

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Improving Bayesian Optimization for Quantum Material Control

Permalink

<https://escholarship.org/uc/item/7qg7r73c>

Author

Thombare, Malhar Manohar

Publication Date

2023

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Improving Bayesian Optimization for Quantum Material Control

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Computer Science

by

Malhar Manohar Thombare

March 2023

Thesis Committee:

Christian R. Shelton, Chairperson
Eamonn Keogh
Amit K. Roy-Chowdhury

Copyright by
Malhar Manohar Thombare
2023

The Thesis of Malhar Manohar Thombare is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am grateful to my advisor, without whose help, I would not have been here.

To my parents, brother, and to all those who lent their support.

ABSTRACT OF THE THESIS

Improving Bayesian Optimization for Quantum Material Control

by

Malhar Manohar Thombare

Master of Science, Graduate Program in Computer Science
University of California, Riverside, March 2023
Christian R. Shelton, Chairperson

Using light one can desirably change the properties of materials. An electromagnetic (EM) field signal can excite a material or chemical system into a higher eigenstate and, as a result, change its properties. We develop an algorithm to find the right EM signal for a chemical system simulated in NIC-CAGE (Novel Implementation of Constrained Calculations for Automated Generation of Excitations). NIC-CAGE simulates the influence of an EM field on a quantum system and provides the probability of desired transition as well as the gradient of the probability as a function of the EM field. The simulation is calculated in a system with one degree-of-freedom which still preserves most of the dynamics of a real high degree-of-freedom systems. A single query on the simulator takes about a second to calculate the probability and more time to find the gradient at the evaluation point. Given the parameters of the quantum system, and the source and target eigenstates, the software provides a theoretical frequency for a sinusoidal EM signal. The optimal frequency may lie in a big region around it. Bayesian Optimization (BO) is suitable to this problem because it minimizes the number of costly evaluations and avoids the need for costlier gradient

calculation. However, typical BO methods expect uniform smoothness over all the search space in the evaluation function to be optimized. For our problem, this is not true for the frequency parameter which leads to slower performance. We found that frequencies with high evaluation probability are present in very small regions and the rest of the space has a uniformly low probability. We propose three methods to minimize the search over these low-probability regions. The first method crops the part of the search space based on the last best point and searches in it. The second method extends the previous method by dynamically adjusting the size of cropped search space and searches over it. The third method warps the search region such that the variance of probability is high and more uniform over the search space. We demonstrate the improved performance of our methods on a suite of quantum control problems.

Contents

List of Figures	ix
1 Introduction	1
1.1 Problem statement	1
1.2 Bayesian Optimization (BO)	2
1.2.1 Method Description	3
1.2.2 Model	3
1.2.3 Acquisition Functions	4
1.3 BO for our problem	6
1.4 Room for improvement	6
2 Chapter 2 Related Work	8
2.1 Search space expansion	8
2.2 Using the bounds on evaluation functions	8
3 Our Methods	10
3.1 Notations	10
3.2 Crop	11
3.3 Crop and Expand	12
3.4 Warp	13
4 Chapter 4 Experiments and Results	15
4.1 Experiment Design	15
4.2 Results	16
4.2.1 Generalization of optimization	16
4.2.2 Speed of optimization	17
5 Conclusions	19
Bibliography	21

List of Figures

3.1	Visualization of proposed methods	12
4.1	Generalization accross problems	16
4.2	Speed accross problems	17

Chapter 1

Introduction

1.1 Problem statement

We want to solve the problem of manipulating the quantum state of the material using a time-varying electromagnetic signal. NIC-CAGE [9], the software, calculates the transition probability from a source to a target state, for a single-dimensional quantum system under the influence of an EM signal. Our goal is to find the signal that maximizes this transition probability. NIC-CAGE discretizes the spatial and temporal dimensions of the system. The resulting simulator propagates the state of the system over time, requiring significant computation. Since the development of the system is temporal in nature the calculation over time steps can not be parallelized. Furthermore, the evaluation of the gradient of the transition probability with respect to the EM signal needs even more calculations. A single evaluation in NIC-CAGE takes about a second to complete. The simulation of a more complete system with more degrees of freedom would need hours of calculation. We want to use the minimum number of such evaluations while searching for the optimal signal.

