pairwise preference matrices for populations of 10 alternatives generated according to the noise perturbed Thurstonian model described in 5.4 with different values of d . Alternatives are indexed by Borda score. We can clearly see this increasing degree of pairwise intransitivity as d increases. For an example of an intransitive triplet, consider (a_0, a_1, a_2) for $d = 0.4$, we have $\mu_{0,1} > 0, \mu_{1,3} > 0, \mu_{0,3} < 0$.



(a) ML-POCBAm

(b) POCBAm



(c) Performance difference

Figure 5.4: Performance of ML-POCBAm and POCBAm on top 4 of 10 selection problems with noise disturbed Thurstone preference with various different degrees of noise.

precisely with the fully transitive Thurstonian model, and $\mathbf{II} = 1$ if the KL divergence between the predicted and observed distributions is infinite. Given a suitable threshold value for \mathbf{II} , we can define a hybrid sampling method that chooses samples according to ML-POCBAM when the currently available sampling data suggests the Thurstonian model fitted by ML-POCBAM to be plausible, and reverts to standard POCBAM for sample acquisition if the divergence between the fitted model and the observed data is too great.

Figure 5.5 shows the relationship between d , \mathbf{II} , and the difference in performance between ML-POCBAM and POCBAM after 250 samples on the noise perturbed Thurstonian model, averaged over 1000 replications. Performance for the two methods is equal at approximately $\mathbf{II} = 0.17$. In the next section we test the hybrid ML-POCBAM using this threshold value on a game player ranking problem.

5.5 Top- k Selection of Poker Players with a Transitive Fitness Model

In this experiment, we return to the No Limit Texas Hold'em (NLTH) player selection problem we considered in Chapter 4 to investigate whether the hybrid ML-POCBAM method can improve the selection accuracy of the top- k players in each generation. We generate 10 populations of 10 candidate players, by randomly selecting weights for the OFRE and SWRE networks uniformly at random from $[-1, 1]$. We generate the ground truth rankings for each population of players by playing a large number (20,000) of poker hands between each possible pair of players. Game rules are as in the AAAI Annual Computer Poker competition (<http://www.computerpokercompetition.org>). Player starting stacks are set to 200 Big Blinds (BB) and reset to the starting amount after each hand. The 20,000 hands between each pair of players is split into two parts, with each player playing 10,000 hands from each position (playing first or second), with duplicate sets of card

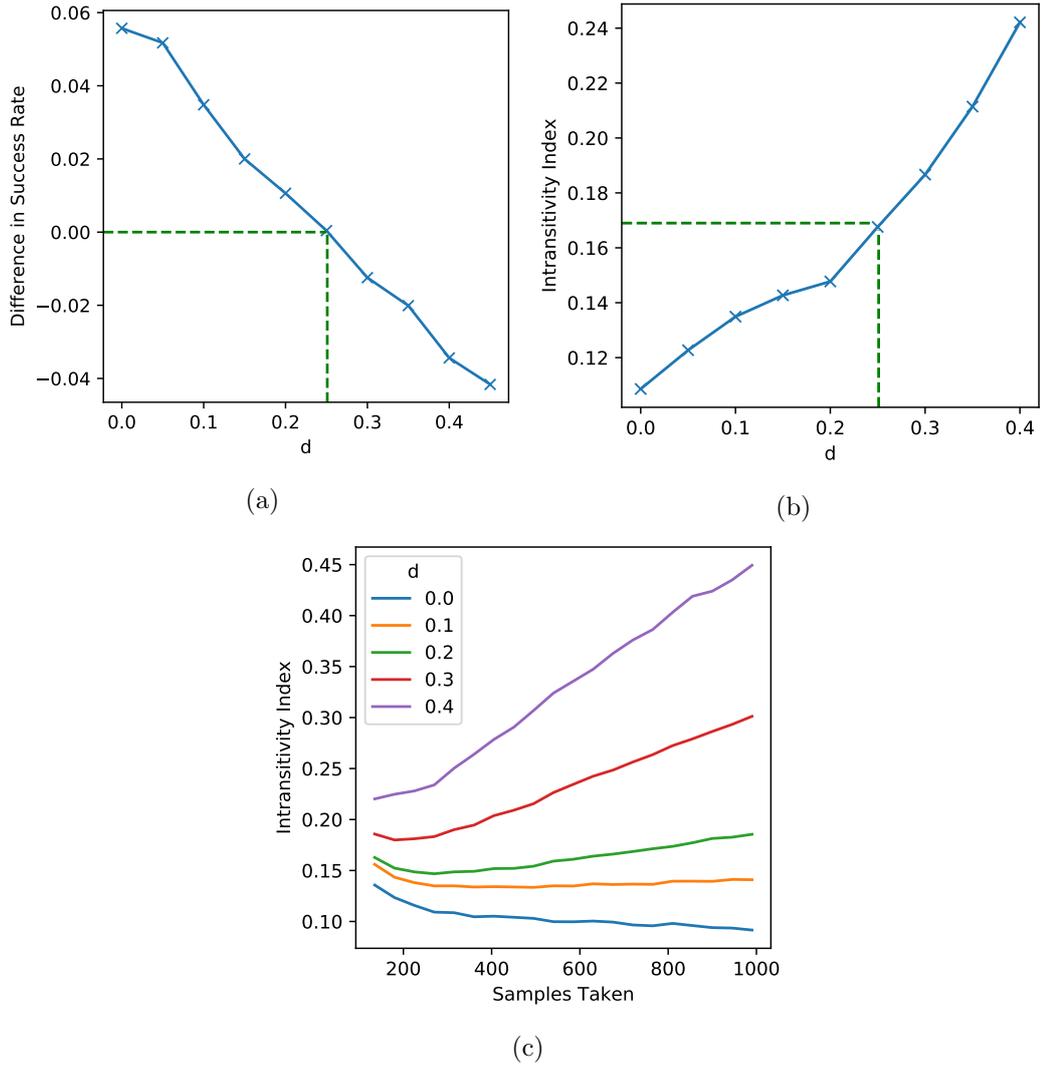


Figure 5.5: Subfigure 5.5a shows relative performance of ML-POCBAM and POCBAM after 250 samples for different values of d , with estimated intransitivity index values for each different d shown in 5.5b. 5.5c shows how the empirically estimated intransitivity index changes as more samples are taken for different d values.

shuffles used in each part. Players are ranked by their cumulative win/loss amount.

Figure 5.6 shows the pairwise performance matrices for each set of players according to these ground truth rankings. We observe that none of the pairwise rankings of the generated player groups is stochastically transitive, with negative pairwise means appearing in the upper triangle of the ordered preference matrices in all ten cases, with an average of 11.1 of 45 per population. However, pairwise performance of players frequently appears to be correlated, as many players who have similar (or different) performance against a given opponent also have similar (or different) performance against their other opponents, as evidenced by the clear vertical and horizontal stripes visible in the figure. As the network weights in the players were randomly generated and untrained, it was relatively common for one of the possible player actions to be dominant. The effect of this was to greatly increase the range of pairwise sampling variances. Compare for example games between two players who almost always fold immediately with games between two players who immediately raise all-in. Both pairs will have a long-term pairwise mean of 0, but the sampling variance of the latter pair will be 40,000 times greater. These large differences in variance scales, combined with the presence of intransitivity, make this a particularly challenging ranking problem.

Figure 5.7 compares the performance of the hybrid ML-POCBAm discussed above, with \mathbf{II} threshold 0.17 against standard POCBAm, SELECT/TOP and uniform sampling. Each subplot shows one of the randomly generated populations of players shown before in Figure 5.6. In each case, the performance of each method is averaged over 500 repetitions, with different (but common across methods) random seeds for deck shuffles in each repetition. Hybrid ML-POCBAm performs best, with similar or better performance to standard POCBAm on all 10 cases. This is perhaps unsurprising, as the hybrid method can revert to using standard POCBAm for sample acquisition wherever sample results significantly disagree with its assumed

