Spring 2021

SPEEDING-UP A RANDOM SEARCH FOR THE GLOBAL MINIMUM OF A NON-CONVEX, NON-SMOOTH OBJECTIVE FUNCTION

Arnold C. Englander
University of New Hampshire, Durham

SPEEDING-UP A RANDOM SEARCH FOR THE GLOBAL MINIMUM OF A NON-CONVEX, NON-SMOOTH OBJECTIVE FUNCTION

By

Arnold C. Englander

M.S. in Engineering and Applied Science, Yale University, 1984

DISSERTATION

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of Doctor of Philosophy

in Electrical and Computer Engineering

May 2021

THESIS COMMITTEE PAGE

This thesis has been examined and approved in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering by:

Dr. Michael Carter, Thesis committee chair, Associate Professor *emeritus*, Electrical and Computer Engineering

Dr. John Gibson, Associate Professor, Mathematics and Statistics

Dr. Kyle Hughes, Aerospace Engineer, NASA Goddard Space Flight Center, Navigation and Mission Design Branch

Dr. Richard Messner, Associate Professor, Electrical and Computer Engineering

Dr. Se Young Yoon, Associate Professor, Electrical and Computer Engineering

On April 13, 2021

Approval signatures are on file with the University of New Hampshire Graduate School.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

being the first 48 of 2,500 trials, and with a Gamma distribution fit to the FPTs of 2,500 independent trials as their FPTD

IV.11: Textured prototypical *f* having a disconnected and sparse feasible domain, upon which MBH operated using a fixed non-Gaussian *p*, with $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 2500 trials, and with a Gamma distribution fit to the FPTs of 2,500 independent trials as their FPTD

IV.12: Textured prototypical *f* having a disconnected and sparse feasible domain, upon which MBH operated using an adaptive *p*, with $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 2500 trials, and with a Gamma distribution fit to the FPTs of 2,500 independent trials as their FPTD

IV.13: MBH operating on an *f* using a fixed Laplace(0,1) *p* that is not similar to $\hat{q}$

IV.14: MBH operating on the same *f* using a fixed p_λ (fixed in the sense that λ in $p_\lambda(0, \lambda)$ is time-invariant)

IV.15: MBH operating on the same *f* using $p_\lambda(0, \lambda[t])$ adapted to $\hat{q}[t]$, where $\lambda[t]$ was generated as described in the text

V.1: Gibsonian *f* upon which MBH operated using fixed non-Gaussian p and SCLS for which a maximum of 32 local steps per MBH time-step was allowed, with $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 1,000 trials, and a Gamma distribution fit to the FPTs of 1,000 independent trials as their FPTD

V.2: Gibsonian *f* upon which MBH operated using fixed non-Gaussian p and SCLS for which a maximum of 64 local steps per MBH time-step was allowed, with $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 1,000 trials, and a Gamma distribution fit to the FPTs of 1,000 independent trials as their FPTD

V.3: Gibsonian *f* upon which MBH operated using fixed non-Gaussian p and SCLS for which a maximum of 96 local steps per MBH time-step was allowed, with $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 1,000 trials, and a Gamma distribution fit to the FPTs of 1,000 independent trials as their FPTD

V.4: Textured prototypical *f* having a disconnected and sparse domain, upon which MBH operated using fixed non-Gaussian *p* and SCLS in which a maximum of 16 local steps per MBH time-step was allowed, with $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 1,000 trials, and a Gamma distribution fit to the FPTs of 1,000 independent trials as their FPTD

DEDICATION

This dissertation is dedicated to my family and my advisor, without whom it would have been impossible.

My children are my inspiration and pathfinders. My pursuit of a Ph.D. after they earned theirs is only one of many ways in which I follow in their footsteps. I can never thank them enough. My daughter Zoë's contributions to the present work are too encompassing to describe. My son Jacob's contributions are easy to describe. Jacob is my colleague, collaborator, and intellectual sparring partner. His Ph.D. research in spacecraft trajectory design and optimization introduced me to important applications of global optimization by random search that are very difficult. His experiences, challenges, hypotheses, and questions, over many years, motivated much of the present work. Jacob has also co-authored papers with me that raised questions addressed in the present work. In addition, Jacob provided me an unofficial role collaborating with the trajectory optimization teams at NASA Goddard Space Flight Center and introduced me to an international community of astronautical engineers, many of whom have contributed, directly or indirectly, to my work.

My wife, Joyce, is my *sine qua non*, my absolute essential without whom none of this would have been possible.

My advisor, Professor Michael J. Carter, in addition to his mentoring, deep knowledge and high intellectual standards, provided me with his exceptional patience and encouragement. Given my age, background, and personality, I am sure that I was a challenge for him many times in the past several years. I am very grateful.

ACKNOWLEDGEMENTS

The goal throughout this dissertation was to provide results that are both useful in practice and challenging analytically. The author is grateful to his advisor, Professor Michael Carter, and to colleagues at NASA's Goddard Space Flight Center, for their strong support for this double-sided goal.

I am also grateful to Professor John Gibson for posing a hypothetical, seemingly pathological, objective function for pedagogical purposes. After recognizing how often such objective functions arise in the real-world, how much they reveal about the MBH convergence process, and well MBH can be made to perform on them, the author chose to use them throughout the present work. In his honor, they are referred to as Gibsonian *f*.

With deep appreciation, I thank Paul Strehle for the steadfast encouragement and patience provided to me as a friend and business partner. Paul enabled me to pursue a Ph.D. while he and I managed a quantitative-based hedge fund. That not only provided the funding for my Ph.D. program; it also provided me with additional ways to think about stochastic processes.

GLOSSARY

$\aleph$	The most *f*-minimizing location $\boldsymbol{x}[t\text{-}1,m] \subset \boldsymbol{X}^{\mathbb{F}}$ of M simultaneous hoppers at MBH time-step *t-1*
ACT	Advanced Concepts Team within the European Space Agency
API	Application Programming Interface
CPU	Central Processing Unit
$\mathcal{C}$	An equivalence class of $\{(f_i,\boldsymbol{X}_i^{\mathbb{F}})\}$ that have similar local minima structure
d	Depth, in terms of descending slices through *f*, of *f*[**x**[t]] at MBH time-step *t*
$\acute{d}$	A particular *d* that is under consideration
dim(***X***)	The dimensionality of the maximum domain of *f*, namely ***X***
$d\boldsymbol{V}$	The infinitesimal volume for the space ***X***
DCLS	Deterministic Constrained Local Search
$D_{K\text{-}L}$	Kullback-Leibler divergence
$D_{K\text{-}L}(p, \hat{q})$	The Kullback-Leibler divergence measure of the similarity between probability mass functions *p* and $\hat{q}$
$\delta\boldsymbol{x}$	A small local random incremental movement in ***X*** which, in some versions of MBH, is used to try to improve $\boldsymbol{\xi}[t]$ in the sense of making $\boldsymbol{\xi}[t]$ more *f*-minimizing after forming $\boldsymbol{\xi}[t]$ = (**x**[t-1] + $\Delta\boldsymbol{x}$)
$\lvert\Delta v\rvert$	Change in velocity of a spacecraft achieved by the consumption of on-board propellant
$\Delta\boldsymbol{x}$	A random incremental movement in ***X*** used to form $\boldsymbol{\xi}[t]$ = (**x**[t-1] + $\Delta\boldsymbol{x}$)
$\mathbb{E}(\eta[\tau_a : \tau_b])$	The expected efficiency η of an MBH search over the interval of MBH time-steps from τ_a to τ_b

ESA	European Space Agency
η	A measure of the efficiency of an MBH search as $f[\boldsymbol{x}[t_a]] \rightarrow f[\boldsymbol{x}[t_b]]$ including as $f[\boldsymbol{x}[0]] \rightarrow f^*$
FPT	First Passage Time
FPTD	First Passage Time Probability Density Function
FPT-PMF	First Passage Time Probability Mass Function
f	Objective function
$f^{-1}[d]$	The set inverse of f at depth d
f^*	The global minimum of objective function f, namely $\min_{\text{global}}(f(\boldsymbol{X}))$
$f[\boldsymbol{x}^{\#}]$	The value of f at local a local minimum located at $\boldsymbol{x}^{\#}$
$\phi_j(\boldsymbol{X})$	The j^{th} penalty function
$\Gamma(\alpha, \theta)$	The Gamma distribution having parameters α and θ fit to MBH FPTs
$g[d]$	The remaining volume of productive search space when $f[\boldsymbol{x}] = d$
GSFC	Goddard Space Flight Center within NASA
i.i.d.	Independent and identically distributed random variables
$I_{f,d}(\boldsymbol{x})$	An indicator function used to evaluate $f[d]$ at x
$I_{\boldsymbol{x}}[t]$	An indicator function used to evaluate $\boldsymbol{\xi}$ as a function of $\boldsymbol{x}$ at time-step t
len($\boldsymbol{X}$)	The span of the longest dimension of the domain of f, namely $\boldsymbol{X}$
LJ	Luus–Jaakola algorithm
λ	The scale parameter of $q_{\boldsymbol{\Delta x}}$
$\hat{\lambda}[t]$	The time-varying scale parameter of $\hat{q}[t]$
MBH	Monotonic Basin Hopping
MCH	Multiple Communicating Hoppers
MCM	Multiple Communicating MBH processes

$\mu(\boldsymbol{X}^{\mathbb{F}})$	A measure of the volume of $\boldsymbol{X}^{\mathbb{F}}$
NASA	The U. S. National Aeronautical and Space Administration
OR	Operations Research
PS	Pattern Search
PyKep	A planetary orbital model designed by ACT ESA
$p_{\boldsymbol{\delta x}}$	The probability distribution from which the $\boldsymbol{\delta x}$ are drawn
$p_{\boldsymbol{\Delta x}}$	The distribution from which the $\boldsymbol{\Delta x}$ are drawn
$q_{\boldsymbol{\Delta x}}$	The distribution comprised of $\boldsymbol{\Delta x}$ such that $\boldsymbol{\xi}[t]$ = ($\boldsymbol{x}$[t-1] + $\boldsymbol{\Delta x}$) is accepted as the new incumbent $\boldsymbol{x}[t]$
$\mathbb{R}^N$	The N-dimensional space of real numbers
SA	Simulated Annealing
SNOPT	Sparse Nonlinear OPTimizer, a Non-linear Programing solver
t	the index of an MBH iteration, referred to as an MBH time-step
$\boldsymbol{U}_f$[d]	$\boldsymbol{U}_f$[d] = f^{-1}[d], where f^{-1} is the set of points comprising the inverse of f
$\widetilde{\boldsymbol{U}}_f[d]$	Those $\boldsymbol{x} \in \boldsymbol{X}^{\mathbb{F}}$ such that $\boldsymbol{U}_f[\acute{d}] \subset \boldsymbol{U}_f$[d] for all $\acute{d} < d$ (in the definition of g[d])
$\boldsymbol{X}$	The domain of f, typically N-dimensional
$\boldsymbol{X}^{\mathbb{F}}$	The domain of feasible f, possibly disconnected and sparse
$\boldsymbol{x}^*$	The location (argmin) of the global minimum of objective function f, *namely* $\text{argmin}_{\text{global}}(f)$
$\boldsymbol{x}^{\#}$	The location of a local minimum that is not the global minimum
$\boldsymbol{x}[t]$	The current incumbent location in $X^{\mathbb{F}}$of the most f-minimizing $\boldsymbol{x}$
$\boldsymbol{\xi}[t]$	The current candidate location for the most f-minimizing location in $\boldsymbol{X}^{\mathbb{F}}$
$\mathbb{Z}^{\mathrm{N}}$	The N-dimensional space of integers

ABSTRACT

The need to find the global minimum of a highly non-convex, non-smooth objective function over a high-dimensional and possibly disconnected, feasible domain, within a practical amount of computing time, arises in many fields. Such objective functions and/or feasible domains are so poorly-behaved that gradient-based optimization methods are useful only locally – if at all. Random search methods offer a viable alternative, but their convergence properties are not well-studied. The present work adapts a proof by Baba *et al.* (1977) to establish asymptotic convergence for Monotonic Basin Hopping, a random search method used in molecular modeling and interplanetary spacecraft trajectory optimization. In addition, the present work uses the framework of First Passage Times (the time required for the first arrival to within a very small distance of the global minimum) and Gamma distribution approximations to First Passage Time Densities, to study MBH convergence speed. The present work then provides analytically supported methods for speeding up Monotonic Basin Hopping. The speed-up methods are novel, complementary, and can be used separately or in combination. Their effectiveness is shown to be dramatic in the case of MBH operating on different highly non-convex, non-smooth objective functions and complicated feasible domains. In addition, explanations are provided as to why some speed-up methods are very effective on some highly non-convex, non-smooth objective functions having complicated feasible domains, but other methods are relatively ineffective. The present work is the first systematic study of the MBH convergence process and methods for speeding it up, as opposed to applications of MBH.

INTRODUCTION

Finding the value and location of the global minimum of a non-smooth and/or non-convex objective function over a high-dimensional and possibly disconnected and sparse feasible domain, is a requirement in many fields of science and engineering. Often, the objective function and/or feasible domains are so poorly-behaved that conventional gradient-based optimization methods are useful only in very small local neighborhoods – if at all. Therefore, global optimization methods based on random search are used. Such optimization problems involving non-smooth and/or non-convex objective functions over high-dimensional spaces are common in the design of trajectories for interplanetary spacecraft missions, molecular modeling, and the modeling of fitness landscapes in evolution [1, 2, 3]. The extent to which non-smooth and/or non-convex objective functions also exist in Deep Learning is an active area of research in the Machine Learning community [4, 5].

One random search-based method of global optimization that has been widely applied to spacecraft design and molecular modeling is Monotonic Basin Hopping (MBH). However, until the present work, MBH had not been supported by any analytical foundation. The present work departs from the historical literature of MBH by focusing on the analysis and speed-up of the MBH process itself, rather than on the application problem to which it is applied.

The first literature addressing the application of MBH in molecular modeling goes back to 1997. The first literature addressing the application of MBH in spacecraft trajectory design goes back to 2010. Until the present work (and papers co-authored by the present author), any analysis in the MBH literature focused on the physics that generates the objective function. From that,

intuition-based hypotheses were offered to explain why MBH "should" be effective on such objective functions.

The present work extends papers, co-authored by the present author, on factors that impact the speed of MBH convergence. The first such paper, published by Englander and Englander in 2014, addressed the impact of the choice of the probability distribution from which MBH hops are drawn [6]. That paper provided empirical findings unsupported by any analytical framework. The first paper to provide an analytical framework for the 2014 paper was published by Englander, Englander, and Carter in 2020 [7].

That 2020 paper adapted to MBH an asymptotic convergence proof for global optimization by random search, first published in 1977, and showed its applicability to MBH under sufficient conditions . At the time, it was believed that MBH was not guaranteed to eventually converge but that did not worry most MBH practitioners because they were using MBH to solve optimization problems that were widely found to be otherwise unsolvable. Further, from a practical point-of-view, a guarantee of asymptotic convergence is unimportant because if an MBH requires more time to converge than is allowed in real-world engineering settings, that is essentially the same as a failure of the MBH to converge. The idea that MBHs are guaranteed to converge – and would in practical amounts of time if they could be sufficiently sped-up – had not yet been recognized. Even if it had been, because the MBH convergence process was not yet understood, there was no systematic, let alone analytically supportable, method for speeding it up.

It was observed years ago that while MBH convergence times seemed unpredictable, they took longer to the extent that the objective function was non-convex and/or non-smooth, and/or the subspaces over which f was feasible was high-dimensional and/or disconnected and sparse. Yet, most spacecraft trajectory optimization problems are very much like that, as are many

molecular modeling problems that are posed as optimization problems involving energy landscapes.

It was also observed years ago that MBH convergence was slower when f was non-smooth, for example sharply textured, in its feasible domain but that too is characteristic of objective functions in many real problems to which MBH is applied because of the physics involved and/or because of numerical effects. The presence of sharp texture is also interesting because it can severely limit the effectiveness, or at least the efficiency, of gradient based methods that one might use instead of, or with, MBH.

Because of the need to perform such optimizations within practical project deadlines, MBH was originally developed heuristically, often with a naïve understanding of the factors that determined MBH convergence time.

Two sets of methods for speeding-up MBH convergence are provided and supported analytically in the present work: The first set of methods involves biasing the shape of the probability distribution of random hop distances, and the second set of methods involves biasing the location from which next hops are taken. Both sets of methods and their analysis are novel, although they build upon empirically-based and applications-oriented discussions between the present author and Jacob Englander during the past decade.

Chapters IV and V are complementary. Chapter IV, regarding biasing the hop length distribution, addresses speeding-up MBH by increasing the efficiency of the search across the feasible domain for the location of the global minimum. Chapter V, regarding biasing the location from which the next hop is taken, addresses speeding-up MBH by increasing the efficiency of descent of the objective function at the hopper's current location to the global minimum.

The present work uses First Passage Times (FPTs) and Gamma distribution approximation of First Passage Time Density functions (FPTDs) to characterize the speed of MBH convergence, how that speed is impacted by *f* and/or the geometry of its feasible domain, the distribution from which incremental hop distances are drawn, and the extent to which methods for speeding up convergence are effective. The present work defines an MBH FPT as the number of MBH time-steps required for an MBH to travel from its starting location to its first arrival at the position of the global minimum (or within a defined small epsilon ball around it). Although the location and value of the global minimum are unknown in real-world applications of MBH, FPTs can be observed in simulated or hypothetical problems in which the global minimum and first passages are known by construction.

The probability density of FPTs, approximated by the fitting of a Gamma distribution to FPTs generated by a large number independent trials of the same MBH operating on the same objective function and feasible domain, driven by random increments drawn from the same distribution, and using the same (or no) method of speed-up, is the FPTD for that set of FPTs. FPTDs are an essential part of the present work because they reveal, and enable the comparison of, important factors that impact MBH FPTs and therefore MBH convergence speed. First Passage Time Probability Mass Functions (FPT-PMFs) can used instead of FPTDs but FPTDs are more time-efficient because far fewer FPTs are needed to construct a well-fit Gamma distribution as a FPTD than are needed to construct a high-fidelity histogram. In addition, PMFs are reserved for use in Chapter V as part of a method for measuring the similarity between two probability distributions that are un-related to FPTs. The empirical justification for using a Gamma distribution is provided in Appendix B. A hypothesis for an analytical justification for using a Gamma distribution is provided as area for future work in Chapter VII.

MBH FPTDs are also used to characterize the benefits of designing or adapting the distribution from which incremental hop distances are drawn in a manner that makes that distribution similar to the distribution of incremental hop distances that move the MBH toward the global minimum. In that context, measuring the similarity between the two distributions is important. In the present work, the tool used for measuring the similarity between two probability distributions is the Kullback-Leibler divergence ($D_{K\text{-}L}$) which was originally developed in the literature of Information Theory, has been used in many statistical applications, and has been adopted by the Machine Learning Community. The inputs to $D_{K\text{-}L}$ are specially constructed $D_{K\text{-}L}$-compliant PMFs (histograms) that should not be confused with the Gamma-fit FPTDs of MBH FPTs.

The use of FPTs and Gamma distributions fit as FPTDs of FPTs, as a way to investigate MBH convergence rates and methods for speeding-up MBH, is novel. Likewise, the use of $D_{K\text{-}L}$ in Chapter IV, to explain the effect and benefit of adapting the shape of the distribution from which hop increments are drawn, is novel.

Novel contributions to engineering practice of the present work includes analytically supported methods for speeding-up MBH. One method is provided in Chapter IV and two methods are provided in Chapter V. Their combined use is illustrated in Chapter VI. The key thought for practitioners is that MBH inevitably converges to the global minimum; the only question is how long that convergence will take. What practitioners consider failure to converge is simply failure to a converge within the calendar time allowed by a project schedule (e.g., failure to converge by next Monday). The present work provides methods that speed-up MBH on many realistic objective functions and geometries of feasible subspaces, thereby enabling MBH to “succeed” when applied

to challenging optimization problems that arise in molecular biology and spacecraft trajectory optimization and practitioners are given relatively tight project schedules.

The following Figures 0.a, 0.b and 0.c below are prototypical 1-dimensional examples of an objective function and its feasible domain. These, among others, will be used as test cases, through the present work. The Pioneer 11 spacecraft trajectory optimization use-case in Chapter VI uses use a 3-dimensional poorly-behaved objective function with disconnected, sparse feasible subspaces.

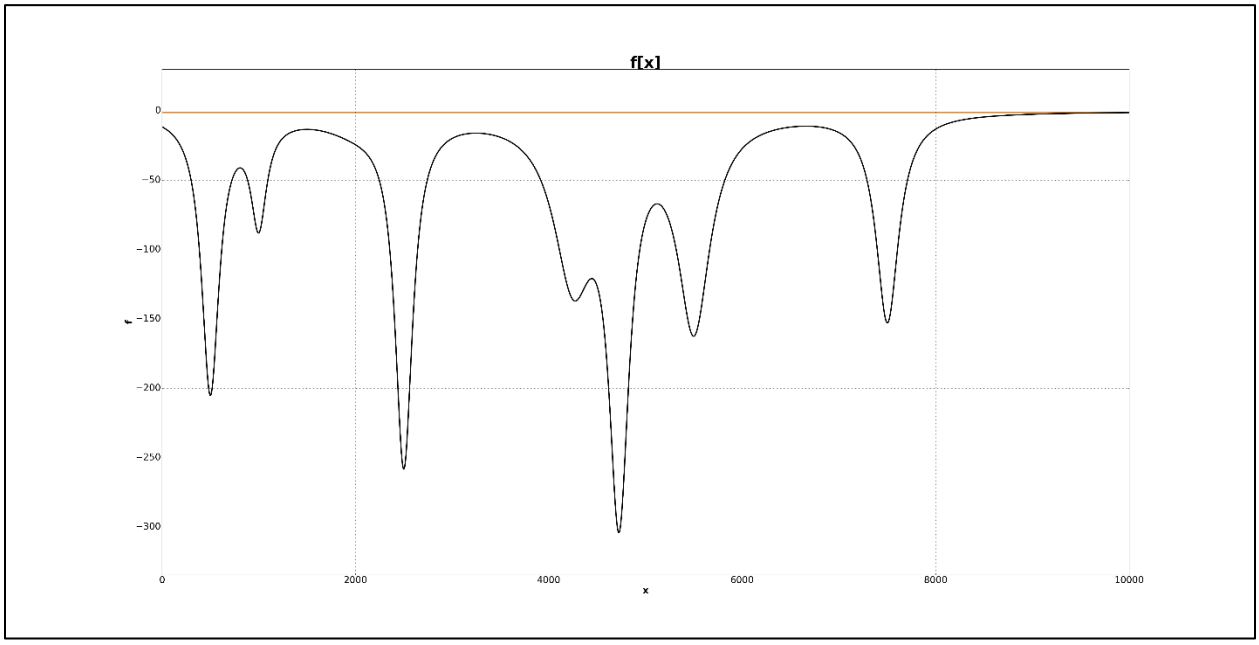


Figure 0.a: Prototypical 1-dimensional non-convex objective function *f* having a feasible domain that is the entirety of the domain. The light brown horizontal line passes through maximum feasible *f*.