We chose to use Bayesian Optimization (BO) [2] because BO seeks to optimize functions with the fewest number of evaluations and does not require gradient calculations. Although the EM signal can take any arbitrary shape, we are restricting its search space to sinusoidal EM waves parameterized with only a frequency ω and an amplitude a . f is the evaluation function provided by NIC-CAGE based on system parameters, the source, and the target states. f provides the transition probability, p , given ω and a . Our problem hence can be stated as follows.

$$\max_{\omega \in \mathcal{F}, a \in \mathcal{A}} f(\omega, a)$$

\mathcal{F} and \mathcal{A} are one-dimensional parameter spaces for frequency and amplitude respectively. We denote NIC-CAGE's provided theoretical potential frequency for the solution as ω_0 . Ratio R defines the size of the search space for frequency, and R_0 is the initial ratio for the same. The parameter space for frequency is defined by the center of search space ω^c and R as follows.

$$\mathcal{F} = \{\omega | (1 - R)\omega^c \leq \omega \leq (1 + R)\omega^c\}$$

The amplitude space is constant for all the methods and it is as follows.

$$\mathcal{A} = \{a | 0 \leq a \leq 0.15\}$$

1.2 Bayesian Optimization (BO)

BO is a class of methods specially developed for solving the problems of format

$$\max_{x \in \mathcal{A}} f(x).$$

A is a hyper-rectangle of a form $\{x \in \mathbb{R}^d : a_i \leq x_i \leq b_i\}$, typically with d smaller than 20 and f is a continuous function [2]. In common applications of BO, f is a noisy and expensive function; it does not have a known structure such as concavity or linearity; and it does not provide any first-order or second-order derivative information.

1.2.1 Method Description

BO comprises two main components: the surrogate model that models the objective function f and an acquisition function to select the next point to evaluate. For this work we are using a Gaussian Process [8] as the surrogate model parameterized by the mean function and a covariance function. The mean function μ_0 dictates the mean value of f at any given x , and the kernel function Σ_0 given two points, x and y , specifies the covariance between $f(x)$ and $f(y)$. Kernel functions convey the notion that closer points influence each other more than the farther points. To initially build the dataset, f is evaluated at a few, n_0 points mostly selected uniformly at random over the search space. The Gaussian process posterior distribution conditioned on the evaluated points is iteratively used by the acquisition function to acquire new points. Once evaluated the new point, x , and evaluation, $f(x)$, are added to the dataset, and iteration continues with the new dataset.

1.2.2 Model

Consider values of f sampled at points $x_1, x_2, x_3, \dots, x_k \in \mathbb{R}^d$ as $[f(x_1), f(x_2), f(x_3), \dots, f(x_k)]$ represented by $x_{1:k}$ and $f(x_{1:k})$ respectively. The Gaussian process assumes a prior of a multivariate Gaussian distribution over these values with a mean vector of size k and co-

variance matrix of size $k \times k$. The mean vector is the evaluation of a mean function μ_0 at $x_{1:k}$ and the covariance matrix is the evaluation of the covariance function or kernel Σ_0 at (x_i, x_j) where $i, j \in 1 \dots k$. The kernel function is selected in such way that the resultant covariance matrix is always positive semi-definite. It should show high correlation between points closer in space. We can now write a distribution for surrogate $f(x_{1:k})$ as follows.

$$f(x_{1:k}) \sim \mathcal{N}(\mu_0(x_{1:k}), \Sigma_0(x_{1:k}, x_{1:k})). \quad (1.1)$$

Let n be the number of observations made in the past. Then using Bayes' rule on Equation 1.1 we can obtain posterior distribution over the next observation as follows (see details in Chapter 2.1 of [8]):

$$f(x)|f(x_{1:n}) \sim \mathcal{N}(\mu_n(x), \sigma_n^2(x)) \quad (1.2)$$

$$\mu_n(x) = \Sigma_0(x, x_{1:n})\Sigma_0(x, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(x) \quad (1.3)$$

$$\sigma_n^2(x) = \Sigma_0(x, x) - \Sigma_0(x, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}\Sigma_0(x_{1:n}, x) \quad (1.4)$$

Intuitively, the posterior mean $\mu_n(x)$ is a weighted average between the prior $\mu_0(x)$ and the estimate $f(x_{1:n})$ where the weights depend on the kernel. The posterior variance, $\sigma_n^2(x)$, is prior variance $\Sigma_0(x, x)$ minus an estimate based on the kernel.