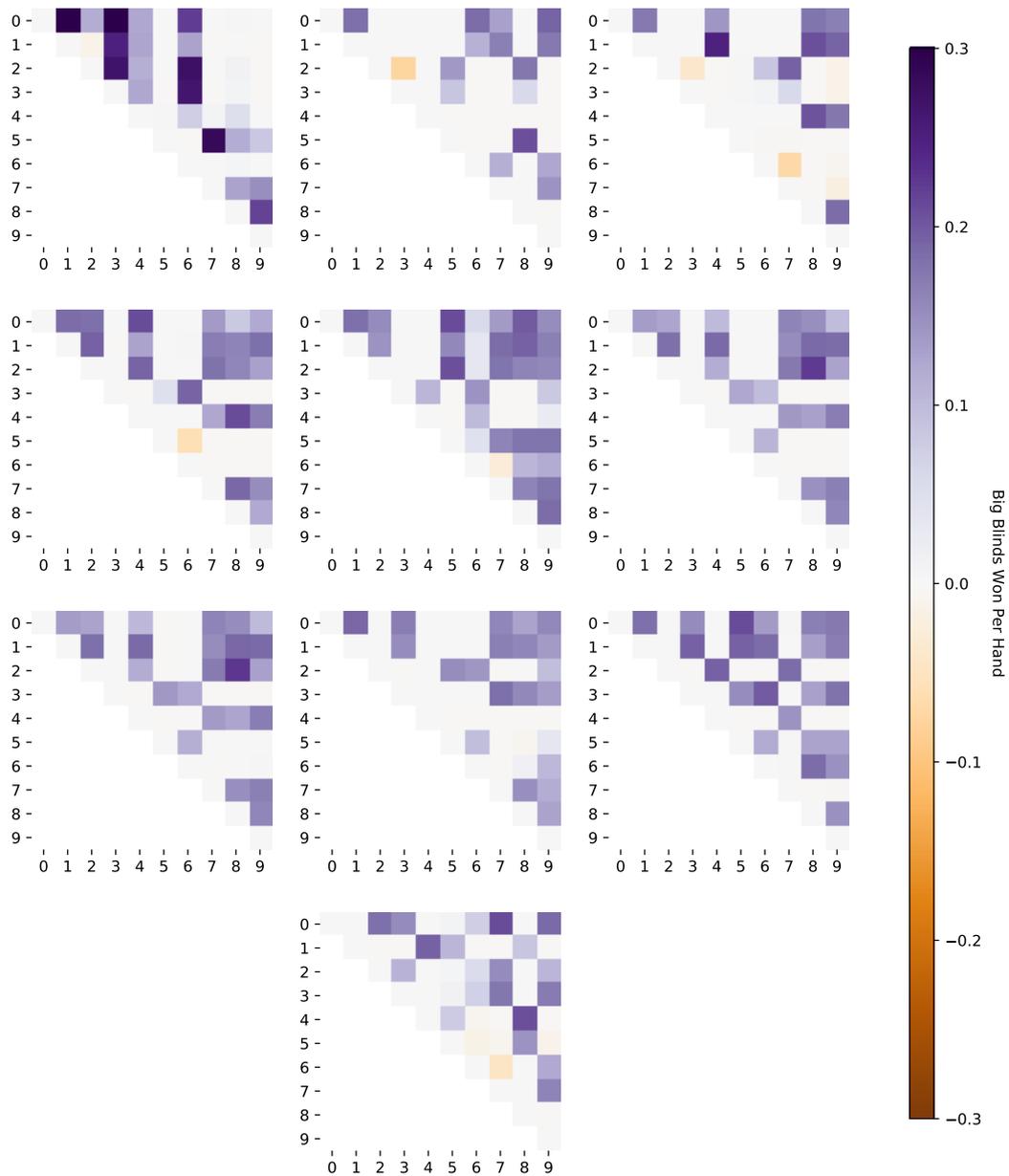


Figure 5.6: Visualization of the pairwise preference matrices of the ten randomly generated populations of poker players used in Section 5.5. Players are indexed in true ranking order.

Thurstonian model. SELECT/TOP is again the worst performing method, presumably because its total ordering assumption fails to hold given the high number of pairwise intransitivities in the test populations.

5.6 Summary

In this chapter, we have adapted the Thurstonian parametric model for Dueling Bandit problems with quantitative sample outcomes and proposed two sample acquisition methods that exploit this model to improve top- k selection accuracy. Both ML-POCBAM and hybrid ML-POCBAM extend and improve upon the previously published POCBAM method by selectively exploiting a parametric pairwise preference model, and significantly outperform both SELECT/TOP and uniform sampling. We suggest that this result may be useful, for example in evolutionary reinforcement learning, where each generation of the evolutionary strategy uses the top- k players to generate the generating distribution for the next generation. Improving the quality of the top- k players without increasing simulation costs can potentially lead to the learning of better final weights for the player networks in fewer generations. We intend to test this application in future work.

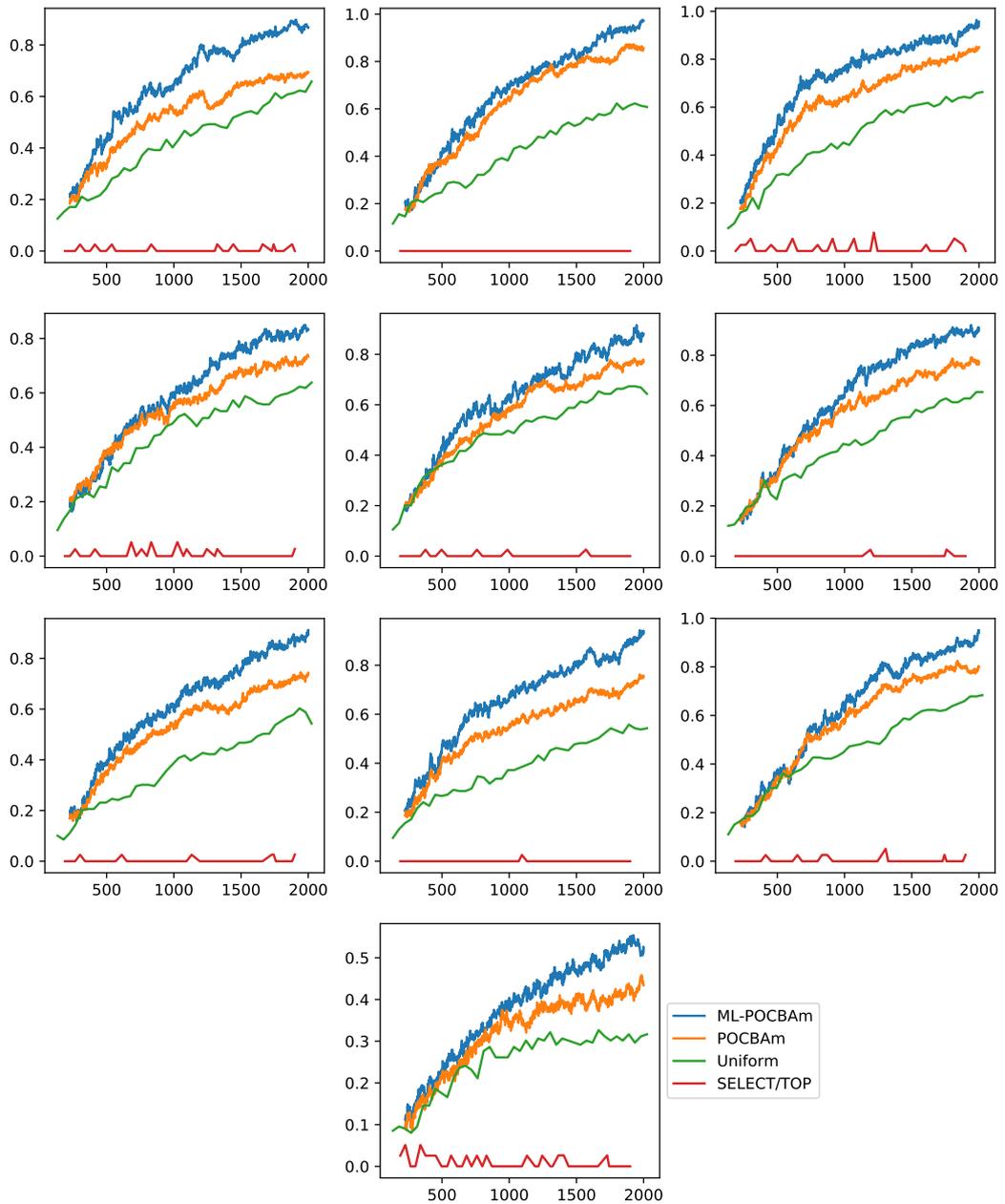


Figure 5.7: Performance of ML-POCBAm, POCBAm and uniform sampling on top 4 of 10 selection for each of the populations of poker players shown in Figure 5.6.

CHAPTER 6

Conclusions

In this thesis, we have considered the problem of the top- k selection of alternatives, where the fitness of alternatives must be estimated from the results of noisy pairwise comparisons. This is an interesting and important problem, with applications to the ranking of game players, preference elicitation for decision makers and in personalization of online advertising.

The first key contribution of the thesis is the adaptation of two well known active learning frameworks for Ranking and Selection to the pairwise setting. In Chapter 3, we described the Pairwise Optimal Computing Budget Allocation method for subset selection (POCBAm), based on the methods of [29] and [25], and the Pairwise Knowledge Gradient (PKG) method, based on the method of [40] and the Value of Information maximization methods of [32]. In contrast to other pairwise sampling procedures for top- k selection, including racing methods like the H-Race [19] and AR method [52], or the tournament-like SELECT/TOP method [73], both POCBAm and PKG select samples in myopically optimal ways (under particular assumptions). They do this by choosing each sample to optimize the value of a particular acquisition function, either to maximize the expected additional confidence gained from a sample (POCBAm), or to maximize the probability of gaining valuable information from taking a single additional sample (PKG). We proved that POCBAm is asymptotically guaranteed to converge to the correct top- k set, and that PKG will

converge correctly with unbounded sampling outcomes, but can fail otherwise, for reasons particular to the pairwise case. In our empirical experiments in Chapter 3, we saw that the efficient myopic sampling can lead to significant improvement in top- k selection accuracy in a variety of settings compared to alternative methods, and that the myopic optimization can give particularly large improvements with highly constrained sampling budgets. POCBAm in particular performed very well, and was the best performing method on all the tests with preference-based (binary) samples, and was competitive with PKG on normally distributed sample outcomes.

Based on this strong performance, we then investigated a potential application case for POCBAm in Chapter 4. The method can be easily integrated into a range of evolutionary optimization methods to help improve individual selection when noise is present. In some applications, for example the evolution of agents for 2-player games, the fitness evaluation stage of these evolutionary optimizers requires pairwise sampling. For such cases, our POCBAm method may provide a useful tool to quickly learn the most relevant fitness information for alternatives in the evolutionary population, whilst requiring fewer samples to reaching a higher accuracy than other methods. Using the popular CMA-ES optimizer as an example, we showed on a range of experiments on different common test functions that the improvements in selection accuracy of POCBAm over uniform sample allocation can help to mitigate the effect of sampling noise. Comparing the performance of CMA-ES after 50 generations with uniform sample selection, POCBAm was able to achieve the same error with between 22% and 84% fewer samples. This result builds upon our earlier work in Appendix B, where we showed how Knowledge Gradient based sample selection methods could improve CMA-ES performance in non-pairwise cases. We also tried evolving artificial neural network poker players, based on a successful player model from [64], using CMA-ES to optimize the network weights. We evolved two different sets of players, one using POCBAm for fitness evaluation each generation, and one using uniform sampling. We measured the

performance of the players by testing the player populations from each generation against the final evolved players from the POCBAM set. It seems that POCBAM sampling was beneficial – the POCBAM evolved players performed significantly better than the uniform evolved ones on this test. However, the overall strength of the evolved players was low as neither set of players were able to beat a set of fixed benchmarks.