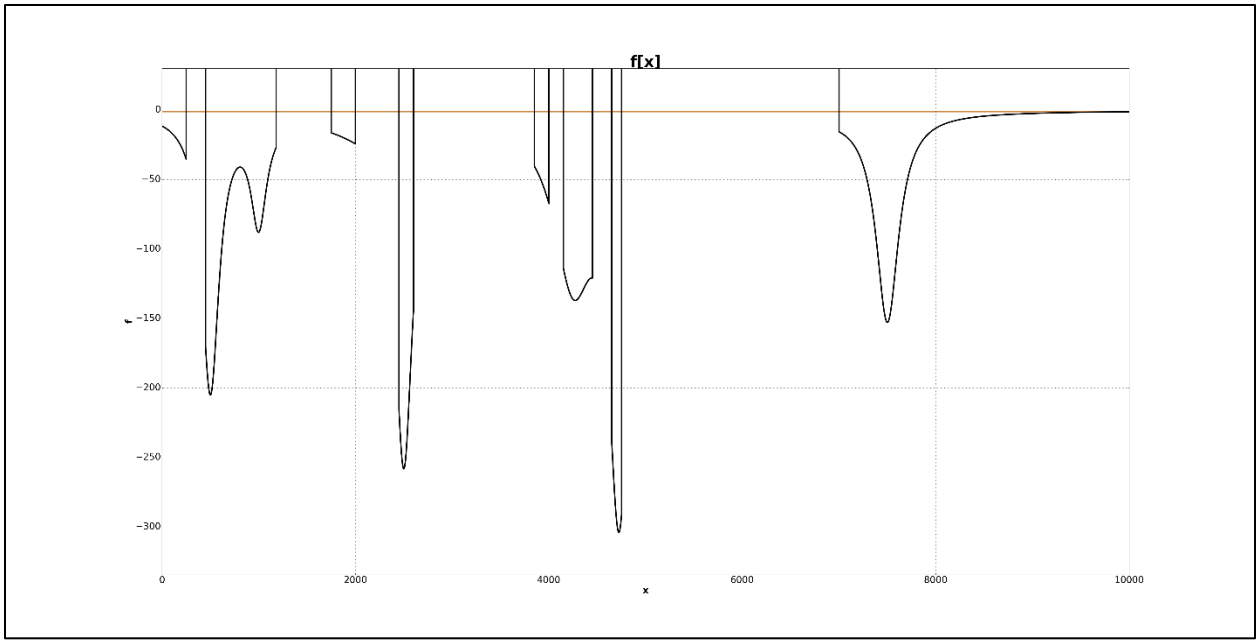


Figure 0.b: Prototypical 1-dimensional non-convex objective function *f* having disconnected, sparse feasible sub-domains. The light brown horizontal line passes through maximum feasible *f*.

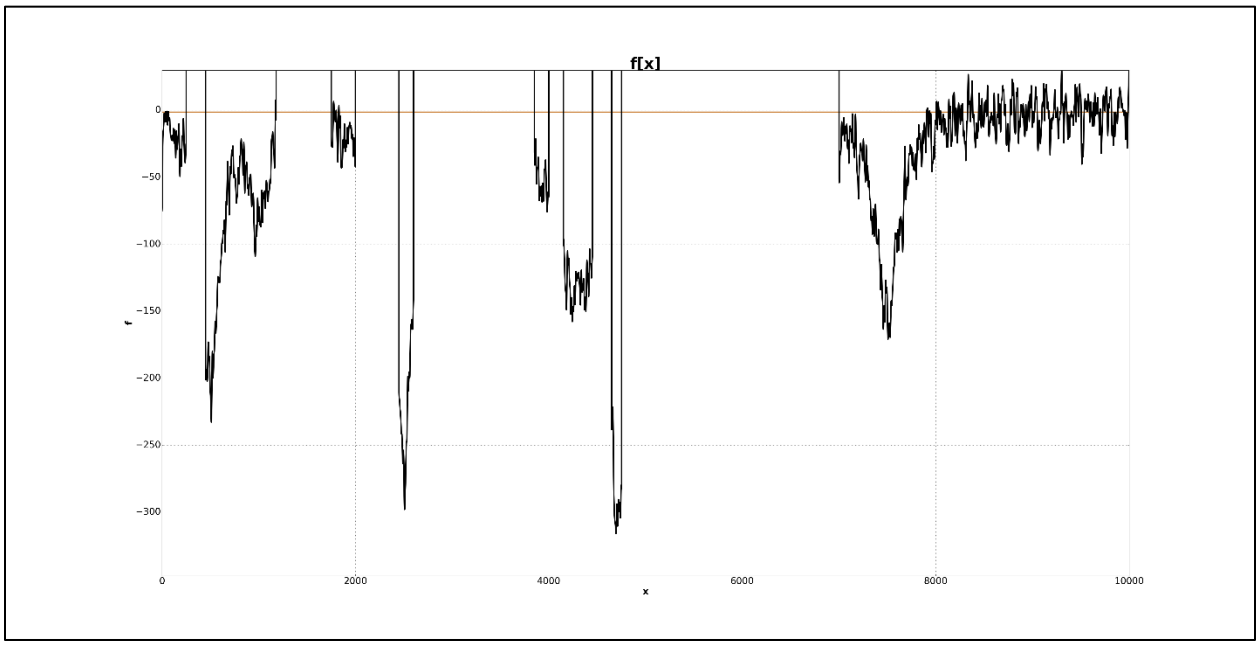


Figure 0.c: Textured prototypical 1-dimensional non-convex *f* having disconnected, sparse feasible sub-domains. The light brown horizontal line passes through maximum feasible *f*.

I. PRIOR WORK

I.a. Prior work on MBH

Other than papers co-authored by the present author, almost all of the prior literature on MBH addresses the applicability and applications of MBH rather than the convergence of MBH and how to speed it up. The exceptions are a few papers that provide improved methods for so-called local gradient search, referred to in Chapter V as Deterministic Constrained Local Search (DCLS) [8, 9]. However, those papers focus on deterministic properties of known constraints on feasible subspaces and do not address MBH being a form of random global search.

Wales and Doye provided one of the earliest papers on MBH [10]. They used a form of MBH to find the lowest energy structures of Lennard-Jones clusters (a macro-molecular system) containing up to 110 atoms. Their success was a landmark achievement in global non-convex optimization. They hypothesized that the effectiveness of MBH on their application problem was due to features they believed were present in the energy landscapes of Lennard-Jones clusters: global non-convexity combined with local convexity in very small neighborhoods. For them, that was a prescription for MBH combining what they called local gradient search with global random search. They were apparently unaware that Baba *et al*. (1977) had already proven, as shown in Appendix A, that local search was already inherent in random search if particular conditions are met [11]. The present work addresses the incorporation of local gradient search into global random search in Chapter V. There, the present author distinguishes between Deterministic Constrained Local Search (DCLS) and Stochastic Constrained Local Search (SCLS). Wales and Doye, as well as their followers up to the present work, used what the present author refers to as DCLS. In so doing they launched an approach that becomes ineffective when the assumed “local convexity” is

sharply textured. The application by Wales and Doye of MBH to energy landscapes that are globally non-convex but locally convex, in particular Lennard-Jones atomic clusters, represents the start of MBH literature. Therefore, local gradient search in the sense of DCLS was considered part of MBH from the beginning. Nonetheless, because of the possible non-smooth objective functions being considered in the present work, DCLS was purposely not used.

In 2000, Leary discussed a form of MBH in his work on global optimization on funneling landscapes in molecular modeling in general [12]. Leary regarded MBH as a method for finding a manageable set of very deep local minima from which the global minimum could later be chosen. Apparently, he did not investigate whether or how MBH could be used to find the one global minimum directly.

Like Wales and Doye, Locatelli and Schoen applied MBH to Lennard-Jones clusters [13]. Like Leary, they regarded MBH as being a method for finding a set of deep local minima which they referred to as a set of "putative" global minima in the domain being searched. They reported dramatic results for the performance of their version of MBH applied to the most difficult Lennard-Jones clusters being investigated at their time: improvements "… by two orders of magnitude … in finding the global optima of clusters of 75, 98 and 102 atoms". They carried on the earlier incorporation of "local gradient search" into MBH, originated by Wales and Doye, without questioning it. They continued the application-focus of MBH literature and did not analyze the MBH convergence process.

Vasile, Minisci, and Locatelli provided a detailed comparison of global optimization methods applicable to spacecraft trajectory design, including MBH [14]. Their paper was a contribution to the optimization of planetary "transfers" modeled as Lambert's problem in orbital mechanics. They used the physics of planetary transfers to show that the optimization problems

were globally non-convex but, in small neighborhoods, locally convex. On that basis, they argued that the optimization of spacecraft trajectories would benefit from methods like MBH. Their argument was similar to that of Wales and Doye, Leary, and Locatelli and Schoen in the context of molecular modeling problems. Locatelli had recently transitioned from molecular modeling to spacecraft optimization and, in the process, brought with him the belief that so-called local gradient search was a necessary complement to the random global search aspects of MBH. Like MBH researchers before them, Vasile, Minisci, and Locatelli did not investigate whether or to what extent MBH might work without local gradient search. Nor did they seek to analyze whether and why "local gradient search" speeds-up the convergence of MBH to the global minimum, as is done in Chapter V of the present work. They were focused on using local gradient search to speed-up finding local minima that were candidates for being the "putative global minimum".

In 2011, Yam, Lorenzo, and Izzo published the first solution for the semi-autonomous design and global optimization of low-thrust (ion propelled) spacecraft missions. Semi-autonomous design and optimization means that the selection of targets of "fly-by" gravity assists (e.g., planets and moons) were performed by humans, whereas the transfer trajectories between "fly-bys" were optimized by machine [15]. Yam, Lorenzo, and Izzo made important contributions to applications of MBH, including ways of measuring, documenting, and displaying the performance of empirical results. However, they did not provide any analysis of the convergence of MBH or address how to speed-up its convergence. They did further entrench "local gradient search" in the MBH recipe.

In 2012, Olson, Hashmi, Molloy, and Shehu wrote about "basin hopping as a general and versatile optimization framework for the characterization of biological macromolecules" [16]. They extended the lineage of Wales and Doye, and Locatelli and Schoen, to protein structure

prediction and molecular docking, and made important contributions to the method and applications of basin hopping. They did not question whether or why “local gradient search” was needed, presumably because by then it was so entrenched in the literature. They did not provide any analysis of the convergence of MBH or methods for speeding-up MBH.

Englander, J., and Conway published the first solution for the fully autonomous (totally machine-generated) design and global optimization of low-thrust spacecraft missions [17]. The solution was based on using MBH in an inner optimization loop to optimize trajectories given choices of “fly-by” targets, and an outer optimization loop that used a Genetic Algorithm (GA) to choose “fly-by” targets. Earlier, Englander, Conway, and Williams published the first solution to fully autonomous design and global optimization of missions using chemical combustion propulsion [18]. Objective functions for chemical combustion propelled missions are simpler than objective functions for low-thrust ion propelled missions.

In 2014, Jacob Englander and the present author published the first empirical evidence that, for MBH operating on real and high-dimensional spacecraft trajectory optimization problems as well as synthetic low-dimensional optimization problems, convergence times are significantly impacted by the probability distribution from which the random search increments (“hop” distances) are drawn [6]. Furthermore, it was shown that for a wide variety of objective functions, specific probability distributions sped-up MBH convergence much more than others. While the empirical results described in that paper were sufficiently compelling to be used to improve the MBH-based trajectory optimization software at NASA Goddard Space Flight Center, the present author was dissatisfied by the lack of analytical support for those empirical results. The then-missing analytical support is provided in Chapter IV and Appendix C of the present work. In that 2014 paper, the empirical results provided by Jacob Englander were generated by an MBH that

incorporated a form of constrained local gradient search referred to in the present work as DCLS, whereas the empirical results provided by the present author were generated by MBH that used random search alone. Because the contribution by Jacob Englander involved a higher-dimensional, and real rather than synthetic problem, and the paper focused on the probability distribution from which the random search increments ("hop" distances) are drawn, the role and necessity of local gradient search was not investigated. However, at that point the present author not only committed himself to an analytical investigation of the effects of the hop length distribution, but also to investigating the previously then unquestioned role of local gradient search in MBH. Chapter IV of the present work purposely does not use or address the use of local gradient search in MBH. Chapter V addresses two versions of constrained local gradient search, referring to them as Deterministic Constrained Local Search (DCLS) and Stochastic Constrained Local Search (SCLS), and provides an analytical explanation of their benefits and liabilities. In Chapter VI, lessons from Chapter IV and parts of Chapter V are applied to the re-design and re-optimization of the historic Pioneer 11 spacecraft trajectory. However, no form of local gradient search is used, if only to demonstrate that it was not needed in at least that real spacecraft trajectory optimization use-case. Despite not using any form of local gradient search in the use-case in Chapter VI, the benefit of SCLS is illustrated at the end of Chapter V using simulation-based evidence from an MBH operating on a 1-D Gibsonian objective function after an explanation for SCLS is provided.

In late 2017, the present author developed the idea of speeding-up MBH by using Multiple Communicating Hoppers (MCH) as described in Chapter V. At the time, the idea was based on intuition confirmed by simulation experiments. The resulting performance benefits on a wide variety of objective functions were so dramatic that the present author discussed them with Jacob Englander and Kyle Hughes at NASA's Goddard Space Flight Center (Mission Design

and Guidance Branch) in early 2018. Englander, Englander, and Hughes then proposed an internal research project within Goddard to investigate the method's applicability to especially challenging spacecraft trajectory optimization problems [19]. The internal project was funded, and applications within Goddard have since been developed as have conceptually related methods such as Multiple Communicating MBHs (MCM) [20]. Meanwhile, an analytic explanation of MCH and its benefits remained missing until the present work.

In 2019, Izzo and the Advanced Concepts Team that he manages at the European Space Agency published a set of Python modules named PyKep for solving Kepler's models for celestial mechanics using Lagrange's solutions to Lambert's problem [21]. A few years earlier, they had published models of historical spacecraft missions that could be designed and optimized by casting their objective functions in a form generated by previous versions of PyKep. The combination of PyKep and PyKep-compatible data-sets for modeling historical spacecraft missions have been used in the present work to benchmark use-case applications based on the analytical framework described in this paper.

I.b. Prior literature on random search for the global extremum

In 1977, Baba, Shoman, and Sawargi, working in Operations Research (OR), provided a general asymptotic convergence proof for stochastic global optimization under specified sufficient conditions [11]. An adaptation of their asymptotic convergence proof to MBH is contained in Appendix A of the present work**.** In 1981, Solis and Wets, also working the field of OR, re-stated the proof by Baba *et al.* and provided additional examples of it [22]. Neither the 1977 paper by Baba, Shoman, and Sawargi, or the 1981 paper by Solis and Wets, limited the dimensionality of the domain of the objective function except that it be finite. The asymptotic convergence proof for

stochastic global optimization by Baba *et al.* is summarized in the 2003 lecture slides by Aghassi [23].

I.b. Prior literature on other approaches to the random search for the global extremum

The literature of random global search contains approaches other than MBH. Some of these branches involve algorithms that are similar to MBH, but they appear to have been developed without an awareness of MBH, and MBH seems to have developed without an awareness of them. Like MBH, these other approaches were first developed as heuristics to solve specific difficult optimization problems. Unlike MBH, some of these branches evolved in ways in which mathematically talented contributors eventually provided convergence proofs and some level of analysis. Three of the approaches that have been given strong analytical foundations include: the Luus–Jaakola (LJ) algorithm; Pattern Search (PS); and Simulated Annealing (SA).

LJ is interesting in the context of the present work because of its close resemblance to MBH even though MBH is not mentioned in the LJ literature, nor does the literature of MBH mention LJ. Like the literature of MBH (until the present work), the literature of LJ does not appear to refer to the work of Baba *et al.* (1977).

The LJ algorithm is often referred to as a heuristic for stochastic global optimization but given the formalism brought to it later, it more than simply a heuristic. It was first proposed by computer scientists Luus and Jaakola in 1973 and further developed by Luus thereafter [24, 25, 26, 27]. The LJ algorithm uses a fixed uniform distribution for the increments that drive the random search, and does not use, presuppose, or therefore require, gradients.

In 1979, Nair and Gopalakrishnan proved asymptotic convergence for LJ in the restricted case in which f is twice continuously differentiable [28]. Nair and Gopalakrishnan also proved that the worst-case complexity of minimization on the class of uni-modal objective functions

grows exponentially in the dimension of the problem, according to the analysis of Yudin and Nemirovsky [29]. Further, they point out that the Yudin-Nemirovsky analysis implies that no method can be "fast" on high-dimensional problems that lack convexity. Of course, that raises questions about the definition of "fast". Simulation experiments for MBH, undertaken as part of the present work, suggest that the relationship between dimensionality and convergence times is much more complicated and rarely as onerous as Nair and Gopalakrishnan hypothesized it was for PS.

In any case, with respect to the present work, the results and hypotheses of Nair and Gopalakrishnan, and Yudin and Nemirovsky, are tangential because the focus here is speeding up the convergence of MBH, specifically when p is chosen, designed, or adapted to q, rather than when p is a single fixed probability distribution used on all f. Moreover, the present work is focused on multi-modal f.

The present work shows that (MBH) convergence time depends upon the geometry (hyper-geometry) of multi-modal f as much or more than the dimensionality of f when p is designed or adapted to be similar to q. But the point here is not to agree with or dispute Nair and Gopalakrishnan, but simply to point out that in parallel with the development MBH another branch of stochastic global optimization, namely LJ, was developed. Further, neither LJ nor MBH appear to have referred to, or learned from, each other. In addition, LJ assumed, as did MBH at the time, that p was fixed for all f and for the duration of optimization process, rather than p being designed or adapted so as to be especially well-suited to f. There was no research into whether, by making p well-suited to f, convergence could be sped-up and/or terminal accuracy could be improved.

PS is related to JL in the sense that, like JL, PS does use or require gradients. The literature of PS includes a family of algorithms that – because they do not require gradients – are useful

when measurements of the objective function are noisy. PS is considered by some to be a variant of an algorithm attributed to Fermi and Metropolis when they worked at Los Alamos National Laboratory, as described by Davidon [30]. If one were to ask when PS algorithms became identifiably different from the earlier work by Fermi and Metropolis, it appears that would be the in the work of Hooke and Jeeves in 1961 [31]. In 1997 Torczon, and in 2003 Dolan, Lewis, and Torczon, provided convergence proofs for special cases of PS algorithms operating on special cases of objective functions [32, 33].

The literature of PS does refer to the literature of JL, but not to the literature of MBH. Nor did the literature of MBH refer to the literature of JL or PS until the present work. As in the literature of JL, PS assumes a fixed p for all f, and does not appear to address a p that is well suited to f by design or adaptation.

Another descendent of the work at Los Alamos attributed to Fermi and Metropolis is SA. The development of SA is attributed to Pincus in 1970, and to Kirkpatrick *et al.* in 1983 [34, 35]. SA is generally thought of as an adaptation of a so-called Monte Carlo sampling method, called the Metropolis-Hastings algorithm, used to generate simulated sample states of a thermodynamic system. The Metropolis-Hastings algorithm is attributed to Metropolis *et al.* in 1953, and to Hastings who published it as a Monte Carlo sampling method explainable by Markov chains in 1970 [36, 37, 38]. Excellent overviews of SA are provided by Schneider and Kirkpatrick, and by Spall [39, 40].

SA is a rich subject that is steeped in analogies to physics, especially statistical thermodynamics. It has an extensive literature that is outside the scope of this paper. However, a few observations about SA are warranted here: While the SA literature does include an asymptotic convergence proof, as well as some level of analysis for special cases of objective functions, it too

lacks any analytically supportable notion of a p that is well suited to f by design or adaptation. Typically, SA literature presupposes that p is a Boltzmann distribution that has a time-varying parameter that is varied by a schedule rather than by properties of f or the statistical dynamics of the on-going optimization process. Therefore, while SA is related in some respects to MBH, it is tangential to the central theme of the present work.

The literature of SA does not appear to refer to MBH, JL or PS, just as the literature of MBH (with the exception of the Englander, Englander 2014 paper which refers to SA, and the present work) refer to JL, PS or SA. Nonetheless, all these branches of stochastic global search are related.

Missing from all of these branches of stochastic global search, including MBH until now, were the concepts and tools of the analytical framework used in the present work.

II. MONOTONIC BASIN HOPPING (MBH)

II.a Definition of MBH and the specification of the basic MBH algorithm

MBH is an iterative method for randomly searching for the global minimum of an objective function f over domain $\boldsymbol{X}$. The global minimum is written as $\min_{\text{global}}(f(\boldsymbol{X})) = f^*$. $\boldsymbol{X}$ is a Cartesian product of bounded subspaces in $\mathbb{R}^N$. Values of f are only of interest for $\boldsymbol{X}^{\mathbb{F}} \subset \boldsymbol{X}$, a set of possibly disconnected, sparse subspaces in $\boldsymbol{X}$. In practice, $\boldsymbol{X}$ is non-uniformly discretized with a granularity of IEEE floating point precision which some practitioners regard as approximately continuous. For other practitioners, the granularity of IEEE floating point precision introduces a kind of "noise" with which they have learned to contend.

In the present work, the objective function f is non-convex and/or non-smooth and, if only for those reasons, is referred to as being "poorly-behaved". The definition of convex is:

$$f \text{ convex} \Leftrightarrow 0 \leq \alpha \leq 1\,,\ f(\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2) \leq \alpha f(\boldsymbol{x}_1) + (1-\alpha) f(\boldsymbol{x}_2)$$

$$\text{for all } \boldsymbol{x_1}, \boldsymbol{x}_2 \in \boldsymbol{X} \text{ such that } \boldsymbol{x}_1 \neq \boldsymbol{x}_2$$

The definition of non-smooth is that the first derivative or gradient (or a numerical approximation to the first derivative of f or gradient) is discontinuous or non-existent at various points in $\boldsymbol{X}^{\mathbb{F}}$ because of kinks or other discontinuities in the hypersurface of f on $\boldsymbol{X}^{\mathbb{F}}$. Even though the objective function f is non-convex and non-smooth, it is assumed that f is not a random function on $\boldsymbol{X}$.

The location in $\boldsymbol{X}^{\mathbb{F}}$ of the search probe (or particle), referred to in MBH literature as the hopper**,** is defined as $\boldsymbol{x}[t]$ where t is the current iteration number, referred to as the MBH time-step. At this point in the present work, t has no particular relationship to more conventional definitions of time, e.g., wall-clock time, calendar-time, CPU execution time, etc. In later Chapters, t will be given more specific meaning.