1.2.3 Acquisition Functions

The acquisition function uses the posterior information from the surrogate model to sample the next point. Each acquisition function uses a different method to define the next best point. For any acquisition function, A , given n observations the next point x_{n+1} is evaluated as follows

$$x_{n+1} = \arg \max_x A(x). \quad (1.5)$$

Below are three of the most common acquisition functions.

Upper Confidence Bound

The upper confidence bound (UCB) acquisition function is based on the upper confidence bound bandit strategy [10]. The value of the point x is defined as

$$\text{UCB}_n(x) = \mu_n(x) + \beta\sigma_n(x) \quad (1.6)$$

where $\mu_n(x)$ and $\sigma_n(x)$ are the posterior of mean and standard deviation after n observations. β controls the influence of the standard deviation on the decision, lower value corresponds to exploiting the mean and a higher value corresponds to exploration where the variance is higher.

Probability of improvement

We define Improvement as

$$I(x) = \max(0, f(x) - f_n^*)$$

where,

$$f_n^* = \max_{i=0}^n f(x_i)$$

Then the probability of improvement (PI) [1] with GP surrogate can be expressed by

$$\text{PI}_n = P(I(X) > 0) = \Phi(\Delta_n(x)). \quad (1.7)$$

where $\Delta_n(x) = \frac{\mu_n(x) - f_n^*}{\sigma_n(x)}$ and Φ is a cumulative distribution function for a unit normal distribution.

Expected Improvement (EI)

EI is similar to PI but optimizes the expected improvement instead of the Probability of Improvement [1]. We can give EI as

$$\text{EI}_n(x) = E[I(x)] \quad (1.8)$$

With GP surrogate, EI in closed form [4] is

$$\text{EI}_n(x) = [\Delta_n(x)]^+ + \sigma_n(x)\phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) - |\Delta_n(x)|\Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) \quad (1.9)$$

where ϕ is a probability density function and Φ is a cumulative distribution function for a unit normal distribution.

1.3 BO for our problem

We have chosen Standard BO [2] with a Gaussian process (GP) surrogate and EI as the acquisition function, as this is the most common BO configuration. We are using a squared-exponential kernel as the covariance function of the GP. We have 40 thousand different simulated systems and for each system we are testing three transitions 0 to 1, 1 to 2, and 0 to 2. We picked 600 systems from this set and tried to search for the best EF for the mentioned transitions. We are using Emukit [7] to implement BO.

1.4 Room for improvement

Upon using Standard BO we found that the search space of frequency had high probability values concentrated in a very small region. The Standard BO kept exploring

the whole search space even after finding this small region. All the evaluations outside this region gave a very small probability. The BO could have performed better if it had searched more in that small region. Our proposed method increases the success rate of Standard BO by maximizing the search in this region.

Chapter 2

Chapter 2 Related Work

2.1 Search space expansion

Large search spaces are one of the main reasons for the slower performance of BO. One popular solution for this problem is starting with a smaller search space and expanding as required. [3] and [5] use information gained over a smaller search space to determine the expansion policy. However, in our case, unless the evaluation is not made near the optimal answer all the evaluations have similar values and do not provide any useful information. This makes it difficult to effectively expand the search space. Our methods start with a larger search space and then exploit the sub-region that gives a better outcome.

2.2 Using the bounds on evaluation functions

A few methods also exploit knowledge of bounds on the optimal value. In our case, since f is a probability we know that the evaluation function has the upper bound of

1 and the lower bound of 0. [6] suggests a technique where these bounds can be used to model the surrogate more accurately. They propose a new acquisition function that uses the upper bound to rectify the estimations where other acquisition functions overestimate the optimal value by crossing this upper bound. In our case, BO almost never finds values near the upper bound which does not create a need for rectification.

We want the standard BO to search in a specific area more often than others which are still difficult with existing methods. Our proposed method exploits such an area as soon as it finds it and brings a significant improvement in BO.