Integrating our sampling methods into CMA-ES is straightforward, but using the standard form of POCBAM, based on maximizing the $\Delta AEPCS$ acquisition function, may not lead to ideal performance. In Section 4.6, we discussed alternative forms of the POCBAM acquisition function, formulated to maximize alternative criteria such as the expected reduction in opportunity cost, or to minimize the divergence between the ideal and selected CMA-ES recombination distribution. This allows the sampler to consider the actual effect of possible selection errors, aiming to minimize the cost of mistakes rather than the probability they occur. One of the strengths of the OCBA framework upon which POCBAM is based is its flexibility, and incorporating these alternative acquisition functions only requires a single line of the algorithm to be changed. However, we have so far not tested these methods empirically, and it would be interesting to see whether they can indeed improve performance of the evolutionary optimizer.

In Chapter 5, we considered how a parametric model could be incorporated into POCBAM to improve pairwise top- k selection performance where a high degree of transitivity is present. Focusing on a specific case – normally distributed pairwise sample outcomes and a Thurstonian style underlying model, we proposed an adapted version of POCBAM that fitted the model parameters to maximize the likelihood of the observed sampling data, and used this fitted model to construct improved estimates of alternative score distributions. We showed that the ML-POCBAM method improves selection accuracy when the model provides a good representation of the ground truth, but can harm selection compared to standard POCBAM when

the model is inaccurate. To counteract this, we defined an Intransitivity Index metric, which tries to quantify in some way how plausible the fitted model is based on empirical data. This can be used in a hybrid sampling method that uses ML-POCBAM to select samples when the Intransitivity Index suggests the parametric model is reasonable, but can fall back to standard POCBAM if necessary. To test this hybrid method, we returned to the poker player selection example of Chapter 4. We created several sets of players with different degrees of transitivity in their pairwise performance. Interestingly, even for player sets with high degrees of intransitivity, the Hybrid ML-POCBAM performed as well or better than regular POCBAM.

6.1 Future Work

The thesis suggests several future directions for research and highlights some areas where further investigation may be needed. Firstly, a possible future direction would be to try and adapt our proposed sampling methods for full ranking, rather than just subset selection. This may require relatively little effort – for example in the case of POCBAM, instead of approximating PCS by calculating the mass of our empirical score distributions that lie on the correct side of a threshold between the top subset and the remaining alternatives, we could simply approximate the *Probability of Correct Ranking* by using multiple thresholds between the peaks of each score distribution, looking at the mass of the score distributions that lies within the correct threshold bounds. Following the same schema as the top-k subset case, we can use thresholds:

$$c_p = \frac{\hat{\sigma}_{p+1}\hat{\mu}_p + \hat{\sigma}_p\hat{\mu}_{p+1}}{\hat{\sigma}_p + \hat{\sigma}_{p+1}} \text{ for } p = 1, \dots, K - 1$$

Thus, we have the following expression for each pair:

$$AEPCR^{i,j} \equiv \mathbb{P}\{\tilde{\mathcal{S}}_1^{i,j} < c_1\} \mathbb{P}\{\tilde{\mathcal{S}}_K^{i,j} > c_{K-1}\} \prod_{p=2}^{K-2} \mathbb{P}\{c_p < \tilde{\mathcal{S}}_p^{i,j} < c_{p+1}\} \quad (6.1)$$

This gives us an Approximation of the Expected Probability of Correct Ranking (AEPCR), given we take a sample of pair (a_i, a_j) . As with standard POCBAm, we can use this as a sample acquisition function. There are a range of active sampling algorithms for ranking in the existing literature, and it would be interesting to compare how well a full ranking form of POCBAm or PKG could perform against them.

In Chapter 4, we tested POCBAm as a fitness evaluation method for CMA-ES on the task of evolving No Limit Texas Hold'em players. Our approach was co-evolutionary, with the fitness of alternatives in the CMA-ES population defined as their Borda score against their peers. One of the key difficulties in competitive co-evolution is the problem of forgetting over time, whereby populations lose the knowledge of how to beat strong strategies identified in past generations because their current fitness evaluation is based only on performance against the current generation. This leads to cyclical performance, preventing the ES from converging to performant solutions [58]. There are several possible ways to try and mitigate this issue, for example using a *Hall of Fame* (HOF) [82]. Exceptionally strong alternatives from across all previous generations can be added to the HOF, for example if they beat all current HOF alternatives, and have the highest fitness in their current generation. Testing against the HOF augments the fitness evaluation stage of the ES by ensuring that the selected alternatives retain the ability to perform well against past strong strategies. Testing all of the alternatives in the evolutionary population against the entire HOF, especially if the HOF grows over time incurs considerable computational cost. It would be interesting to try and adapt the POCBAm framework to efficiently select the most informative pairwise samples from two disjoint sets

– the population and the HOF. This could be useful in improving the performance of evolutionary strategies in co-evolving game players. The alternative acquisition functions for POCBAM based on OppC and KL divergence that we proposed in Section 4.6 also require testing.

The work on exploiting transitivity in pairwise preferences considered only one possible top- k ranking case (normally distributed sample outcomes) and only one parametric model. Other common models, for example the BTL model for binary comparisons could be considered. The ML-POCBAM method we proposed worked well in our empirical tests, but only when the parametric model provides a good approximation of the ground truth. To mitigate this weakness, we proposed a hybrid method that can revert to the more general standard POCBAM when needed. The hybrid method requires an additional parameter to be set by the user – the intransitivity index (**II**) (Definition 5.5) threshold at which the sampler returns to model free POCBAM. Our **II** definition is based on the version proposed in [84], but neither that work, nor this thesis have thoroughly investigated the sensitivity of **II** to the parameters of the top- k problem like population size, top- k subset size, number of samples etc. Knowing a suitable setting for this parameter may be important to ensure good performance in other problem configurations. There may also be other possible approaches for exploiting problem structure. ML-POCBAM seeks to improve the accuracy of the estimated score distributions by using the entirety of the sampling data to fit each of them, based on a parametric model. However, just like standard POCBAM, this still requires $\mathcal{O}(K^2)$ samples. An alternative way to benefit from transitive structure, would be to try and reduce sampling complexity, to make the method more scalable for large populations of alternatives. For example, the knockout structure used by SELECT/TOP implicitly infers the direction of many of the pairwise preferences, reducing sampling complexity to $\mathcal{O}(K \log(K))$. However, the current formulation POCBAM uses estimates of every pairwise mean to construct its score distributions. Formulating a version that avoids the requirement

could be an interesting extension of this work. Finally, we showed that the hybrid ML-POCBAM method could improve the selection of high performing NLTH players over the standard POCBAM method we used in Chapter 4. We have not yet tested what effect this might have on the quality of the players we can evolve using CMA-ES.

APPENDIX A

Proofs From Chapter 3

Here we give a proof that ranking alternatives by Borda score from pairwise comparisons will correctly reconstruct an underlying latent ranking given certain conditions. For simplicity the proof is given for only binary pairwise sample outcomes, but the proof for unbounded, value-based pairwise samples requires only minor modification.

Theorem A.0.1. *Given a set of alternatives \mathcal{A} and some underlying total ordering \succeq on \mathcal{A} , suppose that:*

- *For each pair of alternatives (a_i, a_j) there is associated a Bernoulli random variable $X_{i,j}$ and that these random variables are paired such that $X_{i,j} = 1 - X_{j,i}$.*
- *There exists a function $F : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ defined as:*

$$F(a_i, a_j) = \mathbb{E}[X_{i,j}]$$

with the following properties:

1. **Strong Stochastic Transitivity (SST)** [86] *with respect to \succeq : For alternatives a_i, a_j, a_k :*

$$a_i \succeq a_j \succeq a_k \implies F(a_i, a_k) \geq \max\{F(a_i, a_j), F(a_j, a_k)\}$$

2. **Pairwise Distinguishability:** For any two distinct alternatives a_i and a_j , there exists some alternative a_k such that:

$$F(a_i, a_k) \neq F(a_j, a_k)$$

We define the Borda Score ordering \geq_B on \mathcal{A} as follows:

$$a_i \geq_B a_j \iff B(a_i) = \sum_{a_k \neq a_i} F(a_i, a_k) \geq \sum_{a_k \neq a_j} F(a_j, a_k) = B(a_j)$$

Then \succeq and \geq_B are equivalent, i.e.:

$$a_i \succeq a_j \iff a_i \geq_B a_j$$

Proof. We begin by showing that:

$$a_i \succeq a_j \implies a_i \geq_B a_j \tag{A.1}$$

To do this, we aim to show that the SST condition implies that all terms in the Borda Score $B(a_i)$ of a_i will be at least as large as the corresponding terms in $B(a_j)$.

First note that for any alternative a_i , $F(a_i, a_i) = 1 - F(a_i, a_i) = \frac{1}{2}$. Hence, by the SST condition, for any two alternatives a_i and a_j :

$$a_i \succeq a_j \implies F(a_i, a_j) \geq \max\{F(a_i, a_i), F(a_i, a_j)\} \geq \frac{1}{2} \tag{A.2}$$

and so

$$F(a_j, a_i) \leq \frac{1}{2}$$

Now consider the terms in the sums $B(a_i) = \sum_{a_k \neq a_i} F(a_i, a_k)$ and $B(a_j) = \sum_{a_k \neq a_j}$.