$f[\boldsymbol{x}[t]]$ is the value of f at hopper location $\boldsymbol{x}[t]$. Convergence to f^* is defined as the first arrival by $f[\boldsymbol{x}[t]]$ to within a small epsilon distance of f^*. The value of t at the first arrival is referred to as the First Passage Time (FPT). Although f^* is unknown in real applications of MBH, f^* is known in simulation experiments and analyses in which f is constructed.

Although $\boldsymbol{X}^{\mathbb{F}}$ is often disconnected and sparse, thereby slowing MBH convergence, the boundaries of $\boldsymbol{X}^{\mathbb{F}}$ can defined either by a hypersurface that is not part of f, or by adding penalty functions to f. In practice, there may be reasons to define the boundaries of $\boldsymbol{X}^{\mathbb{F}}$ by a hypersurface that is distinct from f. However, for analytical purposes, defining the boundaries of $\boldsymbol{X}^{\mathbb{F}}$ by one or more penalty functions added to f is equivalent and enables the present work to refer to "poorly-behaved f" as a way of saying that the problem is hard because either f is non-convex and/or non-smooth, or because $\boldsymbol{X}^{\mathbb{F}}$is disconnected and sparse, or both.

In many of the simulation experiments included in the present work, $\boldsymbol{X}$ is a Cartesian product of bounded subspaces of non-negative $\mathbb{Z}^{\mathbf{N}}$, but every attempt has been made to make $\boldsymbol{X} \subset$ non-negative $\mathbb{Z}^{\mathrm{N}}$ as fine-grained, uniformly discretized as can be allowed by available memory in order to approximate $\boldsymbol{X} \subset \mathbb{R}^{\mathbf{N}}$ as closely as possible**.** When N = 1, $\boldsymbol{X}$ and $\boldsymbol{X}^{\mathbb{F}}$ are written X and $X^{\mathbb{F}}$. Points in $\boldsymbol{X}$ and $\boldsymbol{X}^{\mathbb{F}}$ are written as $\boldsymbol{x} = \{x_1, x_2, \ldots, x_{\mathrm{N}}\}$, whereas points in X and $X^{\mathbb{F}}$ are written as x. Incremental signed distances in $\boldsymbol{X}$ and $\boldsymbol{X}^{\mathbb{F}}$ are written as $\boldsymbol{\Delta x} = \{\Delta x_1, \Delta x_2, \ldots, \Delta x_{\mathrm{N}}\}$, whereas incremental signed distances in X and $X^{\mathbb{F}}$ are written as Δx.

In the present work, when $\boldsymbol{X} \subset \mathbb{R}^{\mathrm{N}}$ or $\boldsymbol{X} \subset \mathbb{Z}^{\mathrm{N}}$, the incremental signed distances $\boldsymbol{\Delta x}$ are vector random variables drawn from joint probability distribution $p_{\boldsymbol{\Delta x}}$, whereas when $X \subset \mathbb{R}^1$ or $X \subset \mathbb{Z}^1$, the Δx are scalar random variables drawn from probability distribution $p_{\Delta x}$**.** Except where noted otherwise, it is assumed that when $p_{\boldsymbol{\Delta x}}$ is a joint probability distribution, it is rotationally symmetric ("iso-directional") and the variables of $\boldsymbol{\Delta x}$ are i.i.d. across N. That assumption is based

on f being unknown. If one had some knowledge of f it might be beneficial to use a p that is not iso-directional, however that is not investigated in the present work. In the present work, because $p_{\boldsymbol{\Delta x}}$ is iso-directional and the variables of $\boldsymbol{\Delta x}$ are i.i.d. across N, discussions involving the adaptation of the shape of $p_{\boldsymbol{\Delta x}}$ are indifferent to N, refer to any of the marginal distributions of $p_{\boldsymbol{\Delta x}}$, and are developed in the same manner as they would be in the case of univariate $p_{\Delta x}$. A small neighborhood around $\boldsymbol{x}$ is written as $\boldsymbol{x+/-b}$, whereas a small neighborhood around x is written as $x+/-b$.

Because MBH problems are typically multi-dimensional, although 1-dimensional problems are constructed for analysis and simulation, $\boldsymbol{X}$, $\boldsymbol{X}^{\mathbb{F}}$, $\boldsymbol{\Delta x}$ and $\mathbf{x}$ are written in boldface throughout.

The feasible subspace $\boldsymbol{X}^{\mathbb{F}}$ is defined by j penalty functions $\phi_j(\boldsymbol{X})$ that are specific to the engineering application of MBH. Often, the $\phi_j(\boldsymbol{X})$ have physical meaning (e.g., not allowing a spacecraft to fly too close to the Sun) and are known by MBH practitioners even though f cannot be known.

MBH returns both $f^* = \min_{\text{global}}(f(\boldsymbol{X}))$ and $\boldsymbol{x}^* = \text{argmin}_{\text{global}}(f)$ presuming convergence in acceptable time. Both f^* and $\boldsymbol{x}^*$ are assumed to exist. In practical applications, it is allowable that f^* and/or $\boldsymbol{x}^*$ are not unique provided that one pair ($\boldsymbol{x}^*$, f^*) or ($\boldsymbol{x}^*+/-\boldsymbol{\delta}$, $f^*+/-\varepsilon$), for sufficiently small $\boldsymbol{\delta}$ and ε, is found within the allowable search time. There may be arbitrarily many local minima that are arbitrarily located and nearly as deep as f^*.

In general, both $\boldsymbol{x}^*$ and f^* are unknown except to an oracle. In the present work, the oracle is the author who constructs $f(\boldsymbol{X})$.The oracle's knowledge of $\boldsymbol{x}^*$ and f^* is used throughout the present work to construct FPT histograms which are used to characterize the convergence rates of

MBHs, and methods for accelerating them. In real applications of MBH, FPT histograms cannot be constructed within reasonable time frames.

The basic MBH algorithm, embellished in Chapter V, is specified as follows:

1. At every MBH time-step t, $\boldsymbol{x}[t\text{-}1]$ was defined by the previous time-step t-1 and was necessarily in $\boldsymbol{X}^{\mathbb{F}}$. Likewise, $f[\boldsymbol{x}[t\text{-}1]]$ was evaluated in the previous time-step t-1. For t=1, $\boldsymbol{x}[0]$ is chosen randomly but constrained to be in $\boldsymbol{X}^{\mathbb{F}}$, thereby defining $f[\boldsymbol{x}[0]]$.
2. Then, within the same MBH time-step t, draw $\boldsymbol{\Delta x} \sim \boldsymbol{p}_{\boldsymbol{\Delta x}}$ and generate $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t] + \boldsymbol{\Delta x})$.
3. Determine whether $\boldsymbol{\xi}[t]$ is in the feasible subspace $\boldsymbol{X}^{\mathbb{F}}$ of $\boldsymbol{X}$. If it is not, draw another $\boldsymbol{\Delta x}$ and, thereby, form another $\boldsymbol{\xi}[t]$.
4. Evaluate $f[\boldsymbol{\xi}[t]]$ and compare $f[\boldsymbol{\xi}[t]]$ to $f[\boldsymbol{x}[t]]$.
5. If $f[\boldsymbol{\xi}[t]] < f[\boldsymbol{x}[t]]$, then replace $f[\boldsymbol{x}[t]]$ with $f[\boldsymbol{\xi}[t]]$ and $\boldsymbol{x}[t]$ with $\boldsymbol{\xi}[t]$. If $f[\boldsymbol{\xi}[t]] \geq f[\boldsymbol{x}[t]]$, then $f[\boldsymbol{x}[t]]$ and $\boldsymbol{x}[t]$ remain unchanged.
6. Advance the iteration counter t and return to Step 1.

Thus, at every MBH time-step t, $\boldsymbol{x}[t]$ only moves if that move reduces $f[\boldsymbol{x}[t]]$. Otherwise, $\boldsymbol{x}[t]$ is forced to wait-in-place.

Increment $\boldsymbol{\Delta x}$ is the random perturbation on $\boldsymbol{x}[t\text{-}1]$ that defines $\boldsymbol{\xi}[t]$ and is drawn from distribution $\boldsymbol{p}_{\boldsymbol{\Delta x}}$. For brevity, $\boldsymbol{p}_{\boldsymbol{\Delta x}}$ is written as p even when $\boldsymbol{X}$ is, multi-dimensional. Distribution p is chosen, designed, and in Chapter IV adaptively shaped. It is constructed so as to have zero mean. In the present work, even when p is univariate it is constructed in such a manner that it is isotropic. Therefore, the marginal distributions of p are identical and the $[\Delta x_1, \Delta x_2, ...]$ are i.i.d. Distribution p is easily generalized so that it is anisotropic if there is reason to provide different scale factors in different dimensions, and/or cross-correlations between variates in different

dimensions. Doing so may be useful if f and/or $\boldsymbol{X}^{\mathbb{F}}$ were known to some extent or could be "learned" by an on-going MBH. That is left for future work.

Distribution p is said to be "well-suited" to f if, by the shape of p, $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})$ is frequently accepted as $\boldsymbol{x}[t+1]$. Distribution q is the distribution of "accepted" hop increments, meaning the distribution of $\Delta\boldsymbol{x}$ such that $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})$ is accepted as $\boldsymbol{x}$[t] because $f[\boldsymbol{\xi}[t]] < f[\boldsymbol{x}[t\text{-}1]]$. Distribution q cannot be known *a priori*. In fact, its existence is a conjecture. However, a near-real-time estimate of q, namely $\hat{q}$, can be constructed by a Monte Carlo method that is described in Chapter IV.

Figures II.1 through II.3 depict the basic MBH process operation on three versions of prototypical f. Figure II.4 depicts the histogram of the $\{ \Delta\boldsymbol{x} \}$ drawn from the fixed (non-adaptive) non-Gaussian p used to drive the MBH search in the above figures II.1, II.2, and II.3.

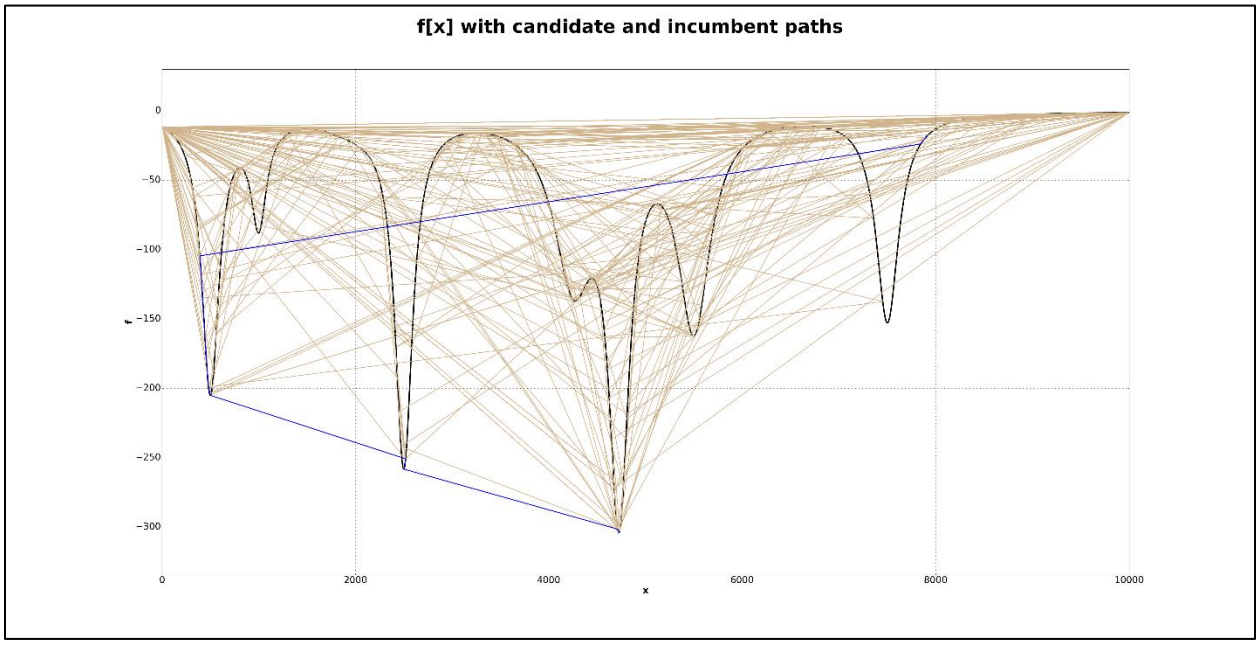


Figure II.1: Prototypical 1-dimensional non-convex f and an example of its associated incumbent path $\{\boldsymbol{x}[t]; t = 1, 2, 3, \ldots\}$ in blue and candidate path $\{\boldsymbol{\xi}[t]; t = 1, 2, 3, \ldots\}$ in tan

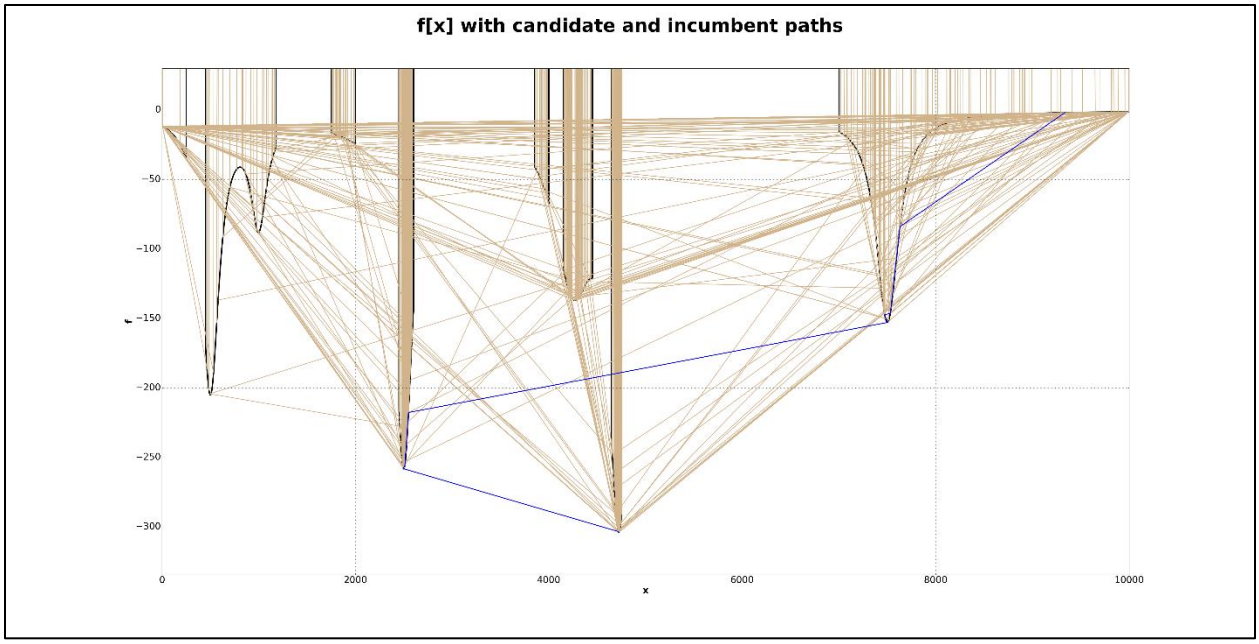


Figure II.2: The same 1-dimensional non-convex f as in Figure II.1, except with a disconnected, sparse $\boldsymbol{X}^{\mathbb{F}}$, and an example of its associated $\{\boldsymbol{x}[t]; t = 1, 2, 3, \ldots\}$ in blue and candidate path $\{\boldsymbol{\xi}[t]; t = 1, 2, 3, \ldots\}$ in tan

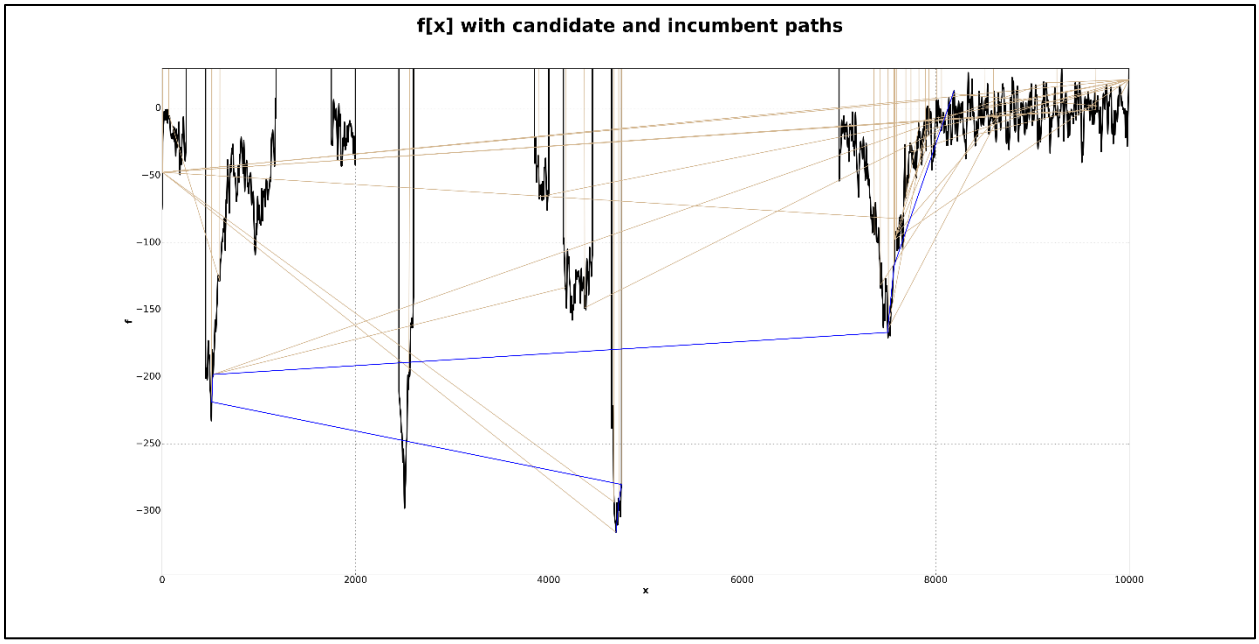


Figure II.3: The same 1-dimensional non-convex f as in Figure II.1, except with a disconnected, sparse $\boldsymbol{X}^{\mathbb{F}}$ and texture, and an example of its associated $\{\boldsymbol{x}[t]; t = 1, 2, 3, \ldots\}$ in blue and candidate path $\{\boldsymbol{\xi}[t]; t = 1, 2, 3, \ldots\}$ in tan

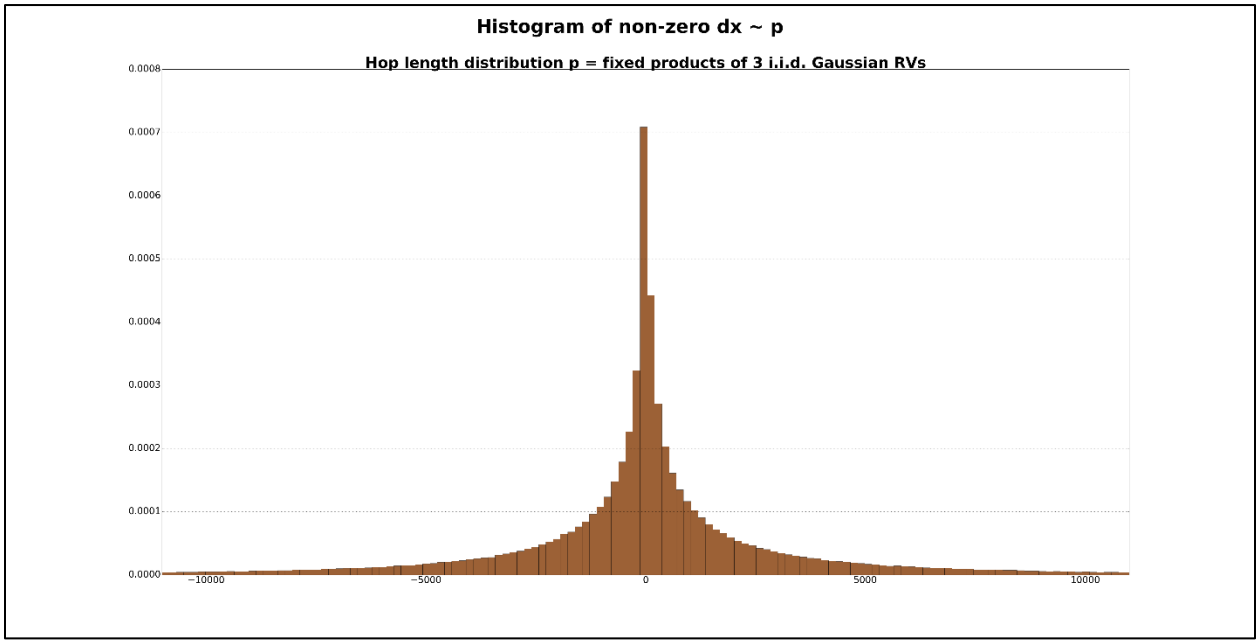


Figure II.4: Histogram of $\{\Delta\boldsymbol{x}\}$ drawn from the fixed non-Gaussian p used to drive the MBH search in the above figures II.1, II.2, and II.3.

II.b. Visualizing the MBH convergence process by plotting *f*[*x*[*t*]]

The MBH convergence process can be visualized by plotting $f[\boldsymbol{x}[t]]$, which is one-dimensional for $\boldsymbol{X}$ having any finite integer dimension. Plotting $f[\boldsymbol{x}[t]]$ not only illustrates the rate of the convergence of MBH operating on a given f having feasible domain $\boldsymbol{X}^{\mathbb{F}}$, driven by incremental hop distances drawn from a given p; it also illustrates the variability in the convergence rates given multiple trials of MBH operating on the same problem. The variability in the convergence rate is important because it suggests the likelihood of whether a given MBH convergence will require as little time as the fastest descending plot of $f[\boldsymbol{x}[t]]$ or slowest descending plot $f[\boldsymbol{x}[t]]$. In Chapter IV, more sophisticated methods, based on MBH FPTs, and fits of FPTs to Gamma distributions as MBH FPTDs, will be developed for characterizing MBH convergence rates. They provide a way to quantify and classify the distribution of MBH convergence times for an MBH operating on a given f sped-up by one method or another. Fits of FPTs to Gamma distributions as MBH FPTDs are an essential tool in assessing the effectiveness of methods for speeding-up MBH that will be provided in Chapters IV and V. Meanwhile, here, plots of $f[\boldsymbol{x}[t]]$ are sufficient to illustrate, in Figures II.5, II.6, and II.7, below, and Chapter III, the convergence impacting properties of various objective functions f given a fixed p.