Chapter 3

Our Methods

In Standard BO trials, we observed that there is a small region in the frequency's search space which yields a significantly higher probability than the rest. We also noticed that searching more in this region gave far better results than searching the entire space. We have created three methods that move the BO's search to these better smaller regions.

3.1 Notations

The notations for n th evaluation are as follows:

$$p_n = f(\omega_n, a_n)$$

$$p_n^* = \max(p_0, p_1, \dots, p_n)$$

$$(\omega_n^*, a_n^*) = \arg \max f(\omega, a) | (\omega, a) \in \{(\omega_0, a_0), \dots, (\omega_n, a_n)\}$$

3.2 Crop

The first method starts with the complete search space and keeps moving the center of the search space and cropping it as better values for the probability are found. Let R_0 be the initial ratio to define the size of the domain. Then we calculate the new ratio R_{n+1} for $(n + 1)$ th evaluation as follows:

$$R_{n+1} = \max(0.01, (1 - p_n^*)^2) \times R_0$$

The center of search for $(n + 1)$ th evaluation is

$$\omega_{n+1}^c = \omega_n^*$$

Then the frequency search space for $(n + 1)$ th evaluation becomes:

$$\mathcal{F}_{n+1} = \{\omega | (1 - R)\omega_n^* \leq \omega \leq (1 + R)\omega_n^*\}.$$

This policy crops the search space by the squared difference between target probability 1 and p_t^* . The chopping of R is bounded by 0.01 to protect the exploration in the good sub-region. This particular formula and the particular value for the lower bound on chopping worked better for us than the other choices.

The Crop algorithm quickly prunes the search space making BO perform intensive exploitation in the selected region. This allows very fast optimization. Although effective, the Crop method assumes that the initially selected sub-region is optimal. These assumptions fail the cropping algorithm when other sub-regions are better than the one selected by it. Figure 3.1a gives a visualization of cropped search space as p reaches 0.81.

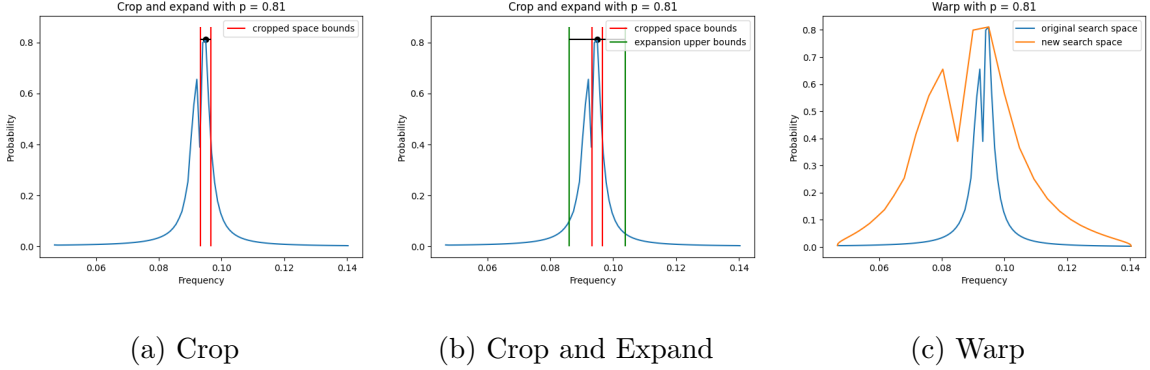


Figure 3.1: Visualization of proposed methods

3.3 Crop and Expand

We found that the Crop exploits too much and may miss even nearby optimal sub-regions. To increase its exploration we added expansion in it. For every evaluation, if the new evaluation is the best point, it will crop the search space (unchanged from Crop). Otherwise, it will keep on expanding. If $p_n^* > p_n$ the ratio for search space becomes

$$R_{n+1} = \min(R_0 \times (1 - p_n^*), R_{n-1} \times (1 + \eta(p_n^* - p_n))).$$

The maximum possible ratio for the search space at any point n is bounded by $R_0 \times (1 - p_n^*)$. The bounding was necessary as without bounding the method was resuming to the Standard BO. As the algorithm keeps finding better regions the upper bound keeps on contracting. The expansion is controlled by expansion rate $\eta \cdot (p_n^* - p_n)$ which ensures that the expansion is faster when the algorithm finds low probability points and is slower when it finds high probability points. We found $\eta = 0.2$ to be optimal for our problems. The Crop and Expand method gave better results than Crop but it needs another parameter η which is sensitive to the performance. Furthermore, the Crop and Expand method still carries the risk of

getting stuck in a local maxima due to the expansion upper bound. Figure 3.1b provides a visual representation of Crop and Expand as p reaches 0.81.