For each alternative a_k , we have three possible cases:

1. $a_i \succeq a_j \succeq a_k$:

This case is simple. By the SST condition:

$$F(a_i, a_k) \geq \max\{F(a_i, a_j), F(a_j, a_k)\} \geq F(a_j, a_k)$$

2. $a_i \succeq a_k \succeq a_j$: From Equation (A.2) we have that:

$$F(a_i, a_k) \geq \frac{1}{2} \geq F(a_j, a_k)$$

3. $a_k \succeq a_i \succeq a_j$: From the SST condition and Equation (A.2), we have:

$$F(a_k, a_j) \geq \max\{F(a_k, a_i), F(a_i, a_j)\} \geq \frac{1}{2}$$

And hence, as $F(a_k, a_j) = 1 - F(a_j, a_k)$, we have:

$$F(a_j, a_k) \leq \min\{F(a_i, a_k), F(a_k, a_i)\} \leq F(a_i, a_k)$$

So for all a_k we have that $F(a_i, a_k) \geq F(a_j, a_k)$. Hence $\sum_{a_k \neq a_i} F(a_i, a_k) \geq \sum_{a_k \neq a_j} F(a_j, a_k)$ and so $a_i \geq_B a_j$.

Now we need to prove the converse, i.e:

$$a_i \geq_B a_j \implies a_i \succeq a_j \tag{A.3}$$

Our approach here is slightly different. First, we show that:

$$B(a_i) = B(a_j) \iff a_i = a_j \tag{A.4}$$

If $a_i = a_j$ it is trivial that $B(a_i) = B(a_j)$. So now let us suppose that $a_i \neq a_j$. As \succeq is a total ordering, exactly one of either $a_i \succeq a_j$ or $a_j \succeq a_i$ must be true, so let us assume without loss of generality that $a_i \succeq a_j$. Now, as $a_i \neq a_j$, and

alternatives are *Pairwise Distinguishable* under F , there exists some $a_{k'}$ such that $F(a_i, a_{k'}) \neq F(a_j, a_{k'})$. As we have shown above, $\forall a_k, F(a_i, a_k) \geq F(a_j, a_k)$, and thus, we must have that for $a_{k'}$, $F(a_i, a_{k'}) > F(a_j, a_{k'})$ and so $B(a_i) > B(a_j)$. By contraposition, this proves Equation (A.4).

As \succeq is reflexive, Equation (A.4) implies that for $a_i = a_j$:

$$a_i \geq_B a_j \implies a_i \succeq a_j$$

So it only remains to show this for $a_i \neq a_j$. But, in proving Equation (A.4), we have already shown that if $a_i \neq a_j$:

$$a_i \succeq a_j \implies B(a_i) > B(a_j) \tag{A.5}$$

Thus the contrapositive of Equation (A.5) is also true, specifically:

$$B(a_j) \geq B(a_i) \implies \neg(a_i \succeq a_j) \implies a_j \succeq a_i$$

With the last implication justified by the totality of the ordering \succeq . □

So, if we have Strong Stochastic Transitivity (SST), and Pairwise Distinguishability (PD), given sufficiently accurate estimates for pairwise means, the Borda Score ranking for alternatives will be consistent with an underlying latent ranking, if one exists. We should consider how reasonable these two conditions are.

Firstly, remember that the only method for gaining information about alternatives is through pairwise measurements. Therefore, if PD does not hold, and there are some alternatives for which all pairwise means are identical, then it is impossible for us to gain any information that would allow us to distinguish one such alternative from another. Thus, PD is a requirement for any pairwise ranking problem where we might wish to create a total ordering of alternatives.

The SST condition is a common condition for modelling pairwise ranking

and selection problems, being either explicitly or implicitly assumed in many of the current methods discussed in Section 3. It also holds for the most common latent fitness models (BTL and Thurstone), as well as generally being consistent with observed real-world pairwise preference data [86]. Note that the POCBAM and PKG methods proposed in this paper are highly general, and do not require that the SST condition holds for pairwise outcomes, it is only required to guarantee equivalence between the Borda Score ordering and an underlying latent ordering, if one exists.

Proof of Inequality 3.3 Let $Y \sim \mathcal{N}(M, \Sigma)$ be the joint distribution of the alternative score estimates of the current top- κ alternatives ($M_i = \tilde{\mathcal{S}}_i$), and define $Y^* \sim \mathcal{N}(M, \Sigma^*)$ to be the multivariate Gaussian distribution with equal mean to Y , and covariance matrix defined by:

$$\Sigma_{i,j}^* = \begin{cases} \Sigma_{i,j}, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Then $\Sigma_{i,j}^* \geq \Sigma_{i,j}, \forall i, j$. Hence by Theorem 2.1.1, Corollary 1 of [94]:

$$\mathbb{P}_Y \left[\bigcap_{i=1}^{\kappa} \{\tilde{\mathcal{S}}_i \geq a_i\} \right] \leq \mathbb{P}_{Y^*} \left[\bigcap_{i=1}^{\kappa} \{\tilde{\mathcal{S}}_i \geq a_i\} \right] = \prod_{i=1}^{\kappa} \mathbb{P}(\tilde{\mathcal{S}}_i \geq a_i)$$

Holds for any $(a_1, \dots, a_{\kappa}) \in \mathbb{R}^{\kappa}$. The proof of the second inequality for alternatives outside the current top- κ follows from Theorem 2.1.1 of [94] by an almost identical argument.

APPENDIX B

Kullback-Leibler Knowledge Gradient

In this appendix, we propose to integrate into CMA-ES a specifically adapted version of a recent ranking and selection technique, the Knowledge Gradient (KG) [40]. More specifically, we use a CMA-ES with $(\mu/\mu, \lambda)$ selection here, although an extension of our method to more advanced versions of CMA-ES should be straightforward. With $(\mu/\mu, \lambda)$ selection, in every generation, a subset of the μ best solutions has to be selected from λ offspring. Then, the next generation's offspring is generated from a multi-variate Gaussian derived from the distribution of the selected μ individuals.

A simple way to use ranking and selection in this framework would be to use a subset-selection technique similar to [25] or the Hoeffding-Bernstein race in [53] to maximize the probability of correctly identifying the best μ individuals. However, we go beyond this straightforward mechanism and instead attempt to quantify the severity of selection errors made. In particular, we propose to measure the impact selection errors have on the distribution used to generate the next generation's offspring. Then, we propose to allocate evaluations in order to minimize the expected divergence between the probability distribution derived from the selected individuals, and the probability distribution based on the true top μ individuals, and we design a variant of the KG method to achieve this.

To summarize, our contributions are as follows:

1. We propose a KG version for top- μ subset selection and demonstrate the benefit of integrating KG into CMA-ES in noisy environments.
2. We propose another variant of KG that, rather than maximizing the probability of correctly identifying the top μ individuals minimizes the expected divergence of the offspring probability distribution from the desired offspring probability distribution.
3. We empirically investigate the performance of the proposed algorithm in various settings.

This appendix is structured as follows. We start in the next section describing in detail the problem setting, assumptions made and performance measure used. Our algorithm is proposed in Section B.2. Empirical results are summarized in Section B.3. The paper concludes with a summary and some ideas for future work.

B.1 Model

We consider the problem of accurately identifying a subset I_μ , of size μ , of individuals from an offspring population $P = \{a_1, \dots, a_\lambda\}$ for CMA-ES, where the top- μ subset is used to produce a sampling distribution for the next offspring generation. Each individual a_i is modeled as a vector $a_i = (\alpha_1^i, \dots, \alpha_d^i) \in \mathbb{R}^d$. Individuals are evaluated on their ability to perform a task, modeled as an unknown function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ mapping individuals to fitness values. Samples are allocated sequentially; at each time step t , we can select an individual a_i for sampling, whereby we receive a measurement x_i of $F(a_i)$, albeit one perturbed by noise ϵ :

$$x_i = F(a_i) + \epsilon$$

In this paper, we assume this noise to be Gaussian distributed, with mean 0, known variance σ^2 and IID across all individuals. We estimate the fitness of an individual a_i using the sample mean of all previous n_i measurements of a_i :

$$S_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_i^j$$

After a finite sampling budget of N samples, we select a subset I_μ containing μ individuals based on their fitness estimates to produce the mean \mathbf{m} and covariance matrix Σ of the CMA-ES sampling distribution. Here we use a simple form of CMA-ES using the mean of the selected individuals and diagonalized covariance matrix:

$$\mathbf{m} = \frac{1}{\mu} \sum_{a_i \in I_\mu} a_i$$

$$\Sigma_{j,j} = \frac{1}{\mu - 1} \sum_{a_i \in I_\mu} [\alpha_j - \mathbf{m}_j][\alpha_j - \mathbf{m}_j]$$

Without noise, we would select I_μ to contain the individuals with the best values of F . In the context of function minimization, this would be the μ individuals a_i with the lowest values of $F(a_i)$, i.e:

$$I_\mu^* = \underset{I_\mu \subset P}{\operatorname{argmin}} \sum_{a_i \in I_\mu} F(a_i)$$

However, as $F(a_i)$ is unknown, we must instead use our individual fitness estimates S_i . Incorrectly omitting an individual in favor of another changes the parameters of the distribution used by the evolutionary algorithm to generate the next generation of offspring. To characterize this error, we use the Kullback-Leibler divergence between the chosen and desired distributions. For continuous distributions P and

Q with density functions p and q respectively, this is defined as [11]:

$$D_{KL}(P||Q) = \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

For d -dimensional multivariate Gaussian distributions (\mathbf{m}_1, Σ_1) and (\mathbf{m}_2, Σ_2) , this can be calculated using [33]:

$$D_{KL}((\mathbf{m}_1, \Sigma_1)||(\mathbf{m}_2, \Sigma_2)) = \frac{1}{2} \log \left(\frac{\det \Sigma_2}{\det \Sigma_1} - d + \text{tr}((\Sigma_2)^{-1} \Sigma_1) \right) + (\mathbf{m}_2 - \mathbf{m}_1)^T (\Sigma_2)^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$$

The optimization problem is therefore how to minimize the divergence between the chosen and noiseless next generation CMA-ES sampling distribution.