In Figures II.5, II.6, and II.7 below, the upper panel shows the f upon which the MBH is operating. The lower panel shows the corresponding $f[\boldsymbol{x}[t]]$, illustrating the advancement of a trial's descent towards the global minimum. In all cases, the p from which the incremental hop distances $\boldsymbol{\Delta x}$ are drawn, is the fixed narrow-bodied, long-tailed p having the histogram shown below in Figure II.4.

In Figures II.6 and II.7, the impact of disconnected $\boldsymbol{X}^{\mathbb{F}}$ is apparent in the greater variability in the convergence times especially for convergence within MBH time-steps < 500.

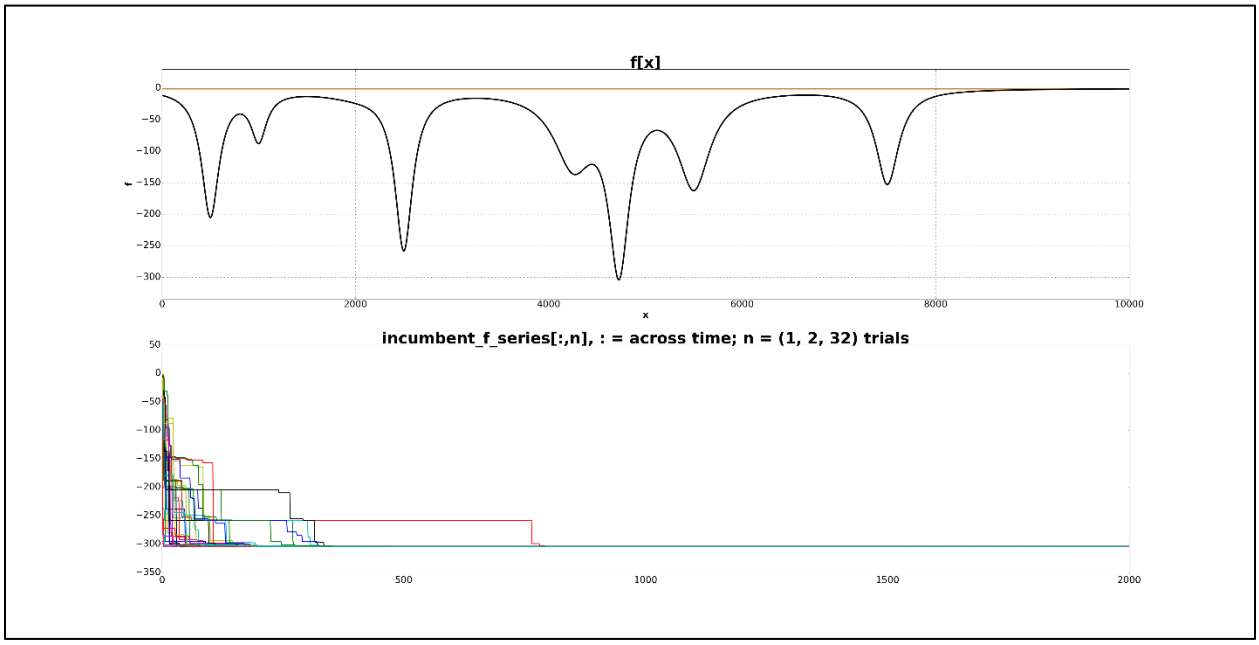


Figure II.5:

Upper panel: The same prototypical 1-dimensional non-convex f shown in Figure 0.a. The light brown horizontal line passes through maximum feasible f.

Lower panel: The corresponding f[$\boldsymbol{x}[t]$] for 32 trials of an MBH, all of which are using the same fixed non-Gaussian p having the histogram shown in Figure II.4. The horizonal axis is in units of MBH time-steps. The vertical axis is in units of values of f.

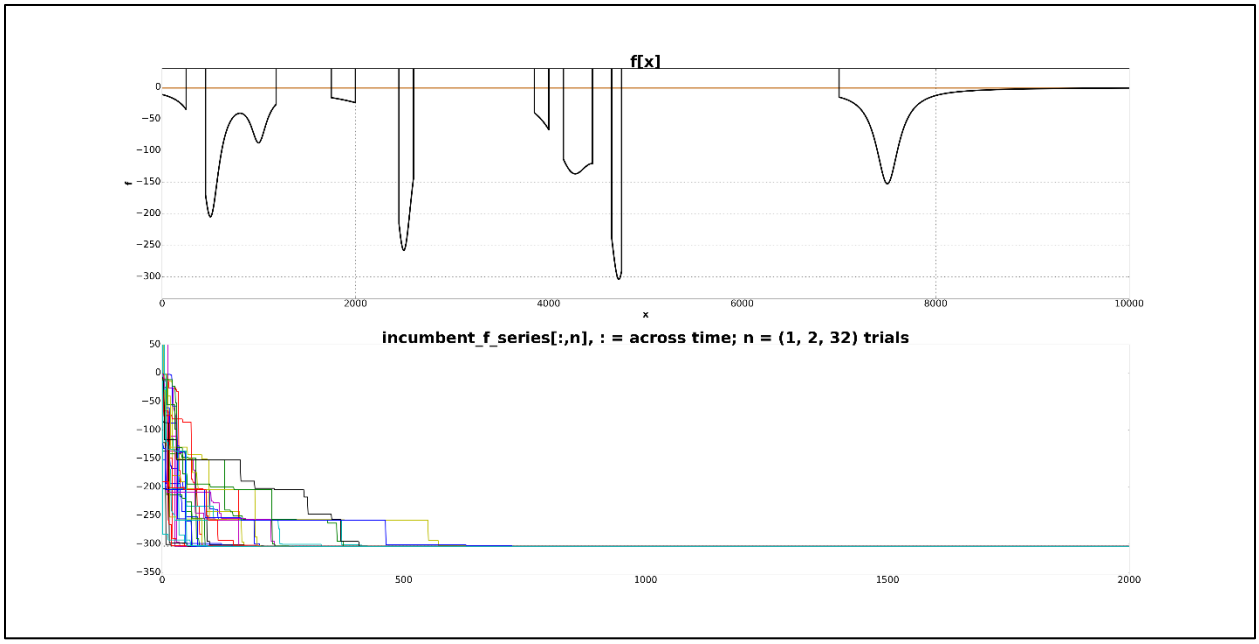


Figure II.6:

Upper panel: The same prototypical 1-dimensional non-convex *f* with a disconnected, sparse domain shown in Figure 0.b. The light brown horizontal line passes through maximum feasible *f*.

Lower panel: The corresponding f[***x***[*t*]] for 32 trials of an MBH, all of which are using the same fixed non-Gaussian *p* having the histogram shown in Figure II.4. The horizonal axis is in units of MBH time-steps. The vertical axis is in units of values of *f*.

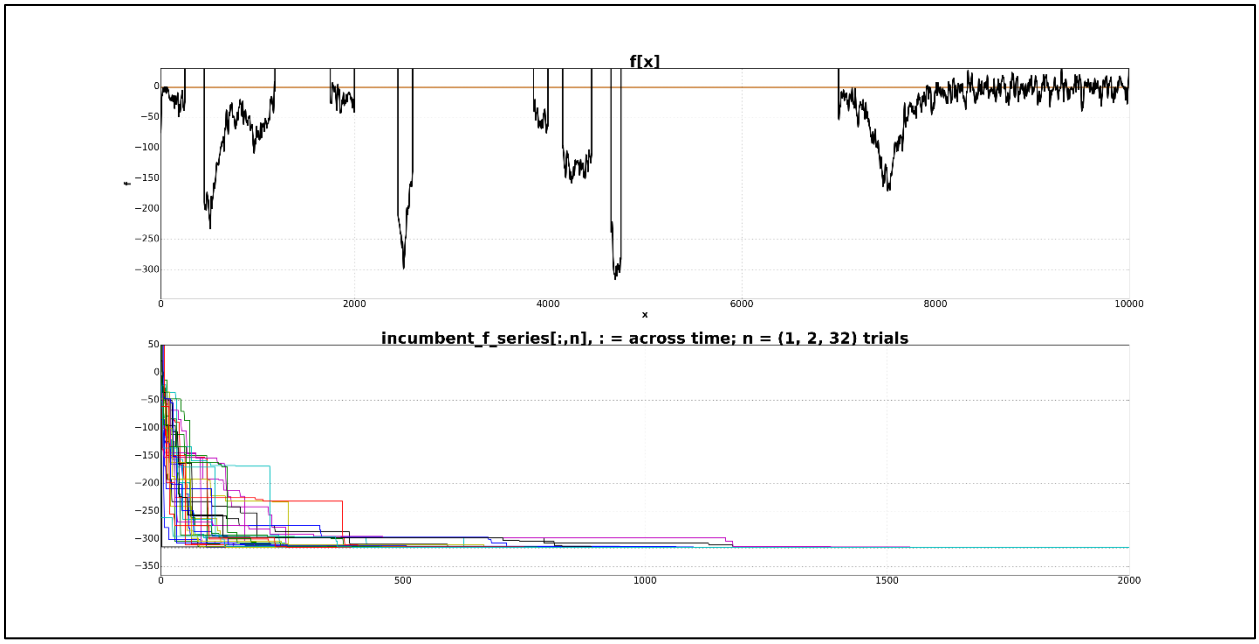


Figure II.7:

Upper panel: The same textured prototypical 1-dimensional non-convex f with a disconnected, sparse domain shown in Figure 0.c. The light brown horizontal line passes through maximum feasible f.

Lower panel: The corresponding f[$\boldsymbol{x}$[t]] for 32 trials of an MBH, all of which are using the same fixed non-Gaussian p having the histogram shown in Figure II.4. The horizonal axis is in units of MBH time-steps. The vertical axis is in units of values of f

Function *g*[*d*]

In order to analyze the dynamics of $f[\boldsymbol{x}[t]]$, a function $g[d]$ is useful. $g[d]$ is the remaining volume of productive search space when $f[\boldsymbol{x}] = d$. The set (level set) $\boldsymbol{U}_f[\mathrm{d}] = f^{-1}[\mathrm{d}]$, where f^{-1}, is the set inverse. The sub-sets $\widetilde{U}_f[d]$ are those $\boldsymbol{x} \in \boldsymbol{X}^{\mathbb{F}}$ such that $\boldsymbol{U}_f[\acute{d}] \subset \boldsymbol{U}_f[\mathrm{d}]$ for all $\acute{d} < d$. This leads to the definition of $g[d]$:

$$g[d] \triangleq (1/\mu(\boldsymbol{X}^{\mathbb{F}}))\int_{\boldsymbol{X}^{\mathbb{F}}{}_{min}}^{\boldsymbol{X}^{\mathbb{F}}{}_{max}} I_{f,d}(\boldsymbol{x})\, d\boldsymbol{V}$$

$$\text{where } I_{f,d}(\boldsymbol{x}) = 1\begin{cases} 1\ if\ f[\boldsymbol{x}] < d \\ 0\ otherwise \end{cases}$$

and $d\boldsymbol{V}$ is the infinitesimal volume for the space $\boldsymbol{X}$. $\mu(\boldsymbol{X}^{\mathbb{F}})$ is the volume of the feasible search space, and $\mu(\boldsymbol{X}^{\mathbb{F}}) = \int_{\boldsymbol{X}^{\mathbb{F}}{}_{min}}^{\boldsymbol{X}^{\mathbb{F}}{}_{max}} 1 \cdot d\boldsymbol{V}$.

Function $g[d]$ is related to the probability, at depth d, of a random probe finding a yet smaller value of f at depths deeper than d, meaning that $\boldsymbol{\xi}[t] = (\boldsymbol{x}[\mathrm{t}\text{-}1] + \Delta\boldsymbol{x})$ is accepted as $\boldsymbol{x}[t]$ because $f[\boldsymbol{x}[t\text{-}1]+\Delta\boldsymbol{x}] < f[\boldsymbol{x}[t\text{-}1]]$. Thus, $g[d]$ is a relative measure of the likelihood of drawing an acceptable $\Delta\boldsymbol{x} \sim p$, therefore a relative measure of the expected rate of progress toward f^*, given $f[\boldsymbol{x}[t]] = d$. Thus, $g[d]$ is determined by f and shows that the relative measure of the expected rate of progress toward f^* depends on f.

The construction of $g[d]$ requires knowledge of f. In real applications of MBH, f is query-able at a limited number of points $\boldsymbol{x}$ but otherwise unknown. So, $g[d]$ cannot be constructed in practice. However, for analytical purposes and in simulations in the present work, $g[d]$ can be constructed because f is known by construction.. However, $g[d]$ establishes a bound on the probability of an accepted hop based on the remaining productive search volume.

$g[d]$ is a non-increasing function. $g[d]$ is normalized to be independent of the range of $f(\boldsymbol{X})$, so max(g) = 1.0 when $d = \max(f[\boldsymbol{x}[t]])$ and min(g) = 0 when $d = f^*$. As $d \rightarrow f^*$, g[d] $\rightarrow$ 0 because smaller values of f become increasingly rare. $g[d]$ is depicted for two different f in Figure II.8. The different rates at which $g[d]$ decays, depending upon the geometry of f, is apparent.

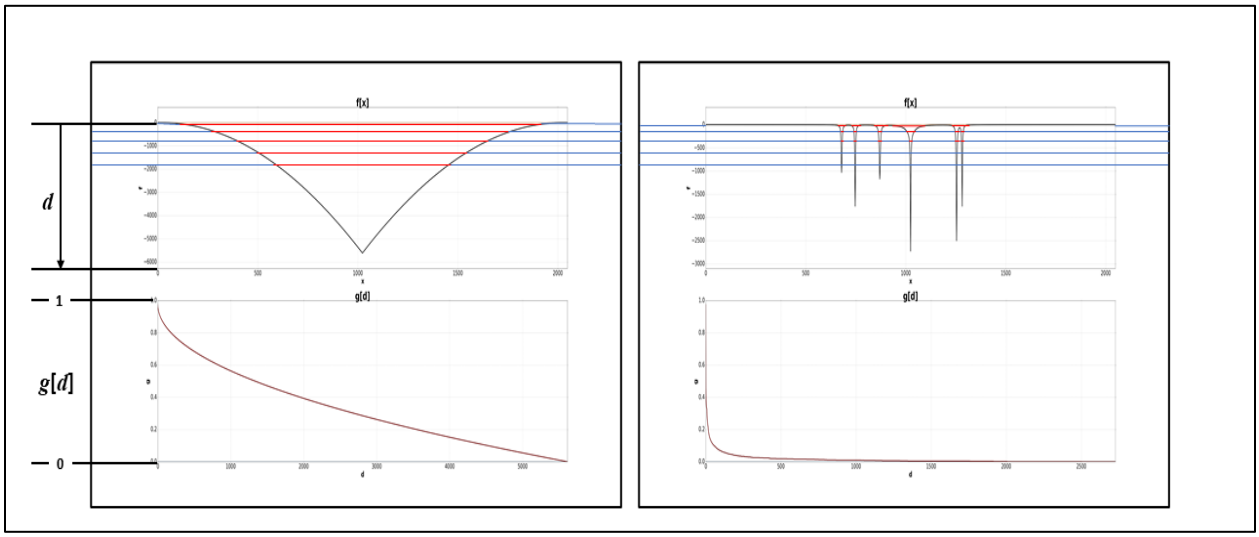


Figure II.8:

Left: $g[d]$ as it would be constructed by an "oracle" given a unimodal globally and locally smooth f. Upper panel: The process of deriving $g[d]$ from f per the definition of $g[d]$ provided in the text above. Lower panel: The resulting $g[d]$

Right: $g[d]$ as it would be constructed by an "oracle" given a multi-modal, non-convex but locally smooth f. Upper panel: The process of deriving $g[d]$ from f per the definition of $g[d]$ provided in the text above. Lower panel: The resulting $g[d]$

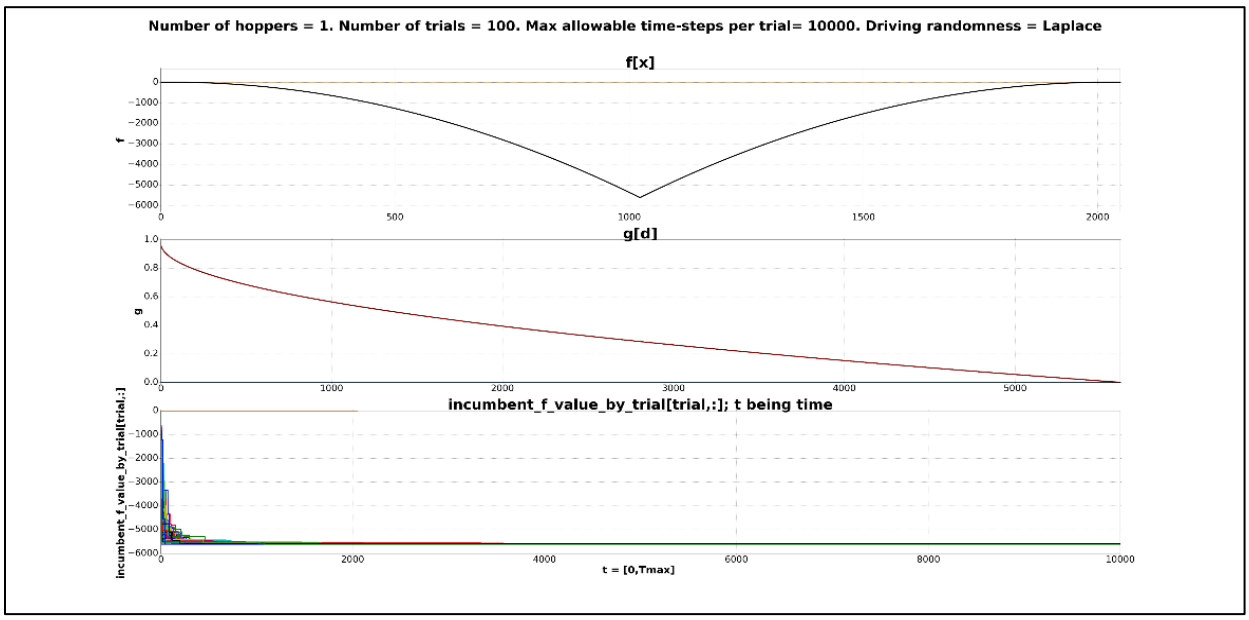


Figure II.9:

Upper panel: A simple f for which for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$.

Middle panel: The corresponding $g[d]$

Lower panel: The corresponding $f[\boldsymbol{x}[t]$ for an MBH operating on f using $\boldsymbol{\Delta x}$ drawn from a fixed Laplace(0,1) p scaled to cover the extent of $\boldsymbol{X}$

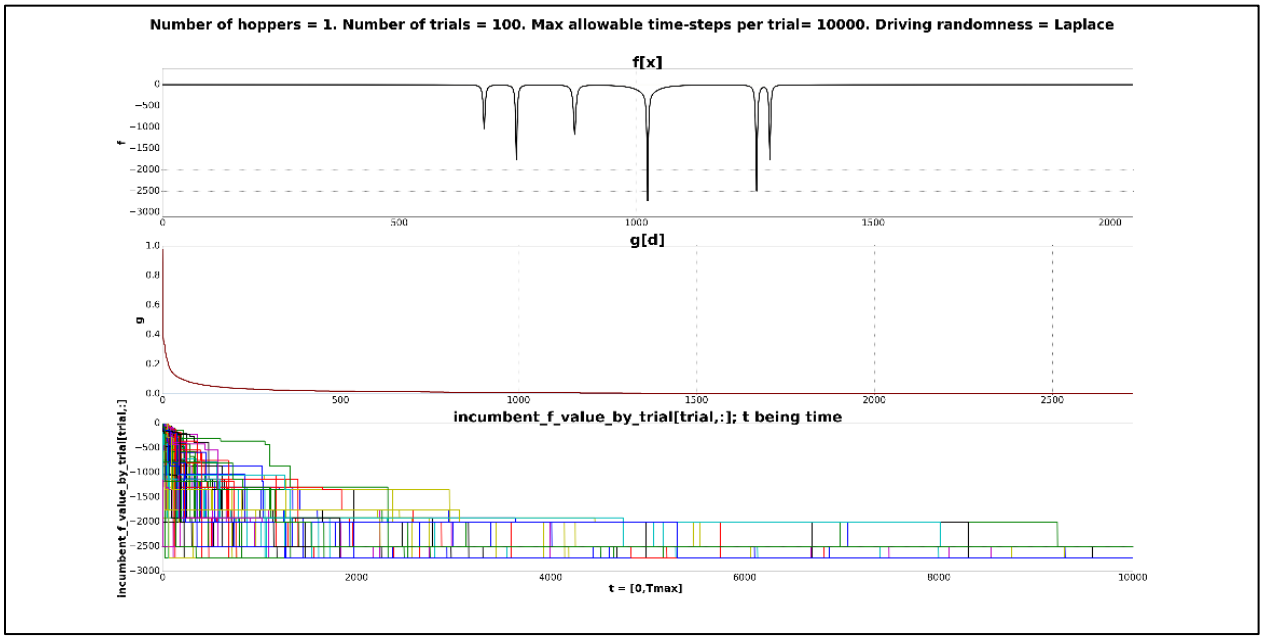


Figure II.10:

Upper panel: Globally rugged f for which for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$.

Middle panel: The corresponding $g[d]$

Lower panel: The corresponding f[$\boldsymbol{x}$[t] for an MBH operating on f using $\boldsymbol{\Delta x}$ drawn from a fixed Laplace(0,1) p scaled to cover the extent of $\boldsymbol{X}$.

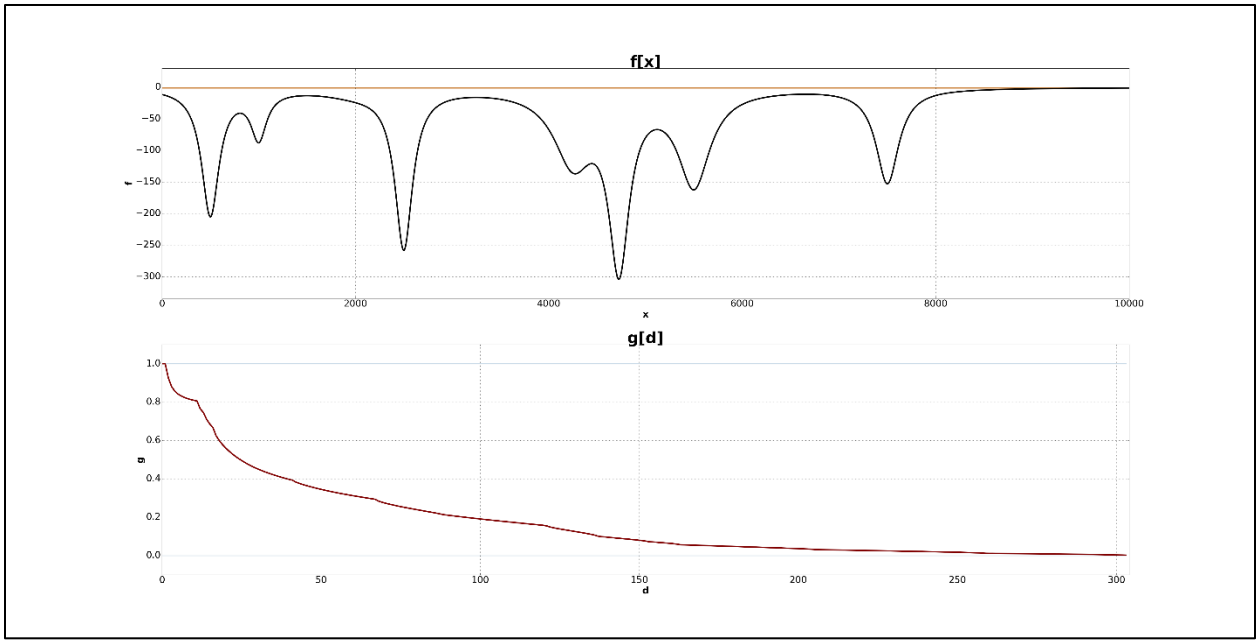


Figure II.11: Prototypical 1-dimensional non-convex *f* and its associated *g*[*d*].