3.4 Warp

Crop and Crop and Expand both increase exploitation but may get trapped into a sub-optimal region. We need a method that can not only impose higher exploitation in the favorable region but also allow global exploration. The Warp method warps the search space into a new space where the favorable region has a larger area, thus encouraging further exploration here while not precluding exploration elsewhere. In the beginning, no warping is applied and it behaves like the Standard BO. As the algorithm starts finding better and better points the warping increases, stretching the search space near ω_n^* . The warping function maps each observation in the original parameter space to the new parameter space. The observations in the new space are then fed to BO and a new sampling point is obtained. This new sampling point is mapped back to the original space. The warping depends on the highest probability found till now, p_n^* . As p_n^* increases the warping becomes more intense. The frequency in original space ω is mapped to the frequency in the new space ω' as follows:

$$\omega' = \sigma \left(\frac{\sigma^{-1}(\omega) - \sigma^{-1}(\omega_n^*)}{1 - p_n^*} + \sigma^{-1}(\omega_n^*) \right)$$

Here *sigma* is the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

In the new mapping, the higher the p_n^* the greater the stretch near ω_n^* . The transformed data is then given to the Standard BO to generate a new sampling point. Since the space closer

to ω_t^* which is the good sub-region is stretched, the probability of sampling in it is increased. The new sampling point is again mapped in the original space and then evaluated. The method does prioritize a particular region over others but does not completely discard the rest which makes it a better explorer than the other two methods. Figure 3.1c represents the Warp method visually as p reaches 0.81.

Chapter 4

Chapter 4 Experiments and Results

4.1 Experiment Design

We have a total of 40,000 one-dimensional simulations and each of them can be used to perform transitions between eigenstates 0 to 1, 1 to 2, and 0 to 2. We picked 600 problems at random from these and tried to solve all three possible transitions. We are evaluating the methods for their speed and their generalization across a variety of problems. For speed, we are comparing the number of problems that achieved transition probabilities of at least 0.1, 0.5, and 0.9 for a given number of iterations of the optimization algorithm. For generality, we are comparing the distribution of the number of problems that achieved a transition probability falling into one of the following bins: below 0.25, between 0.25 and 0.5, between 0.5 and 0.75, and above 0.75.