B.2 Algorithm

To approach this problem, we propose an adapted version of the Knowledge Gradient (KG) sampling framework. KG refers to a class of Bayesian one-step look-ahead policies first proposed in [47] and further developed in [40] and [32]. KG attempts to maximize the expected value of performing an additional sample under the assumption that after this, no additional samples will be taken. In the context of top- μ selection, the standard KG policy would typically attempt to myopically maximize the increase in Probability of Correct Selection (PCS) resulting from each sample allocation. Under the KG assumption that the sampling process will terminate immediately after, should taking the sample fail to change our estimate of the current top- μ , we have, in effect, gained no new information from the sample. Thus, if I_μ^t is our current top- μ subset after t samples, and $I_\mu^{t+1,i}$ the subset after an additional sample has been allocated to individual a_i , the KG policy defines the value

of sampling individual a_i as:

$$V^{a_i} = \mathbb{P}[I_\mu^{t+1, a_i} = I_\mu^* | I_\mu^t \neq I_\mu^*] - \mathbb{P}[I_\mu^{t+1, a_i} \neq I_\mu^* | I_\mu^t = I_\mu^*]$$

As I_μ^* is unknown at the time of sampling, these probabilities cannot be calculated. However, KG makes the further assumption that as we gain information by performing our sample, we are unlikely to discard a correct subset for an incorrect one:

$$\mathbb{P}[I_\mu^{t+1, a_i} = I_\mu^* | I_\mu^t \neq I_\mu^*] \gg \mathbb{P}[I_\mu^{t+1, a_i} \neq I_\mu^* | I_\mu^t = I_\mu^*] \approx 0$$

Thus:

$$V^{a_i} \approx \mathbb{P}[I_\mu^{t+1, a_i} \neq I_\mu^t]$$

Under this assumption, maximising the probability of a sample changing the top- μ subset is equivalent to maximising the expected increase in PCS. By maintaining a distribution over sample outcomes for each individual, we can estimate the probability that sampling a particular individual will sufficiently change our estimate of that individual's score to cause a change in our top- μ subset. If the individual is currently outside the subset, a sample would need to sufficiently increase its estimated score (the mean of all previous sample results for the individual) to be higher than that of the current μ^{th} best, thereby moving it into the selected subset. Similarly, for an individual inside the selected subset, a sample would need to decrease the individual's score estimate to below that of the $\mu + 1^{th}$ best individual score estimate to cause it to drop out of the selected subset. Thus, if S_μ and $S_{\mu+1}$ denote the current μ^{th} and $\mu + 1^{th}$ best score estimates, the sample result of an individual a_i needed to change I is given by:

$$\delta_i = \begin{cases} S_\mu - S_i & \text{if } a_i \notin I_\mu^t \\ S_i - S_{\mu+1} & \text{if } a_i \in I_\mu^t \end{cases}$$

And so, if the individual a_i has so far been sampled n_i times, the sampling result needed to cause a change in S_i of at least δ_i is given by:

$$\gamma_i = (n_i + 1)\delta_i + S_i$$

Thus, under our Bayesian framework, the estimated probability of the sample changing the current top- μ subset is given by:

$$\mathbb{P}[I_\mu^{t+1,i} \neq I_\mu^t] = 1 - \Phi\left(\frac{|\gamma_i - S_i|}{\sigma}\right)$$

Where Φ is the cumulative distribution function of the standard normal distribution.

However, using standard KG to maximize the PCS of our top- μ subset is potentially a sub-optimal approach. PCS treats all errors in selection equally, caring only whether the selected subset is exactly correct or not. When the aim is to select individuals for CMA-ES, we would ideally wish to minimize the effect of any selection errors on the sampling distribution of the next generation. To do this, we propose adapting KG to instead consider the expected divergence between the CMA-ES distribution before and after sampling:

$$\begin{aligned} V^{a_i} &= \mathbb{E}[D_{KL}((\mathbf{m}^i, \Sigma^i) || (\mathbf{m}, \Sigma))] \\ &= \mathbb{P}[I_\mu^{t+1,a_i} \neq I_\mu^t] \times D_{KL}((\mathbf{m}^i, \Sigma^i) || (\mathbf{m}, \Sigma)) \end{aligned}$$

Where (\mathbf{m}, Σ) and (\mathbf{m}^i, Σ^i) are the mean vector and covariance matrices of the CMA-ES sampling distributions pre- and post-sampling individual a_i . Under the same assumptions as the standard form of KG, sampling where V^{a_i} is maximized is equivalent to maximizing the expected decrease in $D_{KL}(\mathbf{m}^*, \Sigma^*) || (\mathbf{m}, \Sigma)$, where (\mathbf{m}^*, Σ^*) is the distribution of the correct top- μ subset I_μ^* . We call the policy that sequentially allocates samples to the individual $\operatorname{argmax}_{a_i \in P} (V^{a_i})$ the *Kullback-Leibler Knowledge Gradient* (KL-KG) policy.

Table B.1: Kullback-Leibler Knowledge Gradient (KL-KG)

INPUT:	Set of λ individuals $\{a_1, \dots, a_\lambda\}$, Required subset selection size μ , Total sampling budget N ,
INITIALIZE:	Perform n_0 samples of each individual; $n_i = n_0$ for all a_i
LOOP:	WHILE $\sum_i n_i < N$ DO:
UPDATE:	Individual fitness estimates $S_i = \frac{1}{n_i} \sum x_i$, Recalculate index set I_μ^t of best μ individuals. Recalculate CMA-ES distribution (\mathbf{m}, Σ) of I_μ^t .
FOR ALL INDIVIDUALS a_i :	
UPDATE:	Required score changes δ_i , sampling results γ_i $\mathbb{P}[I_\mu^{t+1,i} \neq I_\mu^t] = 1 - \Phi\left(\frac{ \gamma_i - S_i }{\sigma}\right)$ Calculate possible distribution (\mathbf{m}^i, Σ^i) changed by sampling a_i $V^{a_i} = \mathbb{P}[I_\mu^{t+1,i} \neq I_\mu^t] \times D_{KL}((\mathbf{m}^i, \Sigma^i) (\mathbf{m}, \Sigma))$
SELECT:	Sample individual a_i that maximizes V^{a_i} , $n_i \leftarrow n_i + 1$
END LOOP	
RETURN	I_μ

B.3 Empirical Testing

In this section, we test the performance of KL-KG against both standard Knowledge Gradient sampling and uniform sample allocation on several well known test functions.

B.3.1 Single Generation

We begin by investigating whether KL-KG can indeed reduce the divergence between the mean and covariance of the final selected top- μ subset with the subset that would be selected without noise. In each experiment, a population of 20 individuals $P = \{a_1, a_2, \dots, a_{20}\}$, $a_i \in \mathbb{R}^2$ was randomly generated, with each value α_j^i being drawn from a Gaussian distribution with mean 5 and variance 5^2 . This is to ensure that the population is shifted away from the optimum of each test function. We set $\mu = 10$ for the selected subset size and $n_0 = 1$ for the initial warm-up samples for both KL-KG and KG.

The following test functions were used:

- Sphere Function:

$$F_{sp}(a) = \alpha_1^2 + \alpha_2^2$$

- Rastrigin Function [78]:

$$F_{ra}(a) = 20 + \sum [\alpha_j^2 - 10\cos(2\pi\alpha_j)]$$

- Rosenbrock Function [81]:

$$F_{ro}(a) = 100(\alpha_2 - \alpha_1^2)^2 + (\alpha_1 - 1)^2$$

- Ackley Function [1]:

$$F_{ac}(a) = -20e^{-0.2\sqrt{0.5(\alpha_1^2 + \alpha_2^2)}} - e^{0.5[\cos(2\pi\alpha_1) + \cos(2\pi\alpha_2)]} + e + 20$$

- Schaffer F7 Function [36]:

$$F_{sch}(a) = \left(\frac{1}{d-1} \sum_{i=1}^{d-1} \sqrt{s_i} + \sqrt{s_i} \sin^2(50s_i^{1/5}) \right)^2$$

where $s_i = \sqrt{\alpha_i^2 + \alpha_{i+1}^2}$

- Levy 13 Function [62]:

$$F_{le} = \sin^2(2\pi\alpha_1) + (\alpha_1 - 1)^2[1 + \sin^2(3\pi\alpha_2)]$$

$$+ (\alpha_2 - 1)^2[1 + \sin^2(2\pi\alpha_2)]$$

To measure performance, we compare average pairwise differences in divergence from the correct CMA-ES distribution (\mathbf{m}^*, Σ^*) between KL-KG and the comparison policies over 10,000 different randomly generated initial populations for a range of noise levels and sampling budgets, with results shown in Figures B.1 and B.2.

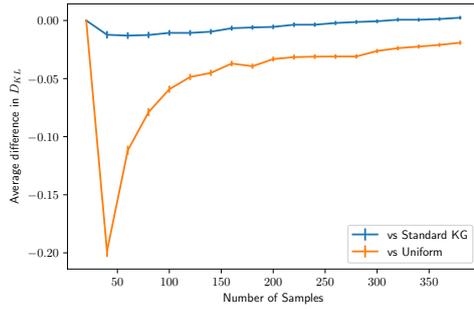
In Figure B.1, noise levels σ were chosen to be generally proportional to the magnitude of each test function around the center of the population. For the Sphere and Rastrigin functions, $\sigma = 100$ was used, for the Rosenbrock function, $\sigma = 1000$, $\sigma = 10$ for the Ackley function, $\sigma = 50$ for the Schaffer F7 function and $\sigma = 25$ for the Levy 13 function.