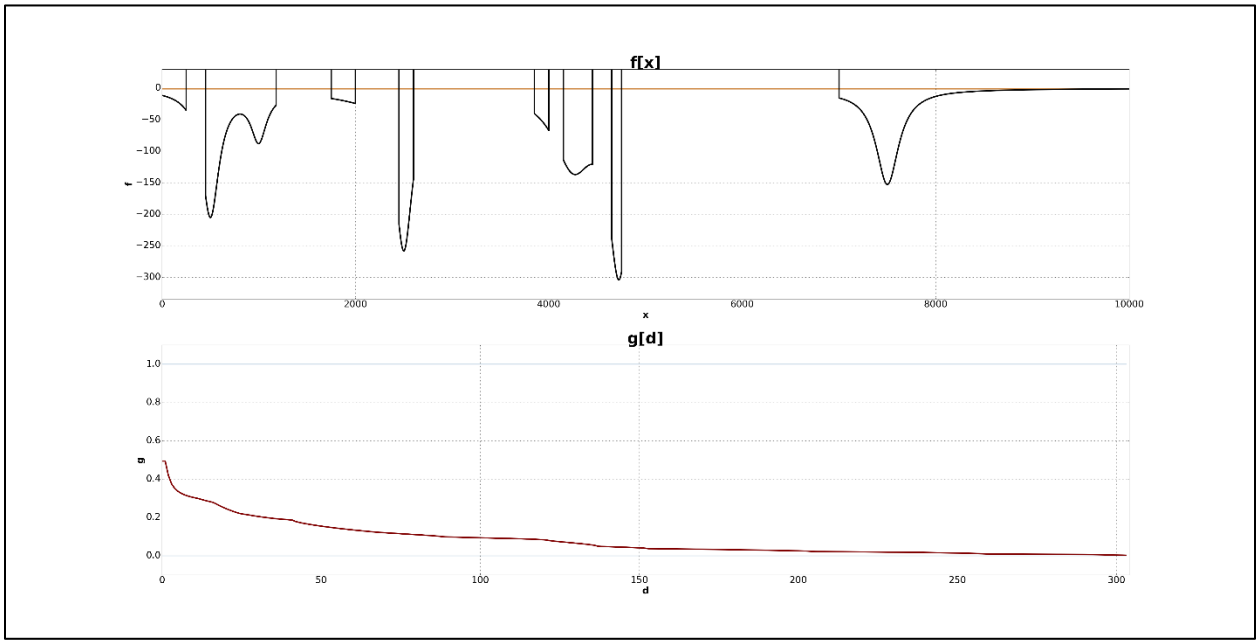


Figure II.12: Prototypical 1-dimensional non-convex *f* having disconnected and sparse feasible sub-domains, and its associated *g*[*d*].

III. THE BEHAVIOR OF f AND $X^{\mathbb{F}}$ IMPACT MBH CONVERGENCE TIME

III.a The nature of the objective function f and its feasible domain $X^{\mathbb{F}}$

Recall that objective function f is assumed to be non-convex, and possibly non-smooth, and defined over the feasible domain $\boldsymbol{X}^{\mathbb{F}} \subset \boldsymbol{X}$. The feasible neighborhood surrounding any minimum in f is referred to as a basin. Their shapes, locations, and the minima within them, are known only by an oracle. In physics and microbiology, particularly in protein-folding research, where f is thought of as an "energy landscape", basins are referred to as "funnels". The funnels are drawn as though they are three dimensional, but they are typically high-dimensional (e.g., 150 dimensions.) In spacecraft trajectory optimization, where the term basin is used, $\boldsymbol{X}$, therefore the basins, may be 500 dimensional (or higher), although the spacecraft trajectory optimization problem chosen as the use-case in Chapter VI is 3-dimensional.

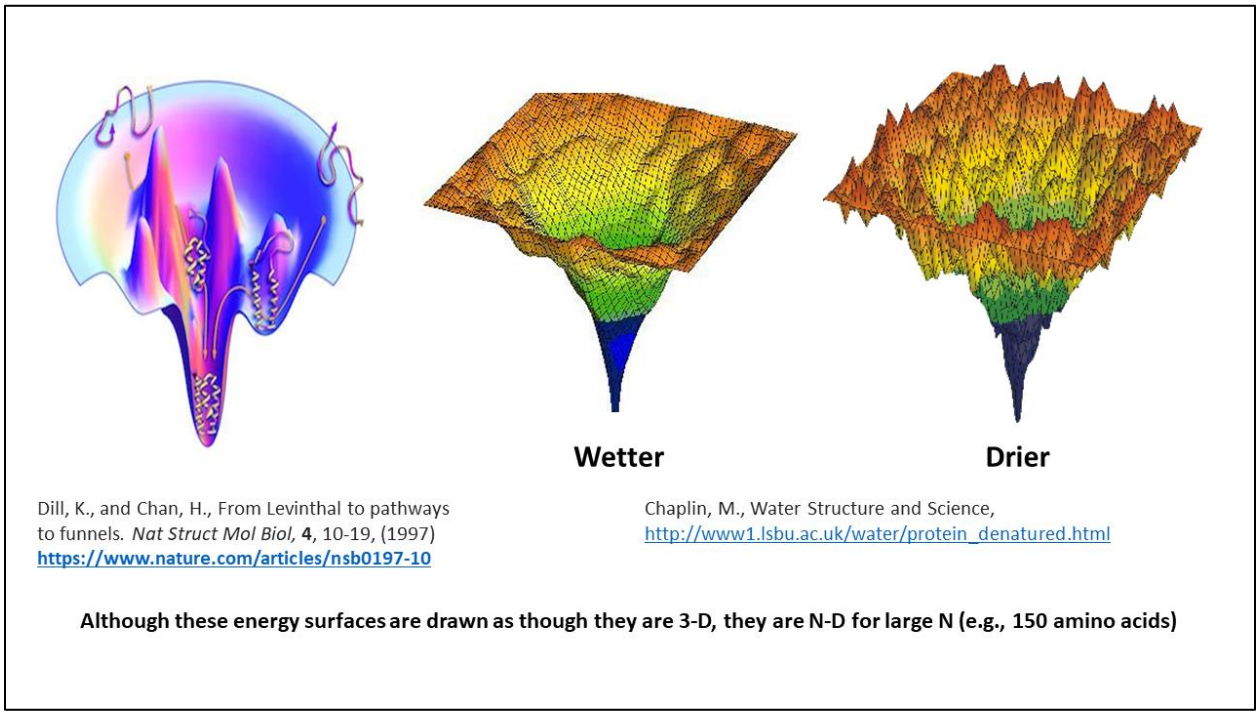


Figure III.1: 3-dimensional "funnels" that are used as pedagogical models of energy landscapes in protein-folding literature, and that that microbiologists intuitively extend to higher dimensions [41, 42]

Although multi-dimensional "funnels" are accepted in microbiology as intuitively sensible, multi-dimensional basins warrant a formal definition: A basin is a neighborhood around a local minimum $\boldsymbol{x}^{\#}$ in which $f[\boldsymbol{x}]$ is everywhere greater than $f[\boldsymbol{x}^{\#}]$. This definition allows basins to be nested inside of one another or spread far apart, and it is meaningful in any number of dimensions. At saddle points in f, the definition needs to be extended to a basin being a neighborhood around a local minimum $\boldsymbol{x}^{\#}$ in which $f[\boldsymbol{x}^{\#}]$ is everywhere greater than $f[\boldsymbol{x}^{\#}]$ in some but not necessarily all dimensions. The added precision in the definition should not be construed to imply that saddle points in f pose challenges for MBH.

Despite the fact that, in practice, f and $\boldsymbol{X}$ are represented in IEEE floating point precision and therefore "noisy" at a "microscopic" scale, in the present work f is not a random variable on $\boldsymbol{X}$. Both f^* and $\boldsymbol{x}^*$ exist, although there may be arbitrarily many local minima that are arbitrarily located, and nearly as deep as f^*. Because $\boldsymbol{X}$ is bounded, then f has an upper bound except where it may have a singularity. f has a lower bound at f^*. Without loss of generality, f may be transformed to have its minimum be zero and thereby be non-negative throughout $\boldsymbol{X}$. In most of the simulations in the present work, f has negative values that are minimized. In the use-case, f has positive values that are minimized. In the use case, f is the integral in change in velocity that results from the necessarily non-negative consumption of on-board propellent.

The MBH convergence process is inherently slower and harder to speed-up when f is poorly-behaved. Visual evidence for this can be seen by comparing figures II.5, II.6, and II.7, and II.9 versus II.10 in Chapter II. Because the present work focuses on random search on poorly-

behaved *f*, it is important to describe examples of poorly-behaved *f* and how they arise in real-world optimization problems.

III.b An extreme example of poorly-behaved *f* and its impact on MBH convergence time

A poorly-behaved *f* that imposes severe adverse impacts on MBH convergence speed, is one in which *f** is far in $\boldsymbol{X}$ from a local minimum that is nearly as deep as *f** and resides in a much wider basin than the basin in which *f** resides. The present work refers to such poorly-behaved *f* as Gibsonian *f* in gratitude to Professor John Gibson who first proposed it in conversation [August 28, 2020]. The challenge of Gibsonian *f* is that a candidate hop has a vector probability distribution that is required to generate sufficiently many long hops when $\boldsymbol{x}[t]$ is within the wide basin of the sub-optimal local minimum, to enable $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1]+\Delta\boldsymbol{x})$ to hop to the narrow basin in which $\boldsymbol{x}^*$ resides. However, once $\boldsymbol{x}[t]$ is inside the narrow basin in which $\boldsymbol{x}^*$ resides, those long hops cause $\boldsymbol{\xi}[t]$ to "overshoot" $\boldsymbol{x}[t]$, slowing the convergence to *f**. As will be shown in Chapter V, MBH on Gibsonian *f* can nonetheless be sped-up, thereby enabling MBH optimizations on many examples of Gibsonian *f* to find the global minimum within allowable periods of time. The methods provided in Chapter V for speeding-up MBH on Gibsonian *f* speeds-up the descent of $f[\boldsymbol{x}[t]]$ deeper into $g[d[t]]$, closer to *f**, regardless of whether $\boldsymbol{x}[t]$ moved closer to $\boldsymbol{x}^*$.

Figures III.2 illustrates Gibsonian *f.* As in the case of Figure III.1 above from the literature of protein folding, Gibsonian *f* may imagined as a projection onto one dimension of a high-dimensional *f* that may be Gibsonian in some but not all dimensions. Real-world applications of MBH often involve a multi-dimensional *f* that is Gibsonian or otherwise severely poorly-behaved when projected onto some but not all dimensions. The use-case described in Chapter VI involves a 3-dimensional *f* that is differently poorly-behaved when projected upon each of the three dimensions.

Methods for speeding-up MBH that are provided and analyzed in Chapter IV are not very effective when MBH is operating on the 1-dimensional Gibsonian *f* depicted in Figure III.2. However, methods provided in Chapter V are highly effective for speeding up MBH operating on the 1-dimensional Gibsonian *f* depicted in Figure III.2 (as will be shown in Figures V.1 through V.3). Methods provided in Chapter V are also highly effective for speeding up MBH operating on a 2-dimensional Gibsonian *f* that will be depicted in Figure V.8 as will be shown in Figures V.9 through V.11.

Figure III.2 and III.3 below illustrate the special challenge posed by Gibsonian *f* being operated upon by the basic MBH algorithm without any speed-up method. The lower panel of Figure III.2 is a magnification of the time-step axis of the lower panel in Figure III.3. In Figure III.2, it can be seen that all of the 48 trials shown find the non-global minimum in the wide basin to the left quickly, i.e., in fewer than 300 MBH time-steps and in most of the trials in fewer than 100 MBH time-steps. However, Figure III. 3 shows that many of the 48 trials did not converge to the global minimum sooner than the $100{,}000^{th}$ MBH time-step, and one of the 48 trials did not find the global minimum until after the $500{,}000^{th}$ MBH time-step.

While the eventual convergence the global minimum by the slowest trial in Figure III.3 is an empirical confirmation that MBH will eventually converge, in this case after 500,000 MBH time-steps, as assured by the asymptotic convergence proof, the large number of MBH time-steps required for convergence on this 1-dimensional example of Gibsonian *f* demonstrates why practitioners would regard MBH as having failed to converge in practically allowable time. When, in Chapter V, a method is provided for speeding up the convergence of MBH operating on this example of 1-dimensional Gibsonian *f*, it will be illustrated using plots that have, as their lower

panel time-step axis, [0,2000] as is typically used in the present work, because a lower panel time-step axis of [0, 650000] will no longer be needed.

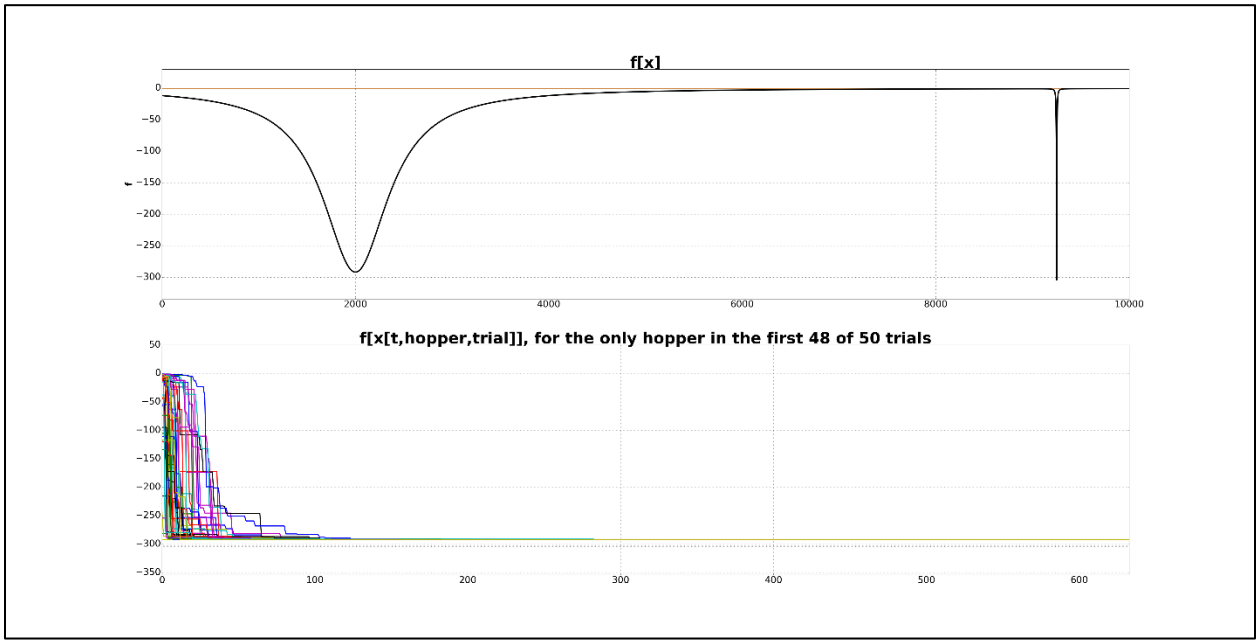


Figure III.2:

Upper panel: 1-dimensional Gibsonian f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$. The MBH used fixed non-Gaussian p, the histogram for which is shown in Figure II.4. The light brown horizontal line passes through maximum feasible f

Lower panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 650 of 650,000 MBH time-steps, n being the first 48 of 50 trials.

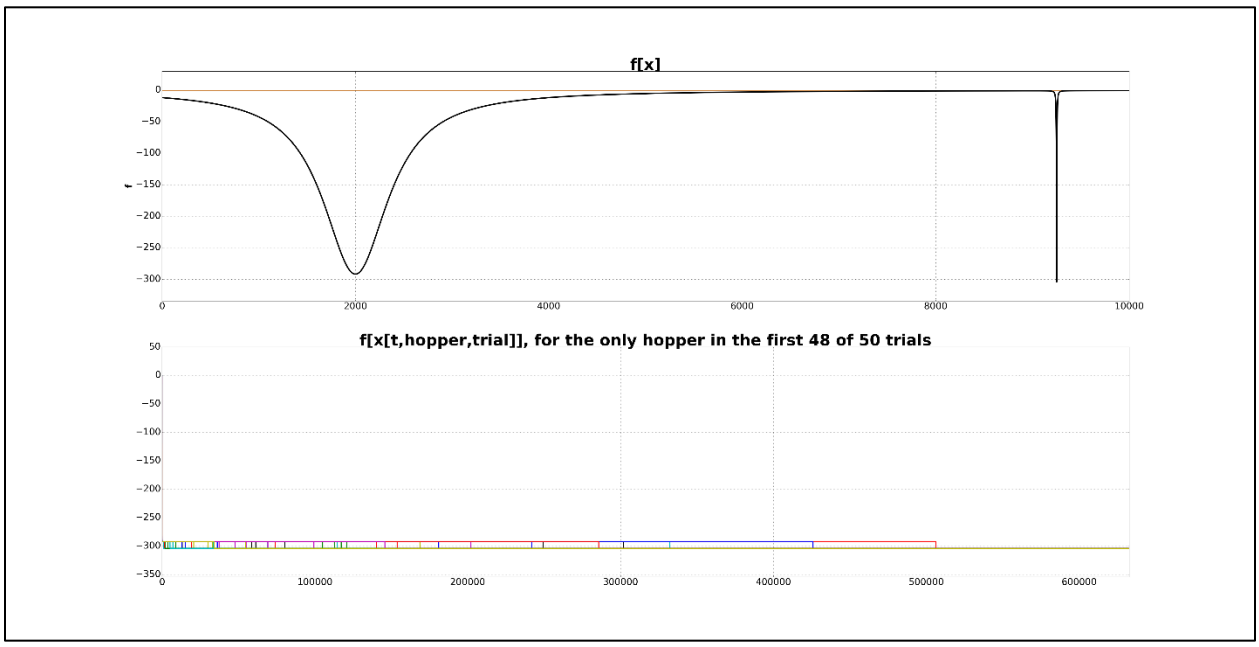


Figure III.3:

Upper panel: 1-dimensional Gibsonian f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$. The MBH used fixed non-Gaussian p, the histogram for which is shown in Figure II.4. The light brown horizontal line passes through maximum feasible f

Lower panel: $f[\boldsymbol{x}[t,n]]$ for t being 650,000 MBH time-steps, n being the first 48 of 50 trials.

III.c Causes of poorly-behaved *f* in optimization applications

Examples and causes of poorly-behaved *f* and/or $\boldsymbol{X}^{\mathbb{F}}$ in real-world optimization applications are plentiful in optimization-based approaches to molecular modeling and in spacecraft trajectory optimization.

In modeling molecular configurations by minimizing energy landscapes, non-convexities and non-smoothness in *f* are inherent in physical laws that are assumed (e.g., multi-molecule Lennard-Jones potentials) [43]. These physical laws also define $\boldsymbol{X}^{\mathbb{F}}$ that are sometimes disconnected and sparse. In addition, some researchers hypothesize that, at least in the case of protein folding, energy landscapes are multifractal (therefore very rough) to an extent that corresponds to the "wetness" of the immediate micro-environment in which the protein is folding [44, 45].

In inter-planetary spacecraft trajectory optimization problems, non-convexity and non-smoothness in *f* may result from one or more of several sources: periodicities and resonances between the gravitational forces imposed on spacecraft by large planets and the Sun; non-linearities in feasible flight control actions; and close proximity in $\boldsymbol{X}^{\mathbb{F}}$ to singularities in trajectory models, such as arise from the use of Lambert's problem [21]. Disconnected and sparse $\boldsymbol{X}^{\mathbb{F}}$ results from singularities in trajectory models, as well as having to avoid traveling through time or space into un-survivable conditions. In addition, disconnected and sparse $\boldsymbol{X}^{\mathbb{F}}$ results from the infeasibility of some flight control actions, and the fact that celestial bodies are in motion and therefore they are only "in range" if launches from Earth and "fly-bys" around certain planets occur within narrow temporal windows. The resulting poorly-behaved *f* is depicted Chapter VI where the *f* for the Pioneer 11 trajectory optimization use-case is illustrated.

Severe poor-behavior in f arises in a class of inter-planetary spacecraft trajectory optimization problems called “moon tours” which involve complicated trajectories around a moon or multiple. The severe poor-behavior in f that results from such resonances arise in tens, out of hundreds, of dimensions in $\boldsymbol{X}$.

Poorly-behaved f also arise in economics (where Paul Samuelson wrote that the difficulties of studying anything other than pure concavity were “shrouded in eternal darkness”), “fitness landscapes” in models of evolution, as well as other fields [46, 47, 48]. Poorly-behaved f also arise in abstract and in hypothetical physical models, such as in the theory of large-scale energy landscapes of randomly pinned manifolds which have been proposed as models of pinned flux lines in superconductors [49].

IV. ACCELERATING MBH BY ADAPTIVELY SHAPING THE DISTRIBUTION FROM WHICH INCREMENTAL HOP DISTANCES ARE DRAWN

This chapter concerns biasing the probability distribution of incremental hop distances in such a manner that makes a "beneficial sequence" of incremental hop distances $\{\Delta\boldsymbol{x}[t]\}$ more likely to be drawn. "Beneficial sequence" means a sequence of short versus long incremental hop distances $\{\Delta\boldsymbol{x}[t]\}$ that is likely to speed-up the convergence of the MBH to f^* given f and $\boldsymbol{X}^{\mathbb{F}}$ by providing $\Delta\boldsymbol{x}$ that will cause $\xi[t] = (\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})$ to be "accepted" as the new $\boldsymbol{x}[t]$ by satisfying the conditions $(\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x}) \subset \boldsymbol{X}^{\mathbb{F}}$ and $f[(\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})] < f[\boldsymbol{x}[t\text{-}1]]$.