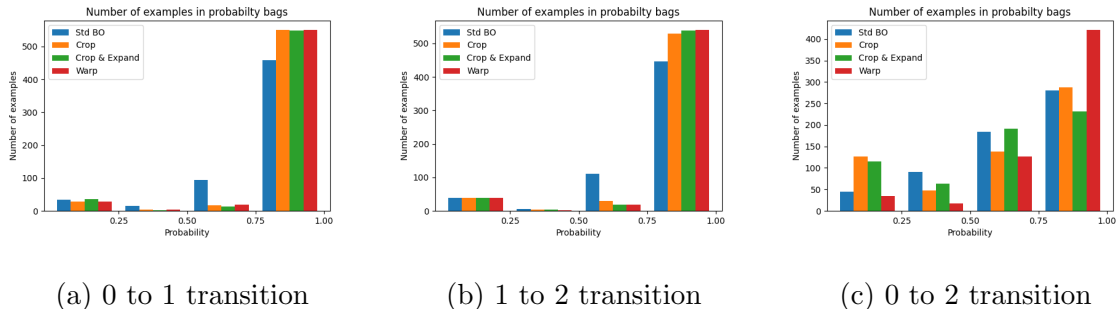


Figure 4.1: Generalization across problems

4.2 Results

4.2.1 Generalization of optimization

As visible in figure 4.1 the Warp method generalizes best. Generally, the 0 to 1 transition is easiest, the 1 to 2 transition is moderately difficult and the 0 to 2 transition is the most difficult. All three methods are able to surpass the Standard BO in 0 to 1. We suspect that most of these problems can be solved by greedily exploiting the first good sub-region found. This also explains why the performance of the Crop is good in this setting. For 1 to 2, the performance of the Crop and Standard BO are comparable whereas the performances of Crop and Expand and Warp are comparable and better than the other two. The reason might be that the best sub-region is in close proximity to other good regions, which is the ideal case for the Crop and Expand algorithm. For 0 to 2 transitions Warp outperforms all the other three methods, and the Crop has the worst performance of all. This type of transition is probably tricky because it may have the best sub-regions farther away from sub-optimal but still reasonable sub-regions. Even Crop and Expand is underperforming relative to Standard BO which reinforces this speculation. Since the Standard

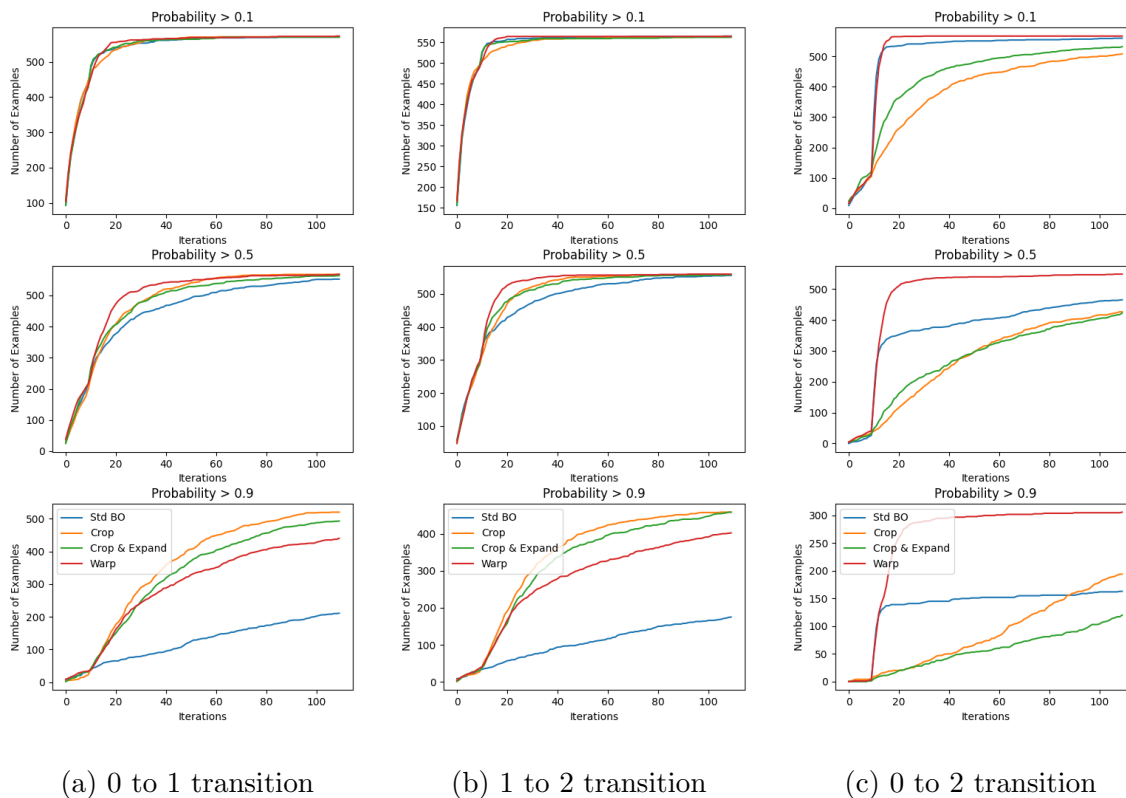


Figure 4.2: Speed across problems

BO is using the whole domain for its search, it is able to optimize this transition better than Crop and Crop and Expand. Warp is also using the whole search space to optimize, but it is imposing a softer exploitation in good sub-region while allowing exploration for distant sub-regions.