Initially, with 20 samples, the performance of all three methods is identical, as the sampling budget is entirely uniformly allocated to the initial n_0 warm-up samples.

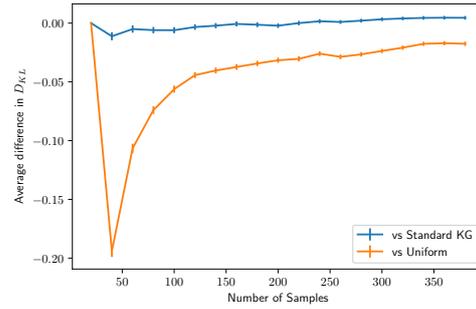
As a myopic sampling method, KL-KG should be highly efficient with small sampling budgets, as it should be optimal with only a single sample to allocate. This is proven for standard KG when maximizing PCS in [40]. Here we see KL-KG does improve performance most for smaller N , especially when compared to uniform allocation. Eventually, as $N \rightarrow \infty$, all methods will converge to the correct solution so the pairwise differences between them converge to zero.

The Rosenbrock function is the only function where we do not observe clear improvement from KL-KG over uniform sampling. Away from the global minimum the slope of the Rosenbrock function rapidly becomes very steep in one dimension. This allows the sampling methods to distinguish between individuals reliably, even with only a few samples. Thus, intelligent sampling is not required and we see much smaller differences in divergence for this function.

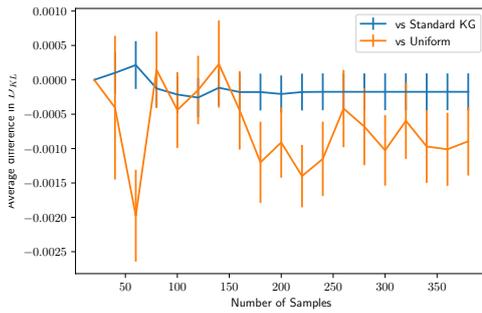
In Figure B.2, we see the effect of varying noise level σ for a fixed sampling budget $N = 200$. For low sigma, all methods make very few mistakes, as it is easy to learn the true fitness values of individuals with low noise. As the noise level increases, the benefit of using KL-KG increases, showing a clear advantage over both KG and uniform sampling for the Sphere, Rastrigin and Ackley functions, and over uniform sampling for the Schaffer F7 and Levy 13 functions. As σ grows very large relative to the magnitude of $F(a)$, it becomes progressively more difficult to gain meaningful information on individual fitness scores from the noise. Eventually as $\sigma \rightarrow \infty$, sampling information converges to zero and all sampling methods become equivalent to uniform allocation. This seems to be the case with the Ackley function for higher σ values, where the difference between methods begins to decrease after initially growing.



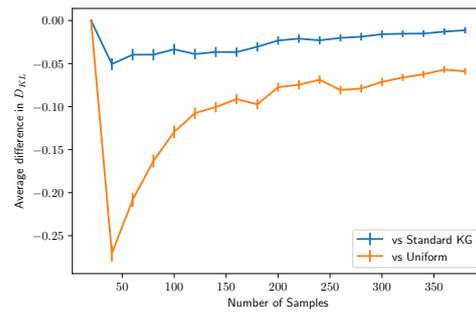
(a) Sphere Function



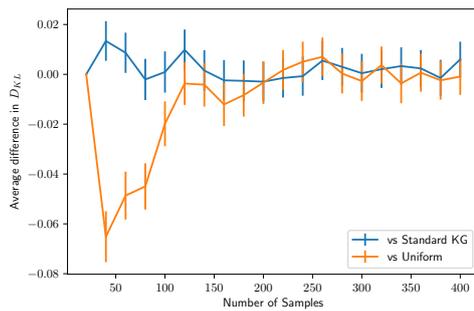
(b) Rastrigin Function



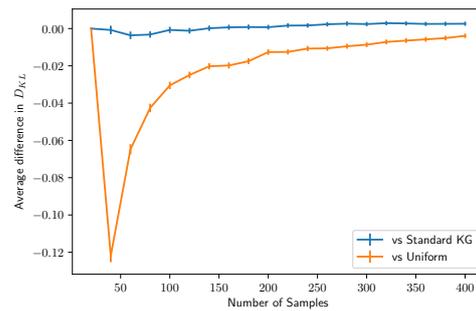
(c) Rosenbrock Function



(d) Ackley Function

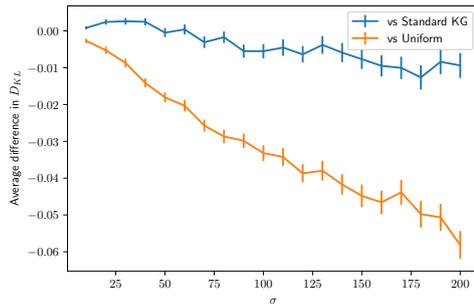


(e) Schaffer F7 Function

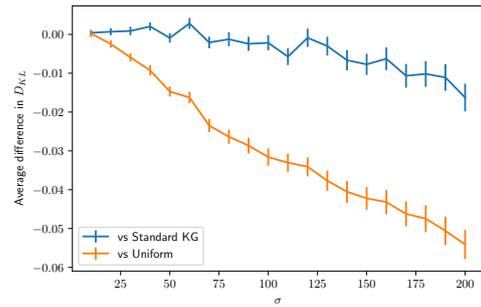


(f) Levy 13 Function

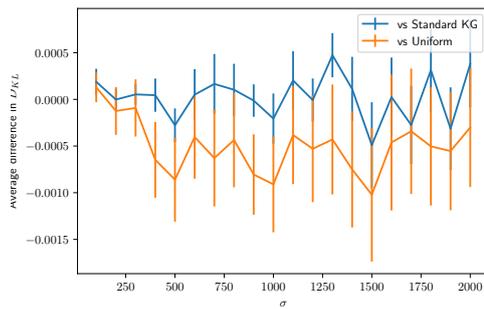
Figure B.1: Pairwise difference in K-L divergence for KL-KG against standard KG and Uniform sample allocation for sampling budgets between 20 and 400. Averaged over 10,000 repetitions.



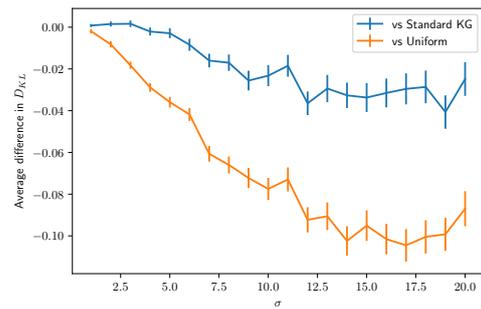
(a) Sphere Function



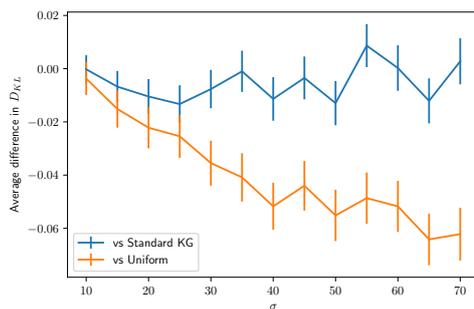
(b) Rastrigin Function



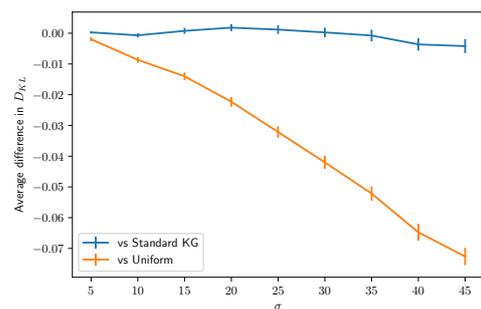
(c) Rosenbrock Function



(d) Ackley Function



(e) Schaffer F7 Function



(f) Levy 13 Function

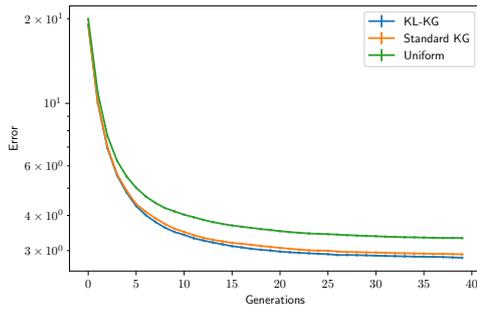
Figure B.2: Pairwise difference in K-L divergence for KL-KG compared to standard KG and Uniform sample allocation for different noise levels. $N = 200$.

Table B.2: Statistical comparison of performance. Standard errors for the difference between the means are given in parenthesis, P values show the result of a two-sided T-test.