IV.a Three questions addressed in this chapter

This chapter asks and answers the following three questions: First, given an f and $\boldsymbol{X}^{\mathbb{F}}$, what is the impact of the shape of distribution p, from which the $\Delta\boldsymbol{x}$ are drawn, on the MBH convergence time? Second, for a set of f and $\boldsymbol{X}^{\mathbb{F}}$, e.g., some appropriately defined equivalence class $\mathcal{C} \equiv \{(f_i, \boldsymbol{X}_i^{\mathbb{F}})\}$ that have similar local minima structure, is there a p that is universal in the sense that its use results in faster convergence on all $(f_i, \boldsymbol{X}_i^{\mathbb{F}}) \subset \mathcal{C}$ compared to different p? And third, given a particular unknown f and $\boldsymbol{X}^{\mathbb{F}}$, is there a method for adaptively shaping p so that MBH convergence can be sped-up?

IV.b Additional concepts and tools for the analytical framework

In order to answer these three questions, new concepts and tools need to be added to the analytical framework that currently contains $g[d]$. Each new concept and tool is introduced as it is needed, and its use builds upon the concepts and tool already introduced. Because the first question is the most general of the three, the concept and tool it requires is the first addition to the

analytical framework: the expected MBH search efficiency over the time-step interval $[\tau_a : \tau_b]$. By itself, expected MBH search efficiency is used to answer question one.

The second question was addressed empirically in a 2014 paper by Englander and Englander wherein it was demonstrated that the answer is affirmative, and the properties of such p were described. But the analysis supporting those empirical findings was not developed until 2020 when parts of this chapter were first published. The analytical answer to question two requires that more tools be added to the analytical framework: A specification of assumptions on an idealized distribution q comprised of the $\boldsymbol{\Delta x}$ that resulted in "accepted" $\xi[t]$, amounting to a description of the nature of q; a Monte Carlo method for constructing an estimate of q, namely $\hat{q}$; the Kullback-Leibler divergence $D_{K\text{-}L}(p, \hat{q})$ used to measure the similarity between p and $\hat{q}$; and the use of First Passage Times (FPTs), First Passage Time Probability Mass Functions (FPT-PMFs), and fits of Gamma distributions to FPTs as First Passage Time Densities (FPTDs), to characterize and quantify the speed-up of MBH convergence that is achieved by p making similar to $\hat{q}$.

The third question, which is a natural extension of questions one and two, is also addressed analytically in this chapter. After explaining the answer analytically, it is illustrated by figures within this chapter and applied to the use-case described in Chapter VI. The analytical answer to question three requires that additional tools be added to the analytical framework: definitions, assumptions, and methods for adapting p to $\hat{q}$ in order to make the shape of p similar to the shape of $\hat{q}$. Then FPTs and FPTDs are used to characterize and quantify the additional speed-up of MBH convergence that is achieved by making p similar to $\hat{q}$ by adaptation rather than by using well-chosen or well-designed fixed p. It will be shown that the source of additional speed-up is the fact that the shape of $\hat{q}$ is time-varying during the MBH convergence process and that any fixed p can

at best be a compromise between being similar to $\hat{q}$ when the MBH is far from converging versus similar to $\hat{q}$ as the MBH approaches convergence.

IV.c Expected MBH search efficiency

The expected MBH search efficiency over the time-step interval $[\tau_a : \tau_b]$:

$$\mathbb{E}(\eta[\tau_a : \tau_b]) = (1/(\tau_b - \tau_a)) \sum_{j=\tau_a}^{\tau_b} I_{\boldsymbol{x}}(j)$$

$I_{\boldsymbol{x}}[t] = 1$ if $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})$ is accepted as $\boldsymbol{x}$[t], and $I_{\boldsymbol{x}}[t] = 0$ if $\xi[t] = (x[\text{t-}1] + \Delta x)$ is "rejected", therefore $\mathbb{E}(\eta[\tau_a : \tau_b])$ is dependent on upon the magnitude of $\Delta\boldsymbol{x}$ making $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})$ "acceptable" sufficiently often, because the more often $\boldsymbol{\xi}[t]$ is "accepted", the less often the MBH process is forced to "wait-in-place", and the fewer MBH time-steps are required for to reach f^*. The efficiency of the search is dependent on f, $\boldsymbol{X}^{\mathbb{F}}$, and the frequency during time-step interval $[\tau_a : \tau_b]$ with which draws from p produce $\Delta\boldsymbol{x}$ that form acceptable $\boldsymbol{\xi}[t]$. $\mathbb{E}(\eta[\tau_a : \tau_b]) = 1$ is completely efficient and $\mathbb{E}(\eta[\tau_a : \tau_b]) = 0$ is completely inefficient.

In order for an MBH process to move efficiently, p needs to provide a sufficient number of long-distance hops in a sequence that cannot be predicted but can be described using scenarios. For example, as $\boldsymbol{x}[t]$ approaches near to $\boldsymbol{x}^*$, short-distance hops are needed to reduce the frequency of $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})$ being "unacceptable because long hops would result in $\boldsymbol{\xi}[t]$ "overshooting" $\boldsymbol{x}^*$. When $\boldsymbol{x}[t]$ is far from $\boldsymbol{x}^*$ long-distance hops are needed, especially if $\boldsymbol{x}[t]$ is "trapped" in a basin far from $\boldsymbol{x}^*$ that contains a sub-optimal local minimum. This implies that if one could choose a p that provides a high likelihood of generating $\Delta\boldsymbol{x}$ that would result in "accepted" $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \Delta\boldsymbol{x})$, one could increase the efficiency as defined above, and thereby speed-up MBH convergence.

IV.d Expected efficiency answers question one and begins to answer question two

By itself, the definition of expected MBH search efficiency provides the answer to question one: given an f and $\boldsymbol{X}^{\mathbb{F}}$, what is the impact of the shape of distribution p, from which the $\boldsymbol{\Delta x}$ are drawn, on the MBH convergence time? The answer is obvious: The shape of p that provides $\boldsymbol{\Delta x}$ that cause $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \boldsymbol{\Delta x})$ to be accepted as $\boldsymbol{x}[t+1]$ the most often and forces the MBH to wait in place least often, contributes the most to speeding-up MBH convergence. But that begs the question: what is that shape and how does one determine or estimate it? Further, it raises question two: for a set of (f , $\boldsymbol{X}^{\mathbb{F}}$) pairs, i.e., some appropriately defined equivalence class $\mathcal{C} \equiv \{(f_i, \boldsymbol{X}_i^{\mathbb{F}})\}$, for which each ($f$, $\boldsymbol{X}^{\mathbb{F}}$) pair in the equivalence class impacts the MBH rate in a manner that is similar to other (f , $\boldsymbol{X}^{\mathbb{F}}$) pairs in the equivalence class, is there a p that is especially effective in speeding-up the MBH? To be more precise, is there a p that by its shape results in faster convergence on all $(f_i, \boldsymbol{X}_i^{\mathbb{F}}) \subset \mathcal{C}$ compared to other p, when MBH convergence speed is measured by Gamma distributions fit as FPTDs to FPTs from large numbers of trials of MBH on the various members of each class and across the different equivalence classes?

The empirical answer, based on extensive simulation experiments, is yes. That p is iso-directional, zero mean, has a very tall mid-section ("head") at its mean, and very long, thin tails (similar to the p having the histogram shown in Figure II.2, in as many dimensions as, and scaled to the span of, domain $\boldsymbol{X}$). That p is highly effective for most f – especially for f that is non-convex and/or non-smooth – and/or when $\boldsymbol{X}^{\mathbb{F}}$ is disconnected and sparse. Empirical evidence of the unusually high effectiveness of such a p compared to Gaussian-shaped and other-shaped p is documented in Englander and Englander, 2014.

If one were to colloquially explain why that shape is so effective, it is because that shape enables the MBH to follow an efficient path to f^* by frequently providing draws $\boldsymbol{\Delta x}$ that result in $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \boldsymbol{\Delta x})$ that are "accepted" as the new $\boldsymbol{x}$[t]. That shape of p provides a favorable mix

of long incremental hop distances required to "escape" from neighborhoods around local minima, and short incremental hop distances which are required to "close-in" on the global minimum in a small convex, reasonably smooth, neighborhood that surrounds the global minimum.

But the present work aims to explain this formally rather colloquially. For that purpose: sub-section IV.e will posit the existence of distribution q comprised of $\boldsymbol{\Delta x}$ that cause the $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \boldsymbol{\Delta x})$ to be "accepted" and thereby enable the MBH to more toward the global minimum rather than be forced to wait-in-place; sub-section IV.f will provide a method for constructing an estimate of the conjectured q, namely $\hat{q}$; sub-sections IV.g through IV.i will show that the similarity between the shape of p and the shape of $\hat{q}$ as measured by the the Kullback-Leibler divergence $D_{K\text{-}L}(p, \hat{q})$ correlates strongly with MBH convergence speed; and sub-sections IV.k through IV.n will explain the speed-up in MBH convergence that is achievable by adapting the shape of p to the shape of $\hat{q}$ as a realizable approximation to the non-realizable goal of adapting the shape of p to the shape of the conjectured q.

IV.e Distribution *q*

Distribution q is a joint probability distribution of the same dimension as p and $\boldsymbol{X}$ when $\boldsymbol{X}$ is multi-dimensional. Like p, q has zero mean if only because the $\boldsymbol{\Delta x} \sim q$ are on $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \boldsymbol{\Delta x})$ being "accepted" and $\boldsymbol{\xi}[t]$ is often not "accepted". Because q is defined by the condition that $\boldsymbol{\xi}[t]$ is "accepted", which depends (at least) on f and $\boldsymbol{X}^{\mathbb{F}}$, q is dependent upon f and $\boldsymbol{X}^{\mathbb{F}}$.

The existence of q is a conjecture supported by an estimate of q, namely $\hat{q}$, and the effectiveness of a method that uses $\hat{q}$ to speed-up MBH. Distribution q is not directly observable. The importance of q is that the similarity between its estimate $\hat{q}$ and p, as measured by the Kullback-Leibler divergence $D_{K\text{-}L}(p, \hat{q})$, correlates strongly with the speed of an MBH operating on f using p as the distribution from which hop distances are drawn, characterized by Gamma

distributions fit to MBH FPTs as their FPTD. Further, the speed-up of MBH that is achievable by adapting p to $\hat{q}$, as explained and demonstrated later in this chapter and in Chapter VI, provides additional support for the conjecture that q exists and is a useful concept.

The Monte Carlo method used to estimate $\hat{q}$ is described in the next sub-section, IV.f. Before explaining the Monte Carlo method for estimating $\hat{q}$ it is worth emphasizing that $\hat{q}$ is only needed to calculate $D_{K\text{-}L}(p, \hat{q})$; $D_{K\text{-}L}(p, \hat{q})$ is calculated to show the similarity or dissimilarity between p and $\hat{q}$; and the similarity or dissimilarity between p and $\hat{q}$ is important because it is predictive of the convergence speed of an MBH operating on f constrained by $\boldsymbol{X}^{\mathbb{F}}$ when using p as the distribution from which hop distances are drawn. Here, saying that $D_{K\text{-}L}(p, \hat{q})$ is predictive of the convergence speed of an MBH means that values of $D_{K\text{-}L}(p, \hat{q})$ are strongly correlated with MBH FPTDs, and MBH FPTDs characterize MBH convergence times. Thus, as explained in Sub-sections IV.f through IV.i, the Monte Carlo estimate $\hat{q}$ is required for explanatory and predictive purposes. However, the Monte Carlo estimate $\hat{q}$ is not required for the adaptation of p to $\hat{q}$; only an estimate of the scale parameter of $\hat{q}$, namely $\hat{\lambda}$, is required as explained in Sub-sections IV.l through IV.n. The challenge therein is that because MBH is a stochastic process comprised of time-varying increments, $\hat{\lambda}$ is a time-varying scale parameter $\hat{\lambda}[t]$. That is addressed in Sub-sections IV.l through IV.n, and in Chapter VI. Strictly speaking, adaptation of p to $\hat{q}$ is an adaptation of p to $\hat{\lambda}[t]$ as will be shown by an equation in Sub-section IV.n. Nonetheless, the present work refers to the adaptation of p to $\hat{\lambda}[t]$ as the adaptation of p to $\hat{q}[t]$ because the key element of the supporting analysis is $D_{K\text{-}L}(p, \hat{q})$.

The key concepts and relationships are: The conjectured the existence of q; the Monte Carlo estimation of $\hat{q}$; the measure of similarity $D_{K\text{-}L}(p, \hat{q})$; the empirical establishment that $D_{K\text{-}L}(p, \hat{q})$ correlates strongly with MBH PFT-PMFs and Gamma distributions fit to MBH FPTs,

and, thereby that $D_{K\text{-}L}(p, \hat{q})$ correlates strongly with speeding-up MBH; and that MBH is sped-up by adapting p to $\hat{q}$. The effectiveness of the resulting speed-up resulting from adapting p to $\hat{q}$ is an empirical validation of the conjectured existence of q, as well the usefulness of conjecturing the existence of q.

IV.f A Monte Carlo method for collecting $\Delta x \sim q$ in order to estimate $\hat{q}$ for use in $D_{K\text{-}L}(p, \hat{q})$

Collecting a statistically sufficient set of $\Delta x \sim q$, for the purpose of measuring the similarity of p and q by using $D_{K\text{-}L}(p, \hat{q})$, is achieved by the following method:

A time-series is generated by inserting into it Δx at each time-step t if $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \boldsymbol{\Delta x})$ is "accepted" as the new $\boldsymbol{x}[t]$ and zero otherwise. Given that $\boldsymbol{\Delta x}$ is often not "accepted" and therefore the time-series contains many zeros, sufficient statistics for obtaining a meaningful histogram of $\hat{q}$ are assured by building the time-series using the $\boldsymbol{\Delta x}$ from multiple concurrent but independent MBH processes operating on the same f and $\boldsymbol{X}^{\mathbb{F}}$, using the same p. The independence of these MBH processes is essential and their respective "accepted" $\boldsymbol{\Delta x}$ cannot be summed or averaged. For the purposes of building a histogram of $\hat{q}$, the respective $\boldsymbol{\Delta x}$ only need to be aggregated, i.e., appended. In a later sub-section, when the estimation of $\hat{q}[t]$ is addressed, it will be shown that the respective $\boldsymbol{\Delta x}$ can be input concurrently to a single low-pass filter but that the resulting temporal correlation is helpful in reducing the temporal variance in $\hat{q}[t]$ as long as the lag introduced by the low-pass filter does not cause too-slow a rate of adaption of p to $\hat{q}[t]$. However, where only the histogram of $\hat{q}$ is important, as is the case so far, the respective $\boldsymbol{\Delta x}$ are only, and must only be, aggregated without introducing dependence between them.

Figures IV.1 and IV.2 illustrate time-series comprised of "accepted" $\Delta \boldsymbol{x}$ for particular f operated upon by using a particular p. In Figures IV.1 and IV.2 $\Delta \boldsymbol{x}$ is referred to as "dx_preferred". As an idealization, q is dependent on f and $\boldsymbol{X}^{\mathbb{F}}$ but independent of p. However, $\hat{q}$ is sensitive to the

choice or design of p. Nonetheless, $\hat{q}$ is reasonably insensitive to a wide range of p when the similarity between p and $\hat{q}$ is measured by $D_{K\text{-}L}(p, \hat{q})$. Thus, the small sensitivities of $\hat{q}$ to p are a very small concern in the sub-sections that follow. Comparing the lower panels in Figures IV.1 and IV.2, respectively, shows that the MBH convergence process takes longer and is more complicated on globally rugged f compared to unimodal f, consistent with Figures II.9 and II.10, respectively.

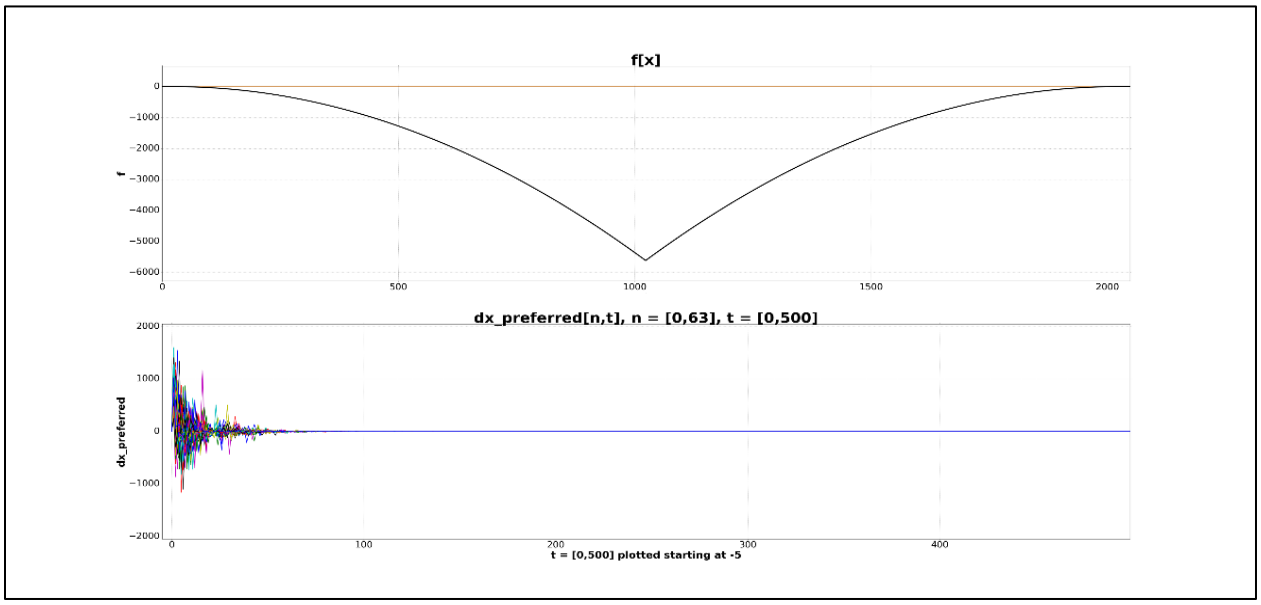


Figure IV.1:
Upper panel: Unimodal f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$
Lower Panel: $\Delta \boldsymbol{x} \sim \hat{q}$ collected from 64 concurrent but independent trials of MBH operating on the f shown in the upper panel, using a fixed p having a Laplace(0,1) shape scaled to the domain of this f

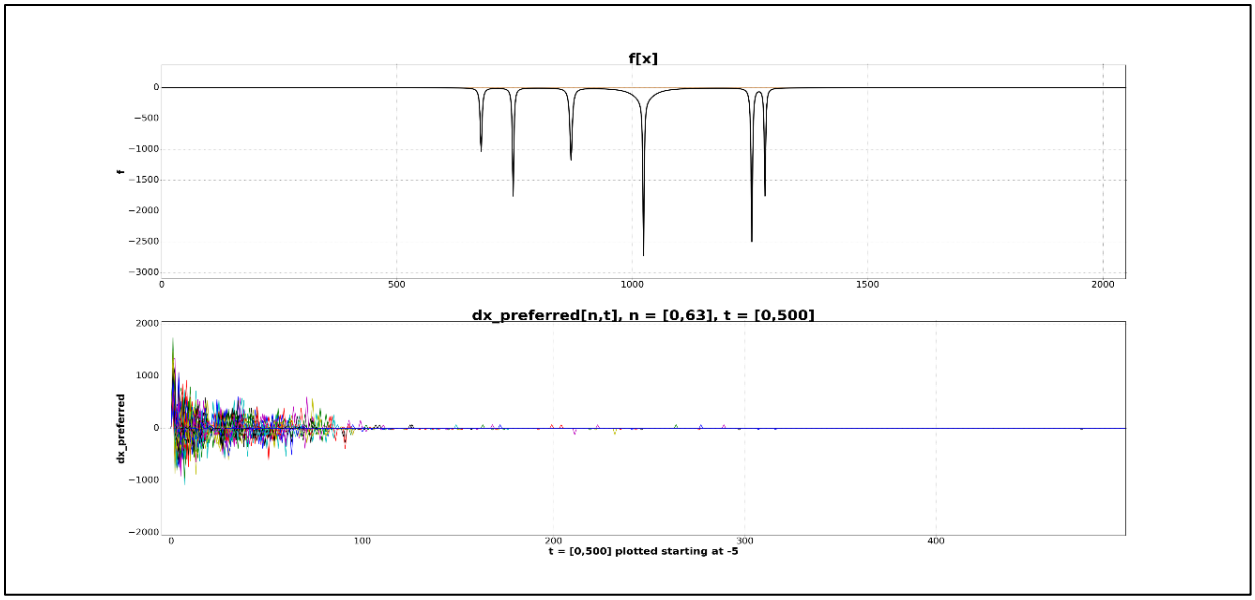


Figure IV.2:
Upper panel: Globally rugged f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$.
Lower Panel: $\Delta \boldsymbol{x} \sim \hat{q}$ collected from 64 concurrent but independent trials of MBH operating on the f shown in the upper panel, using a fixed p having a Laplace(0,1) shape scaled to the domain of this f.

IV.g The Kullback-Leibler divergence $D_{K\text{-}L}(p, \hat{q})$

The method for measuring the similarity between p and $\hat{q}$ is the Kullback-Leibler divergence $D_{K\text{-}L}(p, \hat{q})$, sometimes written as $D_{KL}(p \parallel \hat{q})$. $D_{K\text{-}L}(p,\hat{q})$ is a measure of "similarity" between two PMFs that can be obtained from pdfs or from a large amount of empirical data provided that the two PMFs are prepared in the same manner. In particular, the frequencies of occurrence for the two PMFs need to be "binned" in such a way that binned probability p_i in a PMF p covers the same range of values as binned probability q_i in a PMF q, all of the bins in q and p are equal in width, and the corresponding values they cover are aligned. Technically, non-uniform quantization is acceptable, but rarely used. In addition, the two PMFs must be normalized so that their respective frequencies of occurrence sum to 1 and no bin can contain zero occurrences because logs of the values in each bin, meaning the occurrences, will be taken [50, 51].

Then:

$$D_{K\text{-}L}(p, \hat{q}) = \sum_i p_i \left(\log_b \frac{p_i}{\hat{q}_i}\right) \text{ where } b \text{ is the base of the logarithm.}$$

The bases that are commonly used are b = 2 for measurements in bits, and b = e for measurement in units of nats. From the definition above, is clear that as p becomes more "similar" to $\hat{q}$, $D_{K\text{-}L}(p, \hat{q}) \rightarrow 0$ because $\frac{p_i}{\hat{q}_i} \rightarrow 1 \; \forall \; i$, therefore $\log_b \frac{p_i}{\hat{q}_i} \rightarrow 0 \; \forall \; i$.