4.2.2 Speed of optimization

Out of the four methods, Warp was the most consistently fast across variety of problems. The problems that were difficult for generalization are slower for optimization. Their performances are visible in figure 4.2. For Probability greater than 0.1 the speed

for optimization is similar for the easier problems, 0 to 1 and 1 to 2, but the Standard BO and Warp are faster for the 0 to 2 transition. Similar to the generalization, there is a greater chance that the Crop and Crop & Expand get stuck into the local maxima which slow down their optimization. Due to the slight ability to explore we can see that Crop and Expand have faster optimization than Crop. For Probability greater than 0.5 the Standard BO has slower performance for easier problems 0 to 1 and 1 to 2, but it surpasses the Crop and Crop and Expand for 0 to 2. We believe that localizing search is useful in simpler problems and that is why exploit-oriented methods show better performance in them. But for 0 to 2, the Standard BO again surpasses the exploit-intense methods since these problems need more exploration. The method Warp, however, is the fastest even for this probability threshold which highlights its ability to better explore the search space. For probability greater than 0.9, we can see a similar notion of success but the differences are more significant here. Counterintuitively, The Crop is able to surpass the Standard BO for the hard 0 to 2 transition. The reason for that could be the better ability of the Crop to exploit relative to the standard BO. For those problems where Crop was able to reach the favorable region correctly, it was able to optimize those problems faster. Overall we can see that the Warp method is consistently fast.

Chapter 5

Conclusions

The search-space of the frequency of an EM signal for optimizing the quantum simulation of a photo-excited chemical system has optimal solutions in small sub-regions. Although Standard BO reaches these regions it still keeps exploring the rest of the space which slows down the optimization. The Standard BO method does not have any direct way to exploit the knowledge gained about good sub-regions in the space. The existing methods either focus on gradually expanding based on the estimates made from the smaller region or use the knowledge of bounds to better model the surrogate. In our case, evaluations of non-optimal points are not as useful to estimate the direction of expansion. The knowledge of an upper bound is also not helpful, as finding points near it is itself a challenge. Our methods provide a way to increase the exploitation in Standard BO. We propose three methods where each of which has a different exploitation and exploration policy. The method Crop has the most narrow exploration policy. For Crop the search space keeps decreasing and the exploration is provided by moving the center of the search space only. The method Crop

and Expand has a better exploration policy than Crop. It provides local exploration with both the bounded expansion of search space and the change of search space center. The third method, Warp, provides maximum exploration out of all of our proposed methods. This technique stretches the potentially optimal region to encourage BO but not require it to search there. This allows it to naturally increase BO's sampling in the desired area and hence increase the exploitation. Since it warps the search space instead of chopping, it always keeps the possibility of exploring anywhere in the search space. We found that the method Crop and Crop and Expand both exploit greedily. These methods are useful where sub-optimal regions are poor. The method Warp is most useful in the case where multiple sub-optimal regions are present and the algorithm needs to explore for a longer time. Overall when the difficulty is not known, out of our three methods, Warp is most likely to be successful.

Bibliography

- [1] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. 12 2010.
- [2] Peter I. Frazier. A tutorial on Bayesian optimization. 7 2018.
- [3] Huong Ha, Santu Rana, Sunil Gupta, Thanh Nguyen, Hung Tran-The, and Svetha Venkatesh. Bayesian optimization with unknown search space. *Advances in Neural Information Processing Systems*, 32, 2019.
- [4] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- [5] Vu Nguyen, Sunil Gupta, Santu Rane, Cheng Li, and Svetha Venkatesh. Bayesian optimization in weakly specified search space. *Proceedings - IEEE International Conference on Data Mining, ICDM, 2017-November*:347–356, 12 2017.
- [6] Vu Nguyen and Michael A. Osborne. Knowing the what but not the where in bayesian optimization. *37th International Conference on Machine Learning, ICML 2020, PartF168147-10:7273–7282*, 5 2019.
- [7] Andrei Paleyes, Mark Pullin, Maren Mahsereci, Cliff McCollum, Neil Lawrence, and Javier González. Emulation of physical processes with emukit. In *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS*, 2019.
- [8] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. *Gaussian Processes for Machine Learning*, 11 2005.
- [9] Akber Raza, Chengkuan Hong, Xian Wang, Anshuman Kumar, Christian R Shelton, and Bryan M Wong. NIC-CAGE: An open-source software package for predicting optimal control fields in photo-excited chemical systems. 7 2020.
- [10] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *IEEE Transactions on Information Theory*, 58:3250–3265, 12 2009.