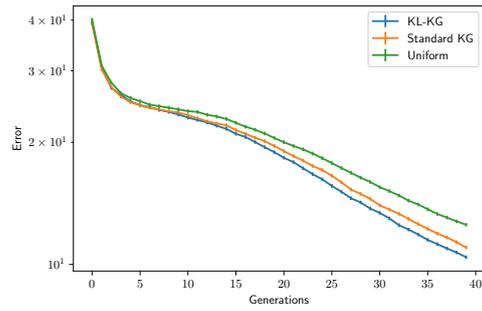
Fn.	Improvement over Uniform (generation 40)				Improvement over KG	
	KL-KG	P	KG	P	KL-KG	P
F_{sp}	0.495 (0.044)	$< 10^{-5}$	0.415 (0.044)	$< 10^{-5}$	0.080 (0.041)	0.051
F_{ra}	2.108 (0.163)	$< 10^{-5}$	1.516 (0.167)	$< 10^{-5}$	0.592 (0.152)	$< 10^{-4}$
F_{ro}	2.182 (0.217)	$< 10^{-5}$	2.156 (0.217)	$< 10^{-5}$	0.026 (0.201)	0.90
F_{ac}	0.263 (0.021)	$< 10^{-5}$	0.230 (0.021)	$< 10^{-5}$	0.033 (0.019)	0.082
F_{sch}	8.216 (2.174)	0.00016	6.533 (2.179)	0.0027	1.683 (2.120)	0.43
F_{le}	3.683 (0.217)	$< 10^{-5}$	3.737 (0.217)	$< 10^{-5}$	-0.053 (0.201)	0.79

B.3.2 Multiple Generation Optimization

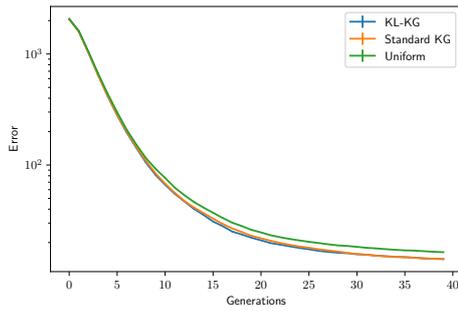
In this section, we test whether the reduction in K-L divergence between the estimated top- μ subsets and the true top- μ subsets improves the quality of the individuals that CMA-ES is able to evolve over multiple generations. Using the same test functions as in the previous experiments, starting with an initial population of 20 randomly generated individuals $a_i \in \mathbb{R}^2$, again with each individual's parameters sampled from a Gaussian distribution with mean 5 and variance 5^2 . For each function, $\mu = 10$ and $N = 200$ per generation. For the Sphere and Rastrigin functions, the noise level is set to $\sigma = 100$; for the Rosenbrock function, $\sigma = 1000$; for the Ackley function, $\sigma = 10$; for the Schaffer F7 function, $\sigma = 50$; and for the Levy 13 function, $\sigma = 25$. Each optimization was performed for 40 generations, with the results averaged over 10,000 replications and shown in Figure B.3. Both KL-KG and KG seem to offer significant improvements over uniform allocation for all the test functions. For the Sphere, Rastrigin and Ackley functions, KL-KG also performs better than standard KG, offering a small reduction in error for the Sphere and Ackley functions, but only the clear improvement for the Rastrigin function is statistically significant at the 95% level. Improvement over Uniform sampling and statistical significance tests of each of the methods are summarized in Table B.2.



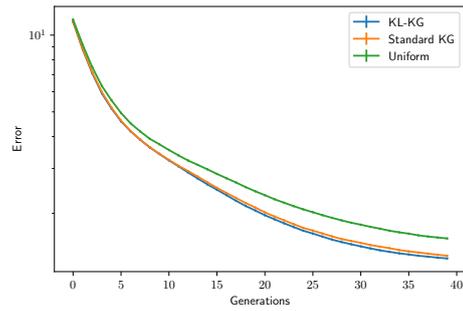
(a) Sphere Function



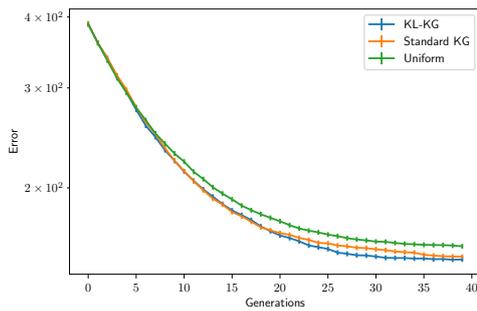
(b) Rastrigin Function



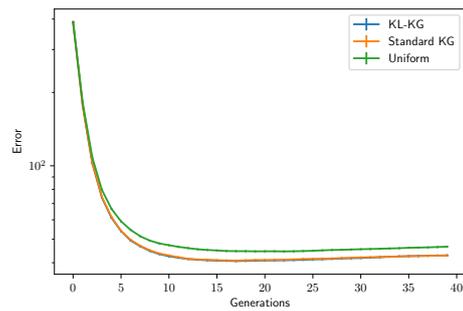
(c) Rosenbrock Function



(d) Ackley Function



(e) Schaffer F7 Function



(f) Levy 13 Function

Figure B.3: Performance of CMA-ES optimization for each function and sampling method over 40 generations. Error refers to the difference between a function's value evaluated at the mean of the CMA-ES distribution and the global minimum.

B.4 Summary and Future Work

We have proposed a sequential sampling mechanism specifically designed to be integrated into CMA-ES for noisy optimization problems. This sampling mechanism is based on the Knowledge Gradient idea and iteratively and myopically allocates the next fitness evaluation to the individual that promises the largest information gain for the CMA’s selection step. More specifically, we proposed a KG version that maximizes the probability of correctly selecting the best μ individuals, and another version that minimizes the Kullback-Leibler divergence between the distribution used for generating the next offspring population based on the μ selected individuals, and the distribution that CMA-ES would have created from the true μ best individuals in the absence of noise. Empirical results demonstrate the efficiency gain that can be obtained by integrating sequential sampling into CMA-ES, and that our specifically designed sampling scheme based on K-L divergence can work better than the more standard KG algorithm based on probability of correct selection.

Currently, we have only tested the KL-KG method using a very basic version of the CMA-ES algorithm. Several more advanced variants exist [49], designed to adaptively exploit correlations between steps on the evolution path. In theory, it should be very simple to incorporate KL-KG into these methods so long as they maintain an explicit representation of the sampling distribution that can be recalculated to account for changes to the selected subset.

Another possible extension would be to consider sampling problems with unknown variance, and non-Gaussian distributions. Variants of Knowledge Gradient that account for unknown variance already exist [32]. Allowing our method to exploit additional information from differences in variance could potentially further improve performance over uniform sample allocation. Non-Gaussian distributions could be tackled by batching.

Bibliography

- [1] D. H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. Springer, 1987.
- [2] M. Adler, P. Gemmell, M. Harchol-Balter, R. M. Karp, and C. Kenyon. Selection in the presence of noise: The design of payoff systems. In *SODA*, pages 564–572, 1994.
- [3] J.-Y. Audibert, R. Munos, and C. Szepesvári. Tuning bandit algorithms in stochastic environments. In *International conference on algorithmic learning theory*, pages 150–165. Springer, 2007.
- [4] T. Bäck, G. Rudolph, and H.-P. Schwefel. Evolutionary programming and evolution strategies: Similarities and differences. In *In Proceedings of the Second Annual Conference on Evolutionary Programming*. Citeseer, 1993.
- [5] T. Ballinger and N. Wilcox. Decisions, error and heterogeneity. *The Economic Journal*, 107(443):1090–1105, 1997.
- [6] T. Bartz-Beielstein, D. Blum, and J. Branke. Particle swarm optimization and sequential sampling in noisy environments. In K. Doerner, M. Gendreau, P. Greistorfer, W. Gutjahr, R. Hartl, and R. M., editors, *Metaheuristics*. Springer, 2007.
- [7] H. Bastani and M. Bayati. Online decision making with high-dimensional covariates. *Operations Research*, 2019.

- [8] E. Benhamou, J. Atif, R. Laraki, and A. Auger. A discrete version of cma-es. *Available at SSRN 3307212*, 2018.
- [9] H.-G. Beyer. Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):239–276, 2000.
- [10] D. Billings, D. Papp, J. Schaeffer, and D. Szafron. Poker as a testbed for ai research. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 228–238. Springer, 1998.
- [11] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.
- [12] J. C. Borda. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781*, 1784.
- [13] R. Bradley and M. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- [14] J. Branke, S. Chick, and C. Schmidt. Selecting a selection procedure. *Management Science*, 53(12):1916–1932, 2007.
- [15] J. Branke and J. Elomari. Racing with a fixed budget and a self-adaptive significance level. In *Learning and Intelligent Optimization*, pages 272–280. Springer, 2013.
- [16] J. Branke and C. Schmidt. Selection in the presence of noise. In *Genetic and Evolutionary Computation Conference*, pages 766–777. Springer, 2003.
- [17] M. Braverman and E. Mossel. Noisy sorting without resampling. In *19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 268–276. Society for Industrial and Applied Mathematics, 2008.

-
- [18] R. Busa-Fekete, T. Fober, and E. Hüllermeier. Preference-based evolutionary optimization using generalized racing algorithms. In *23rd Workshop on Computational Intelligence; Dortmund*, pages 237–245. KIT Scientific Publishing, 2013.
- [19] R. Busa-Fekete and E. Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *International Conference on Algorithmic Learning Theory*, pages 18–39. Springer, 2014.
- [20] R. Busa-Fekete, E. Hüllermeier, and B. Szörényi. Preference-based rank elicitation using statistical models: The case of mallows. In *31st International Conference on Machine Learning*, pages 1071–1079, 2014.
- [21] R. Busa-Fekete, B. Szorenyi, W. Cheng, P. Weng, and E. Hüllermeier. Top-k selection based on adaptive sampling of noisy preferences. In *International Conference on Machine Learning*, pages 1094–1102, 2013.
- [22] K. Chellapilla and D. B. Fogel. Evolving neural networks to play checkers without relying on expert knowledge. *IEEE transactions on neural networks*, 10(6):1382–1391, 1999.
- [23] K. Chellapilla and D. B. Fogel. Evolving an expert checkers playing program without using human expertise. *IEEE Transactions on Evolutionary Computation*, 5(4):422–428, 2001.
- [24] C.-H. Chen. A lower bound for the correct subset-selection probability and its application to discrete event simulations. *IEEE Transactions on Automatic Control*, 41(8):1227–1231, 1996.
- [25] C.-H. Chen, D. He, M. Fu, and L. H. Lee. Efficient simulation budget allocation for selecting an optimal subset. *Inform Journal on Computing*, 20(4):579–595, 2008.