$D_{K\text{-}L}(p, \hat{q})$ compares the similarities between two distributions in a manner that would otherwise require the calculation and comparison of a large number of moments. In that sense, $D_{K\text{-}L}(p, \hat{q})$ is not only effective, but also parsimonious and computationally simple as long as the probability mass functions are properly normalized, and their bins are properly constructed and aligned. By the definition of entropy used in information theory, $D_{K\text{-}L}(p, \hat{q})$ is the relative entropy between p and $\hat{q}$.

In Figure IV.3 below, p is Laplace(0,k) where k is the length of X (in this case 2,500), and $\hat{q}$ is the narrow-bodied, long-tailed, blue histogram in the middle panel generated by the "dx_preferred" time-series in the middle panel of Figure IV.2 above. The $D_{K\text{-}L}(p, \hat{q})$ is shown in the middle panel as 2.938. Since, $D_{K\text{-}L}(p, \hat{q})$ approaches zero as the shape of p approaches the shape of $\hat{q}$, $D_{K\text{-}L}(p, \hat{q})$ being 2.938 indicates that the shapes of p and $\hat{q}$ are not similar (as is evident visually).

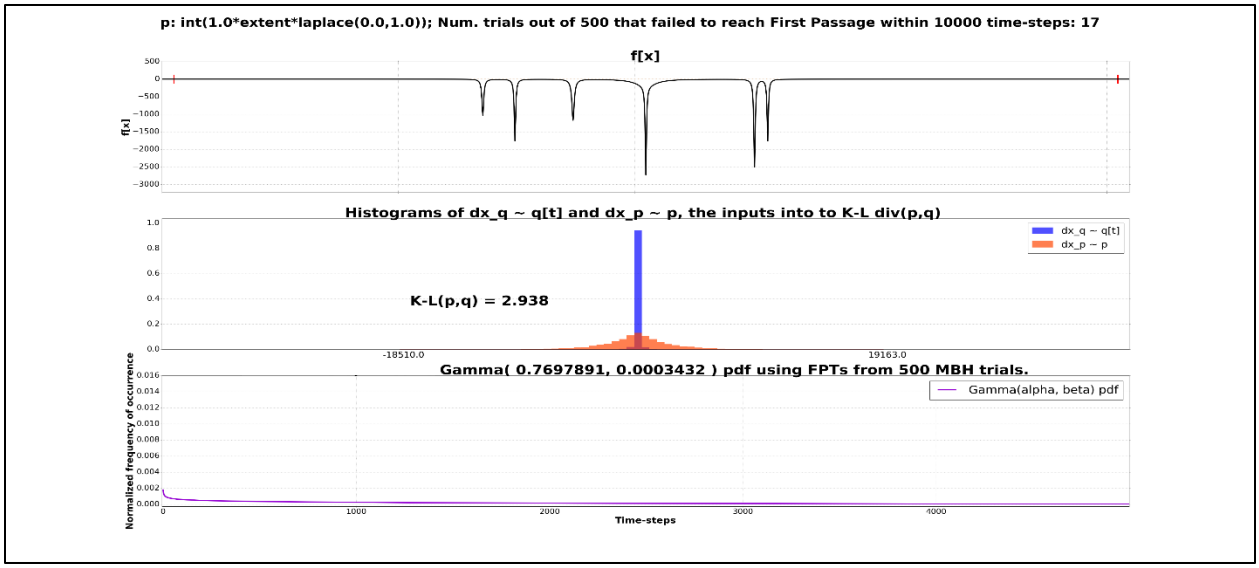


Figure IV.3:

Upper panel: Globally rugged f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$.

Middle panel: Histograms of p (orange) and $\hat{q}$ (blue) collected from a time-series (shown in the lower panel of Figure IV.2) comprised of "accepted" $\boldsymbol{\Delta x}$ from 64 concurrent but independent trials of MBH operating on the f shown in the upper panel, using a fixed p having a Laplace(0,1) shape scaled to the domain of f

Lower panel: The Gamma distribution fit to the FPTs of 500 independent trials. Later, this panel will be compared with the lower panel in Figure IV.5 to show the speed-up that is achieved by using a p that is adaptive and thereby made highly similar to $\hat{q}$.

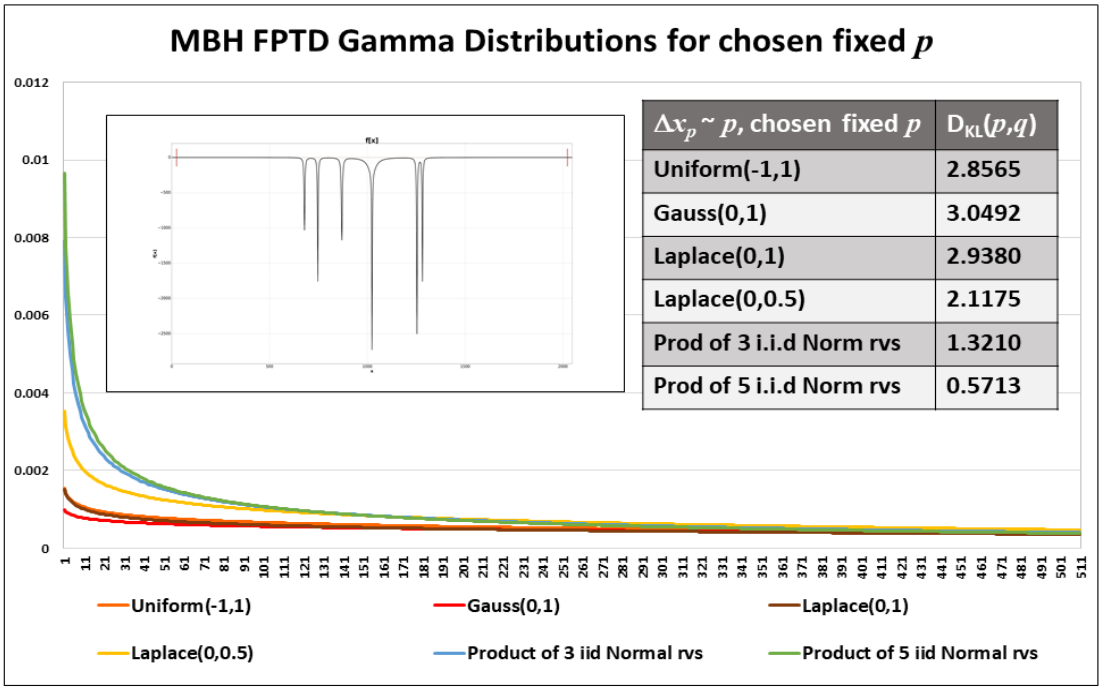
IV.h MBH FPTs and the fit of Gamma distributions to MBH FPTs as MBH FPTDs

In the present work, MBH FPTs and the fit of Gamma distributions to FPTs as FPTDs, are used to characterize the speed-up that is achievable by the methods provided in this chapter and Chapter V. In this chapter, MBH FPTs and the fit of Gamma distributions to FPTs as FPTDs, are used to characterize the speed-up of MBH that is achievable by making p being similar to $\hat{q}$, including by adapting p to $\hat{q}$. For that purpose, fits of Gamma distributions to FPTs as FPTDs are correlated with $D_{K\text{-}L}(p, \hat{q})$ to show that MBH speeds up as p is made similar to $\hat{q}$. Correlations between Gamma distributions fit to FPTs as FPTDs are also used in Chapter VI to demonstrate the speed-up of MBH that is achieved by additional methods.

IV.i The correlation between fits of MBH FPTs to a Gamma distribution as a measure MBH convergence speed, and $D_{K\text{-}L}(p, \hat{q})$ as a measure of the similarity between p and $\hat{q}$

Figure IV.4 below shows that Gamma distributions fit to MBH FPTs as FPTDs have taller modes, more area concentrated under their mode, and thinner tails, as $D_{K\text{-}L}(p, \hat{q})$ indicates greater similarity between p and $\hat{q}$. Gamma distributions fit to MBH FPTs as FPTDs that have taller modes, more area concentrated under their mode, and thinner tails, indicate fast MBH convergence and less variability in convergence times across MBH trials. In each case below, the f upon which the MBH is operating is the objective function shown in the inset in the sub-plot in Figure IV.4 below. The distribution p from which the $\Delta \boldsymbol{x}$ are drawn is a fixed distribution labeled in the inset table, distribution $\hat{q}$ is the narrow-bodied, long-tailed, blue histogram shown in the middle panel of Figure IV.3 that is characteristic of the f upon which the MBH is operating, and the numerical value of $D_{K\text{-}L}(p, \hat{q})$ is shown in the inset table. The corresponding Gamma distribution corresponding to each $(p, \hat{q})$ pair is color-coded as indicated in the legend under the plot. The figure indicates that over a range of hop length distributions p used by MBH to operate on f, the

p that were most similar to $\hat{q}$, meaning produced the smallest values of $D_{K\text{-}L}(p, \hat{q})$, sped-up the MBH convergence rate the most. In all cases, f is as shown in the left inset sub-figure and $\hat{q}$ is the narrow-bodied, long-tailed, blue histogram shown in the middle panel of Figure IV.3 comprised of the $\boldsymbol{\Delta x} \sim \hat{q}$ shown in Figure IV.2.



$\Delta x_p \sim p$, chosen fixed p	$D_{KL}(p,q)$
Uniform(-1,1)	2.8565
Gauss(0,1)	3.0492
Laplace(0,1)	2.9380
Laplace(0,0.5)	2.1175
Prod of 3 i.i.d Norm rvs	1.3210
Prod of 5 i.i.d Norm rvs	0.5713

Figure IV.4: A set of Gamma distributions fit to MBH FPTs, and their corresponding $D_{K\text{-}L}(p, \hat{q})$ for distributions p listed in the right inset table. In all cases, f is as shown in the left inset sub-figure and $\hat{q}$ is the narrow-bodied, long-tailed, blue histogram shown in the middle panel of Figure IV.3 comprised of the $\boldsymbol{\Delta x} \sim \hat{q}$ shown in Figure IV.2

IV.j Estimators of parameters used to fit FPTs to a Gamma distribution as the FPTD

The probability density function of a Gamma distribution parameterized by shape parameter α and scale parameter θ is:

$$f(x) = \frac{1}{\Gamma(\alpha)\theta^{\alpha}} x^{\alpha-1} e^{-\frac{x}{\theta}}$$

The parameter estimators used in the present work to fit N FPTs, each y MBH time-steps in length, to a Gamma distribution as their FPTD are the well-known estimators:

Shape parameter $\widehat{\alpha} = \frac{N \sum_{i=1}^{N} y_i}{N \sum_{i=1}^{N} y_i ln(y_i) - \sum_{i=1}^{N} ln(y_i) \sum_{i=1}^{N} y_i}$

Scale parameter $\hat{\theta} = \frac{1}{N^2}(N\sum_{i=1}^{N} y_i \, ln(y_i) - \sum_{i=1}^{N} y_i ln(y_i) \sum_{i=1}^{N} y_i)$

In charts that refer to an estimated rate parameter $\hat{\beta}$, $\hat{\beta} = 1/\hat{\theta}$.

See Hogg and Craig (1978), and Papoulis (1984) for an overview, and Zhi-Sheng Ye & Nan Chen (2017), and Francisco Louzada, Pedro L. Ramos, and Eduardo Ramos (2019) for a discussion of consistency and bias, including methods for correcting biases[52, 53, 54, 55]. In the present work these estimators were used without correcting for biases which, as they are defined, appear to be very small for N=1,000, or 100 (or even in one case involving 2-dimensional Gibsonian f, N=50). In addition, in the present work the Gamma distributions parameters are used for comparative purposes only.

IV.k Additional concepts and tools required to answer question three

Question three requires a few additional tools. Recall question three: given a particular unknown f and $\boldsymbol{X}^{\mathbb{F}}$, is there a method for adaptively shaping p so that MBH convergence can be sped-up? The answer is yes. By adapting p to $\hat{q}$ in such a manner that p is made similar to $\hat{q}$, the speed-up of MBH convergence can be achieved. As will become clear in Sections IV.l, IV.m and

IV.n, adapting p to $\hat{q}$ does not require an estimate of $\hat{q}$ but rather only its time-varying scale parameter $\hat{\lambda}[t]$ which is time-varying for reasons explained in Sections IV.m and IV.n.

Until question three was asked by the present author, practitioners had little motivation to pursue an analytic answer to question two because the empirical results published in 2014 were all that were needed to achieve very effective implementations. However, once question three was asked, and such adaptive strategies were demonstrated to yield large improvements in MBH convergence rates as shown in the figures at the end of this chapter and in Chapter VI, the notions of q, $\hat{q}$, measuring the similarity between p to $\hat{q}$, and adapting p to $\hat{q}$, became practically as well as analytically important.

By describing the nature of distribution q comprised of hop lengths $\boldsymbol{\Delta x}$ that resulted in "accepted" $\boldsymbol{\xi}[t]$, a Monte Carlo method for estimating q, namely $\hat{q}$, and the Kullback-Leibler divergence $D_{K\text{-}L}(p, \hat{q})$ used to measure the similarity between p and $\hat{q}$, question three can be answered. Because the notion of q, the Monte Carlo method for estimating $\hat{q}$, and the use of $D_{K\text{-}L}(p, \hat{q})$ are general, question three can be answered generally rather than specific to the any example or the use-case in the present work. Nonetheless, once question three is answered in general, and its answer is used to explain the adaptation of p to $\hat{q}$, specific examples of the speed-up of MBH convergence achieved by adapting p to $\hat{q}$ are demonstrated using simulation experiments.

Although $D_{K\text{-}L}(p,\hat{q})$ is used to measure the similarity between p and $\hat{q}$, the MBH speed-up achieved by adapting p to $\hat{q}$ only involves adapting the scale parameter of p to $\hat{\lambda}$ within bounds that are required to assure that p will always retain some minimal probability of generating large increment hop distances. In the present work, $\hat{q}$ is modeled as being isotropic (therefore having equal marginal distributions in each coordinate random variable) and comprised of $\boldsymbol{\Delta x}$ that are

independent across dimensions. Therefore, the adaptation of p to $\hat{q}$ involves a single $\hat{\lambda}$ that is used in all dimensions. The iso-directionality of the scale parameters of p and $\hat{q}$ can easily be generalized, but that is not done so here. It can be argued that anisotropies in realistic f and $\boldsymbol{X}^{\mathbb{F}}$ would give rise to non-iso-directional scale parameters, therefore a vector rather than scalar-valued $\hat{\lambda}$, but the non-iso-directionality of $\hat{q}$ and therefore p was not investigated in the present work.

IV.l Adapting p to $\hat{q}$

The adaptation of p to $\hat{q}$ requires timely estimates of $\hat{q}$ without which the adaptation of p to $\hat{q}$ often cannot be achieved. Therefore, before explaining the adaptation of p to $\hat{q}$, timely estimates of $\hat{q}$ are discussed.

Effective estimates of $\hat{q}$ can be achieved by estimating only the scale parameter of $\hat{q}$, namely $\hat{\lambda}$. But parameter $\hat{\lambda}$ is dependent on x[t] and f[$\boldsymbol{x}$[t]] and their respective proximity to $\boldsymbol{x}$* and f*. Therefore, $\hat{\lambda}$ is time-varying and is written as $\hat{\lambda}[t]$. In that case, $\hat{q}$ is written as $\hat{q}[t]$. Because $\hat{\lambda}[t]$, and therefore the shape of $\hat{q}$, is time-varying, the adaptation of p to $\hat{q}[t]$ requires a timely estimation of $\hat{\lambda}[t]$. Indeed, the benefit of adapting p to $\hat{q}$ is that p is made suitable to f in a timely manner, in approximate accordance with f[$\boldsymbol{x}$[t]], rather than in a way that is a compromise with the respect to the p that is most suitable for the search overall. Thus, for purposes of adapting p to $\hat{q}$, methods that rapidly estimate $\hat{\lambda}[t]$ are more effective than methods that estimate $\hat{\lambda}[t]$ slowly, especially as $\boldsymbol{x}$[t] approaches $\boldsymbol{x}$* or f[$\boldsymbol{x}$[t]] approaches f*. One implementation for estimating $\hat{\lambda}[t]$ is based on a moving-window variance estimator, described immediately below. The implementation used in the Pioneer 11 trajectory optimization case use-case is provided in Chapter VI.

IV.m Estimation of $\hat{\lambda}[t]$, thereby $\widehat{q[t]}$, based on a moving-window variance estimator

The estimation of $\hat{\lambda}[t]$ based on a moving-window variance estimator involves taking the running variance of the time-series of the $\Delta \boldsymbol{x}[t]$ that result in the "acceptance" of $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \Delta \boldsymbol{x})$ as the new $\boldsymbol{x}[t]$. However, whereas taking the running variance of a time-series requires a history of some length $[(t\text{-}\tau): t]$, the variance of $\Delta \boldsymbol{x}[t]$ and therefore the scale parameter of $\hat{\lambda}[t]$ are dependent on $f[\boldsymbol{x}[t\text{-}1]]$, $\boldsymbol{X}^{\mathbb{F}}$ at $\boldsymbol{x}[t\text{-}1]$, and $\boldsymbol{x}[t\text{-}1]$. Therefore short-term $\hat{\lambda}[t]$, meaning $\hat{\lambda}[t]$ estimated over $t\text{-}\tau_1$ MBH time-steps, is different from long-term $\hat{\lambda}[t]$, meaning $\hat{\lambda}[t]$ estimated over $t\text{-}\tau_2$ MBH time-steps, for τ_2 significantly larger than τ_1. In practice, estimating $\hat{\lambda}[t]$ as a surrogate for $\hat{q}[t]$ and adapting p to $\hat{q}[t]$ thereby, are most effective when they are implemented in near-real time and the width of the temporal sliding window length $[(t\text{-}\tau): t]$ is kept narrow.

However, if f is highly non-convex or non-smooth, the running estimates of the variance of time-series of $\Delta \boldsymbol{x}$, therefore $\hat{\lambda}[t]$, need to be low pass-filtered because a very noisy $\hat{\lambda}[t]$ may de-stabilize the algorithm that adapts p to $\hat{q}$. That low-pass filter introduces further delay in the estimation of $\hat{q}$ and therefore to the adaptation of p to $\hat{q}$. Thus, there is an engineering trade-off in the design of the low-pass filter: Too much low-pass filtering may reduce the effectiveness of adapting p to $\hat{q}$, especially as $\boldsymbol{x}[t]$ approaches $\boldsymbol{x}^*$, at which time the shape (scale parameter) of $\hat{q}$ may be changing rapidly and adapting p quickly may provide the most benefit; too little low-pass filtering may result in such a noisy $\hat{\lambda}[t]$ that it makes the adaptation process less stable and effective. Fortunately, in practice, the tuning of the low-pass filter is not critical or difficult.

An algorithm for adapting p to $\hat{q}$ is empirically shown to be highly effective in Figure IV.5 of this chapter, less effective in Figures IV.10 and IV.12 of this chapter, and highly effective in the Pioneer 11 use-case in Chapter VI. Adapting p to $\hat{q}$ is not effective on the 1-dimensional

example of Gibsonian *f* shown in Figures III.2 and III.3, but methods that are very effective for speeding-up MBH operating on Gibsonian *f* are provided in Chapter V.

IV.n Adaption of p to $\hat{q}$ after a low-delay estimate of $\hat{q}[t]$ is formed

After a low-delay estimate of $\hat{\lambda}[t]$ is formed according to the first and second the equations below, $p[t]$ is controlled according to the third equation below.

$$\psi[t] = \sigma(y[t-\tau{:}\,t]) \text{ for } t > \tau,$$

where y is the collection of Δx s.t. $\boldsymbol{\xi}[t] = (\boldsymbol{x}[t\text{-}1] + \boldsymbol{\Delta x})$ was "accepted"; and σ is the running standard deviation of the $y[t-\tau{:}\,t]$ in the sliding temporal window $[t-\tau{:}\,t]$.

$$\hat{\lambda}[t] = ((1\text{-}a)\cdot(\psi[t]) + (a\cdot \hat{\lambda}[t-1])$$

$$p[t] = \mathrm{k}\cdot(\,(1-b)\cdot \mathrm{Laplace}(0,(\mathrm{c}\cdot\hat{\lambda}[t-1])) + (b\cdot\mathrm{Laplace}(0,1)\,),$$

where k = *len*(***X***)/2; Laplace(0,*s*) is a Laplace distribution of mean = 0 and scale factor = *s*; and len(X) is the span of ***X*** in one dimension assuming that all dimensions of ***X*** have equal spans or that X and *f* are rescaled so that all rescaled dimensions of ***X*** have equal spans. Typically, at least one of these two assumptions are true practice. Distribution $p[t]$ is an additive mixture model comprised of two Laplace distributions. The proportionality of the weighting of Laplace distribution having the time-varying scale factor, to the Laplace distribution having the fixed scale factor, controlled by *c*, is typically 0.95:0.05. The purpose of the Laplace distribution having the fixed scale factor is to assure that long hops have at least a small probability of occurring despite the adaptation process. Parameter c controls the gain of the effect of $\hat{\lambda}[t-1]$. $\hat{\lambda}[t-1]$ is used rather than $\hat{\lambda}[t]$ because the point in each MBH iteration at which $p[t]$ is calculated and used is prior to the calculation of $\hat{\lambda}[t]$.

In practice, the tuning of parameters is not critical except that when the adaptation of p to $q[t]$ is applied to an MBH operating on a highly non-convex and or non-smooth *f*, and/or a highly

disconnected, sparse $\boldsymbol{X}^{\mathbb{F}}$, the tuning of parameter a can be important. Nonetheless, typically a is approximately 0.9 to 0.95 and the user can easily tune a using a small amount of experimentation. For the most part, the adaptation of p to $\hat{q}[t]$ is either very effective in speeding up MBH on a given f and $\boldsymbol{X}^{\mathbb{F}}$ without careful tuning of parameters, or a method provided in Chapter V should be used instead or in addition.

An example of the speed-up of MBH convergence that results from the adaptation of p to $\hat{q}[t]$ is illustrated by comparing Figure IV.3 above to Figure IV.5 below. In addition, a similar demonstration will be seen in Chapter VI when comparing Figure VI.4 to Figure VI.5. In Figure IV.5, $\hat{q}[t]$ was estimated using 16 concurrent but independent hoppers while the MBH operated on f driven by the adaptive p. This raises the question of whether feedback introduced by the way in which adaptive p "excited" the "accepted" $\boldsymbol{\Delta x}$ differently than was done by the fixed Laplace p, and thereby biased the estimate of $\hat{q}[t]$. While this was not investigated systematically, it appeared not to have any affect in practice to the speed-up of MBH enabled by the adaptation of p to $\hat{q}[t]$. Note that, whereas in the middle panel of Figure IV.3 where $D_{K\text{-}L}(p,\hat{q})$ was 2.938, $D_{K\text{-}L}(p,\hat{q})$ is now 0.1969 indicating that p and $\hat{q}$ are now much more similar. In the lower panel of Figure IV.5 the very tall left-hand side of this Gamma distribution and the fact that most of the probability density is now amassed very close to the mode, indicates that convergence was sped-up significantly adapting p to $\hat{q}[t]$.