- [26] C.-h. Chen and L. H. Lee. *Stochastic simulation optimization: an optimal computing budget allocation*, volume 1. World scientific, 2011.
- [27] C. H. Chen, J. Lin, E. Yücesan, and S. Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems: Theory and Applications*, 10(3):251–270, 2000.
- [28] C. H. Chen, E. Yücesan, L. Dai, and H. C. Chen. Efficient computation of optimal budget allocation for discrete event simulation experiment. *IIE Transactions*, 42(1):60–70, 2010.
- [29] H.-C. Chen, C.-H. Chen, L. Dai, and E. Yücesan. New development of optimal computing budget allocation for discrete event simulation. In *Winter Simulation Conference*, pages 334–341. Citeseer, 1997.
- [30] X. Chen, S. Gopi, J. Mao, and J. Schneider. Competitive analysis of the top-k ranking problem. In *28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1245–1264. SIAM, 2017.
- [31] Y. Chen and C. Suh. Spectral mle: Top-k rank aggregation from pairwise comparisons. In *32nd International Conference on Machine Learning*, pages 371–380, 2015.
- [32] S. Chick, J. Branke, and C. Schmidt. Sequential sampling to myopically maximize the expected value of information. *Journal on Computing*, 22(1):71–80, 2010.
- [33] J. Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 2007.
- [34] A. E. Elo. *The rating of chessplayers, past and present*. Arco Pub., 1978.
- [35] M. Falahatgar, A. Jain, A. Orlitsky, V. Pichapati, and V. Ravindrakumar. The

-
- limits of maxing, ranking, and preference learning. In *International Conference on Machine Learning*, pages 1426–1435, 2018.
- [36] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Presentation of the noisy functions. Technical report, Technical Report 2009/21, Research Center PPE, 2010.
- [37] P. Fishburn. Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical psychology*, 10(4):327–352, 1973.
- [38] P. Frazier and W. Powell. Asymptotic optimality of sequential sampling policies for bayesian information collection. *Submitted for publication*, 2008.
- [39] P. I. Frazier and W. B. Powell. Paradoxes in learning and the marginal value of information. *Decision Analysis*, 7(4):378–403, 2010.
- [40] P. I. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.
- [41] J. E. Gentle. *Computational statistics*, volume 308. Springer, 2009.
- [42] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.
- [43] M. Groves and J. Branke. Optimal subset selection with pairwise comparisons. In *Proceedings of the DA2PL’2016 EURO Mini Conference*, pages 15–20. EURO, 2016.
- [44] M. Groves and J. Branke. Sequential sampling for noisy optimisation with cma-es. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1023–1030. ACM, 2018.

- [45] S. Guo, S. Sanner, T. Graepel, and W. Buntine. Score-based bayesian skill learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 106–121. Springer, 2012.
- [46] S. S. Gupta and K. J. Miescke. Bayesian look ahead one stage sampling allocations for selecting the largest normal mean. *Statistical Papers*, 35(1):169–177, 1994.
- [47] S. S. Gupta and K. J. Miescke. Bayesian look ahead one stage sampling allocations for selecting the largest normal mean. *Statistical Papers*, 35(1):169–177, 1994.
- [48] B. Hajek, S. Oh, and J. Xu. Minimax-optimal inference from partial rankings. In *Advances in Neural Information Processing Systems 27*, pages 1475–1483. Curran Associates, Inc., 2014.
- [49] N. Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.
- [50] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *ArXiv e-prints*, arXiv:1603.08785, 2016.
- [51] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.
- [52] R. Heckel, N. Shah, K. Ramchandran, and M. Wainwright. Active ranking from pairwise comparisons and when parametric assumptions don’t help. *arXiv preprint arXiv:1606.08842v2 [cs.LG]*, 2016.
- [53] V. Heidrich-Meisner and C. Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *26th International Conference on Machine Learning*, pages 401–408. ACM, 2009.

-
- [54] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.
- [55] M. Jang, S. Kim, C. Suh, and S. Oh. Top- k ranking from pairwise comparisons: When spectral ranking is optimal. *arXiv preprint arXiv:1603.04153v1 [cs.LG]*, 2016.
- [56] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - a survey. *IEEE T-EVC*, 9(3):303–317, 2005.
- [57] M. Johanson. Measuring the size of large no-limit poker games. *arXiv preprint arXiv:1302.7008*, 2013.
- [58] E. D. D. Jong and J. B. Pollack. Ideal evaluation from coevolution. *Evolutionary computation*, 12(2):159–192, 2004.
- [59] B. Kamiński. Refined knowledge-gradient policy for learning probabilities. *Operations Research Letters*, 43(2):143–147, 2015.
- [60] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [61] L. H. Lee, E. P. Chew, S. Teng, and Y. Chen. Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. *European Journal of Operational Research*, 189:476–491, 2008.
- [62] A. V. Levy and A. Montalvo. The tunneling algorithm for the global minimization of functions. *SIAM Journal on Scientific and Statistical Computing*, 6(1):15–29, 1985.
- [63] X. Li and R. Miikkulainen. Evolving adaptive poker players for effective opponent exploitation. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

- [64] X. Li and R. Miikkulainen. Opponent modeling and exploitation in poker using evolved recurrent neural networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 189–196. ACM, 2018.
- [65] G. J. Lidstone. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8(182-192):13, 1920.
- [66] M. Littman and M. Zinkevich. The 2006 aai computer poker competition. *ICGA Journal*, 29(3):166, 2006.
- [67] I. Loshchilov. A computationally efficient limited memory cma-es for large scale optimization. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 397–404. ACM, 2014.
- [68] O. Maron and A. Moore. Hoeffding races: accelerating model selection search for classification and function approximation. *Advances in Neural Information Processing Systems*, pages 59–66, 1994.
- [69] O. Maron and A. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 5(1):193–225, 1997.
- [70] V. Mattila and K. Virtanen. Ranking and selection for multiple performance measures using incomplete preference information. *European Journal of Operational Research*, 242(2):568–579, 2015.
- [71] L. Maystre and M. Grossglauser. Robust active ranking from sparse noisy comparisons. *arXiv preprint arXiv:1502.05556 [stat.ML]*, 2015.
- [72] K. Misra, E. M. Schwartz, and J. Abernethy. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38(2):226–252, 2019.

-
- [73] S. Mohajer, C. Suh, and A. Elmahdy. Active learning for top-k rank aggregation from noisy comparisons. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2488–2497. JMLR. org, 2017.
- [74] S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems 25*, pages 2474–2482. Curran Associates, Inc., 2012.
- [75] W. B. Powell and I. O. Ryzhov. *Optimal learning*, volume 841. John Wiley & Sons, 2012.
- [76] L. Priekule and S. Meisel. A bayesian ranking and selection problem with pair-wise comparisons. In *Proceedings of the 49th conference on Winter simulation*, pages 2149–2160. IEEE Computer Society, 2017.
- [77] P. Rakshit, A. Konar, and S. Das. Noisy evolutionary optimization algorithms-a comprehensive survey. *Swarm and Evolutionary Computation*, 33:18–45, 2017.
- [78] L. A. Rastrigin. Systems of extremal control. *Nauka, Moscow*, 1974.
- [79] C. P. Robert. Intrinsic losses. *Theory and decision*, 40(2):191–214, 1996.
- [80] R. Ros and N. Hansen. A simple modification in cma-es achieving linear time and space complexity. In *International Conference on Parallel Problem Solving from Nature*, pages 296–305. Springer, 2008.
- [81] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.
- [82] C. D. Rosin and R. K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.
- [83] J. Rubin and I. Watson. Computer poker: A review. *Artificial intelligence*, 175(5-6):958–987, 2011.

- [84] S. Samothrakis, S. Lucas, T. P. Runarsson, and D. Robles. Coevolving game-playing agents: Measuring performance and intransitivities. *IEEE Transactions on Evolutionary Computation*, 17(2):213–226, April 2013.
- [85] C. Schmidt, J. Branke, and S. Chick. Integrating techniques from statistical ranking into evolutionary algorithms. In *Applications of Evolutionary Computation*, volume 3907 of *LNCS*, pages 753–762, 2006.
- [86] N. Shah, S. Balakrishnan, A. Guntuboyina, and M. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *33rd International Conference on Machine Learning*, pages 11–20, 2016.
- [87] N. Shah and M. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *arXiv preprint arXiv:1512.08949v2 [cs.LG]*, 2016.
- [88] K. Sims. Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.
- [89] D. Sklansky. *The theory of poker*. Two Plus Two Publishing LLC, 1999.
- [90] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of artificial intelligence research*, 21:63–100, 2004.
- [91] C. Suh, V. Tan, and R. Zhao. Adversarial top- k ranking. *IEEE Transactions on Information Theory*, 63(4):2201–2225, 2017.
- [92] A. Syberfeldt, A. Ng, R. John, and P. Moore. Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling. *European Journal of Operational Research*, 204:533–544, 2010.
- [93] L. L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

-
- [94] Y. L. Tong. *Probability inequalities in multivariate distributions*. Academic Press, 1980.
- [95] T. Urvoy, F. Clerot, R. Féraud, and S. Naamane. Generic exploration and k-armed voting bandits. In *30th International Conference on Machine Learning*, pages 91–99, 2013.
- [96] H. Xiao and L. H. Lee. Simulation optimization using genetic algorithms with optimal computing budget allocation. *Simulation*, 90(10):1146–1157, 2014.
- [97] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.
- [98] Y. Yue and T. Joachims. Beat the mean bandit. In *28th International Conference on Machine Learning*, pages 241–248, 2011.
- [99] S. Zhang, J. Xu, L. H. Lee, E. P. Chew, W. P. Wong, and C. H. Chen. Optimal computing budget allocation for particle swarm optimization in stochastic optimization. *IEEE Transactions on Evolutionary Computation*, 21(2):206–219, 2017.