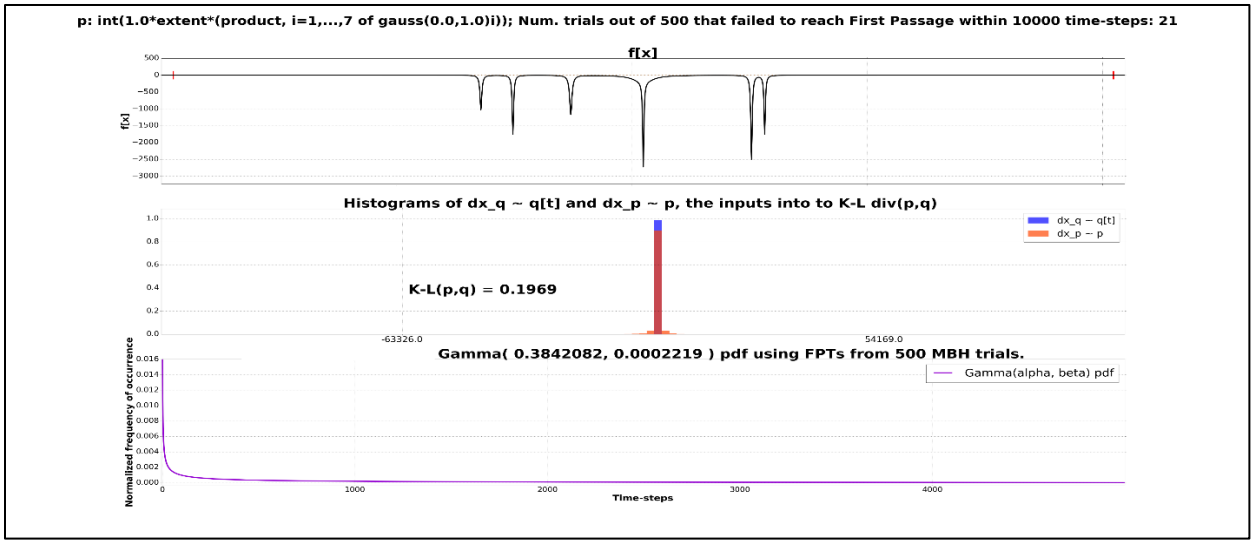


Figure IV.5:
Upper panel: Globally rugged f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$.
Middle panel: Histograms of p (orange) adapted to $\widehat{q[t]}$ (blue) collected from a time-series (shown in the lower panel of Figure IV.2) comprised of "accepted" $\boldsymbol{\Delta x}$
Lower panel: The Gamma distribution fit to the FPTs of 500 independent trials. Compare to the lower panel of Figure IV.3.

The effectiveness of adapting p to $\hat{q}[t]$ can also be demonstrated using the prototypical 1-dimensional f shown in Figure 0.a. First, the histogram of a fixed Gaussian p is shown in order to make comparisons of MBH convergence speed when the MBH is operating on a f using a fixed Gaussian p, a fixed non-Gaussian p having the histogram shown in Figure II.4, and an adaptive p.

In order to make comparisons of MBH convergence times given different p, a fixed Gaussian p is used along with the fixed non-Gaussian p having the histogram depicted in Figure II.4, as well as (by definition) f-specific adaptive p. Figure IV.6 depicts the histogram of the fixed Gaussian p.

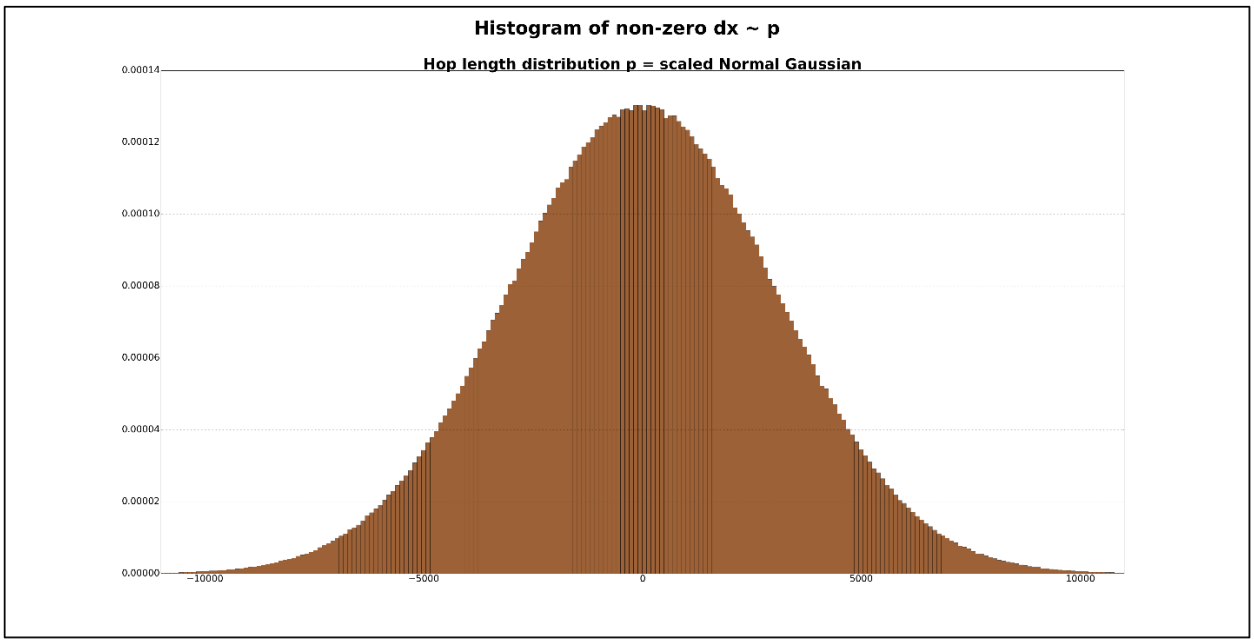


Figure IV.6: Histogram of fixed Gaussian p

Figure IV.7 below shows, in the top panel, prototypical *f* for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$, upon which MBH operated using the fixed Gaussian *p* illustrated in Figure IV.6; in the middle panel, $f[\boldsymbol{x}[t,n]]$ indicating the progress of the MBH in 48 independent trials; and, in the lower panel, a Gamma distribution fit to the 2,500 FPTs as their FPTD. A maximum of 50,000 MBH time-steps per trial was used to assure the convergence of every trial because MBH operating on prototypical *f* using fixed Gaussian *p* is inherently slower than MBH operating on prototypical *f* using fixed non-Gaussian *p*. In Figure IV.7 it is evident that all 48 trials shown converged to the global minimum within 1,600 MBH time-steps

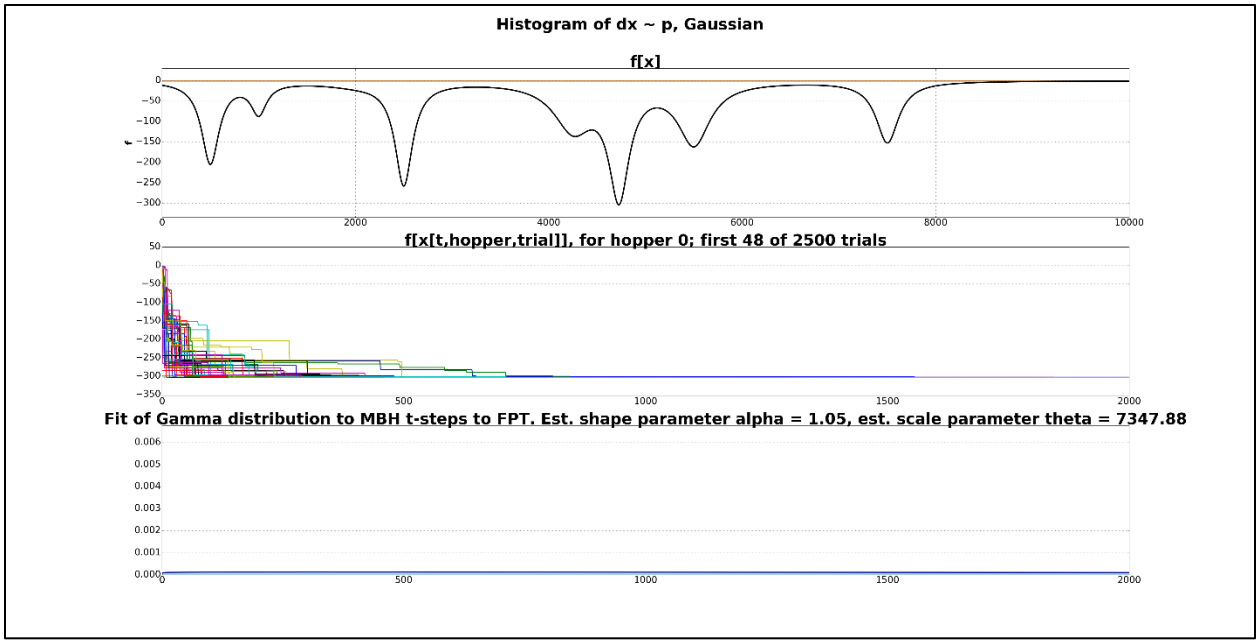


Figure IV.7:
Upper panel: Prototypical *f* for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$, upon which MBH operated using the fixed Gaussian *p* illustrated in Figure IV.6.
Middle panel: *f*[$\boldsymbol{x}$[*t*,*n*]] for *t* being the first 2,000 of 50,000 MBH time-steps, *n* being the first 48 of 2,500 trials.
Bottom panel: $\Gamma(\alpha,\theta)$ fit as the FPTD of the 2,500 FPTs

Figure IV.8 below shows, in the top panel, prototypical *f* for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$, upon which MBH operated using the fixed non-Gaussian *p* illustrated in Figure II.4; in the middle panel, *f*[$\boldsymbol{x}$[*t*,*n*]] indicating the progress of the MBH in 48 independent trials; and, in the lower panel, a Gamma distribution fit to the 2,500 FPTs as their FPTD. A maximum of 50,000 MBH time-steps per trial was used to assure the convergence of every trial because MBH operating on prototypical *f* using fixed Gaussian *p* is inherently slower than MBH operating on prototypical *f* using fixed non-Gaussian *p*. In Figure IV.8 it is evident that all 48 trials shown converged to the global minimum within 650 MBH time-steps (as opposed to in Figure IV.8 using a fixed Gauusian *p*, which required 1,600 MBH time-steps on the same *f*).

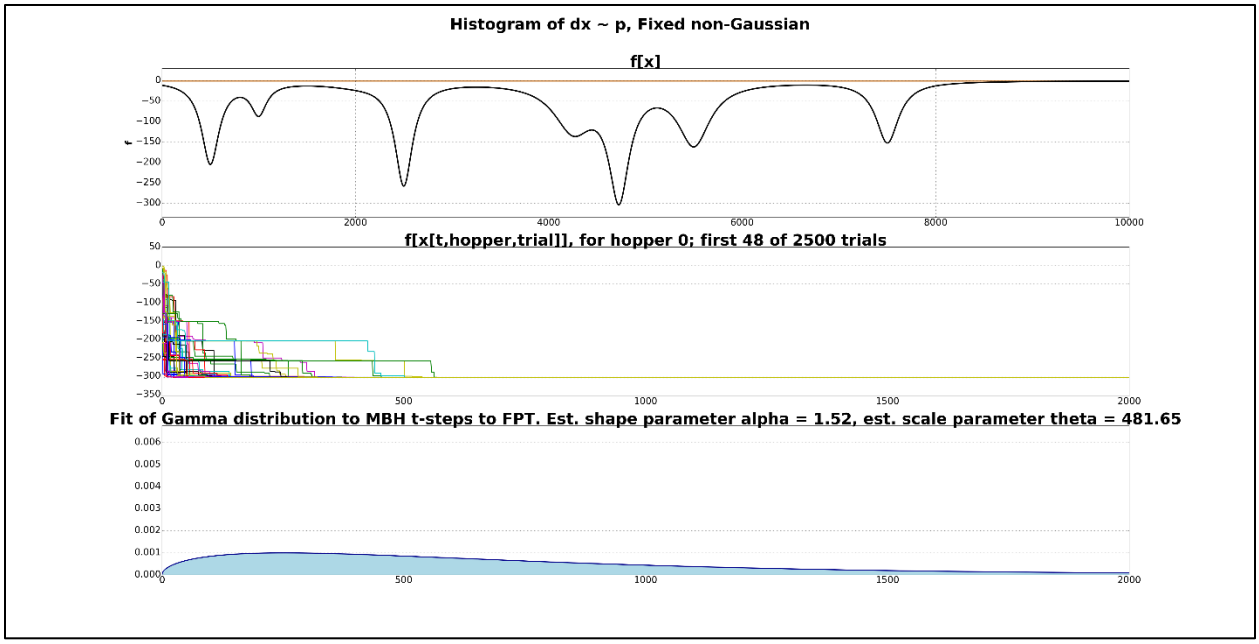


Figure IV.8:
Upper panel: Prototypical f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$, upon which MBH operated using the fixed non-Gaussian p illustrated in Figure II.4.
Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 2,500 trials
Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 2,500 FPTs

Figure IV.9 below shows, in the top panel, prototypical f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$, upon which MBH operated using adaptive p; in the middle panel, $f[\boldsymbol{x}[t,n]]$ indicating the progress of the MBH in 48 independent trials; and, in the lower panel, a Gamma distribution fit to the 2,500 FPTs as their FPTD. A maximum of 50,000 MBH time-steps per trial was used to assure the convergence of every trial because MBH operating on prototypical f using fixed Gaussian p is inherently slower than MBH operating on prototypical f using fixed non-Gaussian p. In Figure IV.9 it is evident that all 48 trials shown converged to the global minimum within 650 MBH time-steps, and all but 3 of the 48 trials converged within 300 MBH time-steps. The improvement between the use of fixed non-Gaussian p and adaptive p is evident in the scale parameter theta of the Gamma-fit for the

adaptive p (211.95) vs. fixed non-Gaussian p (481.65). The large improvement between the result of using fixed Gaussian p (Figure IV.6), compared to both fixed non-Gaussian p (Figure II.4) and adaptive p, indicates that Gaussian p is a poor shape choice for an MBH operating on textured prototypical f having a disconnected, sparse feasible domain.

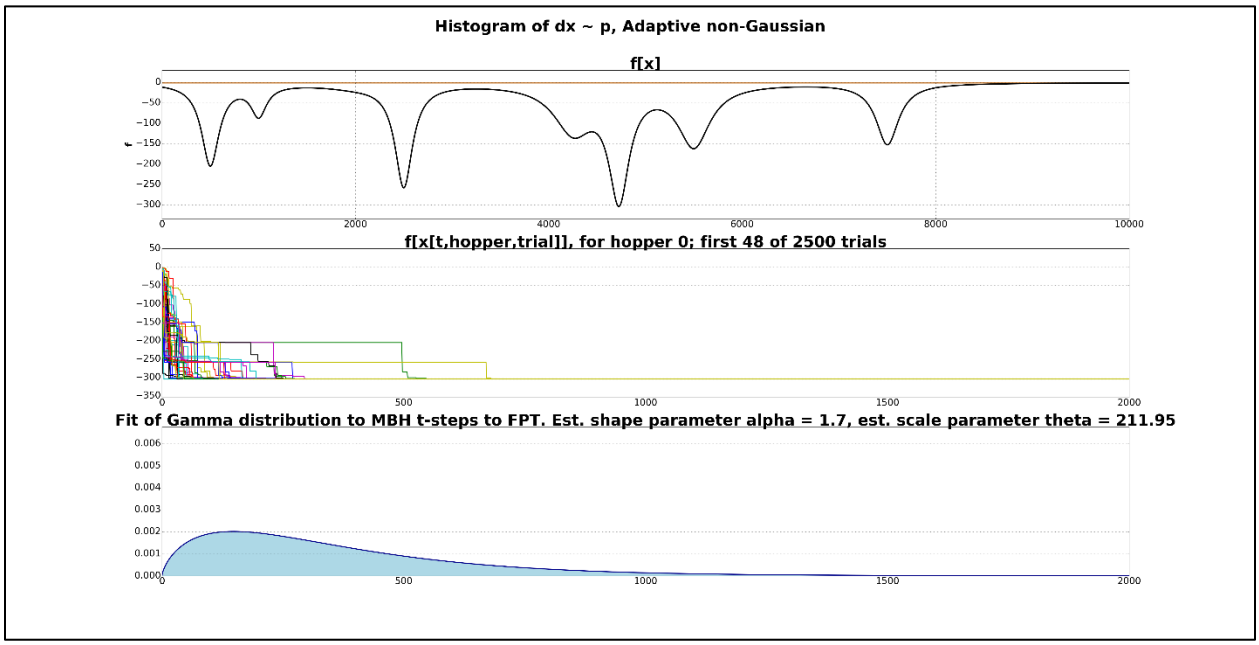


Figure IV.9:
Upper panel: Prototypical f for which $\boldsymbol{X}^{\mathbb{F}}$ is all of $\boldsymbol{X}$, upon which MBH operated using adaptive p
Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 2,500 trials
Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 2,500 FPTs

Likewise, the effectiveness of a well-chosen p, or adapting p to $\hat{q}[t]$, when the MBH is operating on the 1-dmensional example of textured prototypical f with a disconnected and sparse feasible domain is illustrated below.

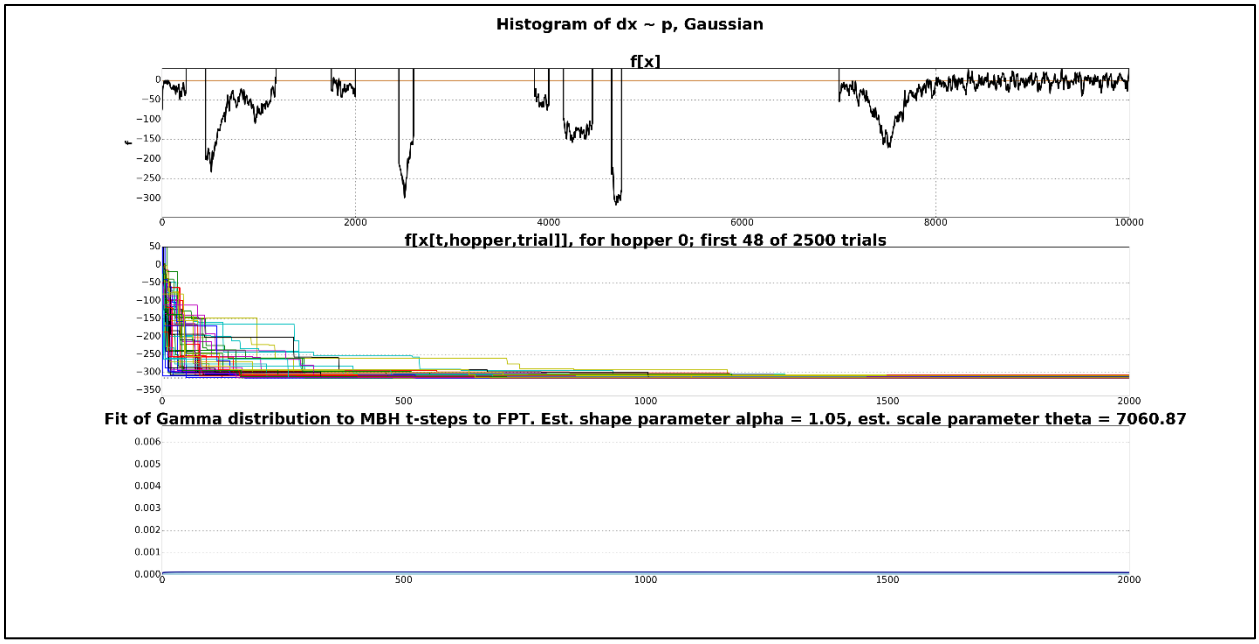


Figure IV.10:

Upper panel: Textured prototypical f having a disconnected and sparse feasible domain, upon which MBH operated using the fixed Gaussian p illustrated in Figure IV.6

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 50,000 MBH time-steps, n being the first 48 of 2500 trials

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 2,500 FPTs

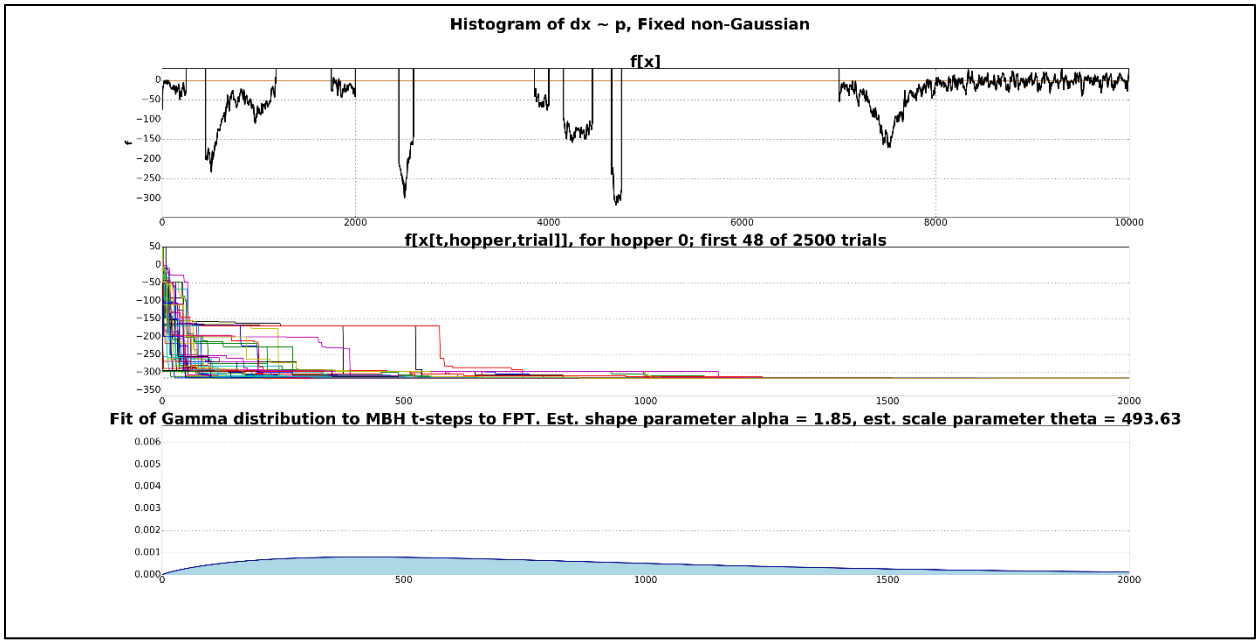


Figure IV.11:

Upper panel: Textured prototypical f having a disconnected and sparse feasible domain, upon which MBH operated using the fixed non-Gaussian p illustrated in Figure II.4

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 2500 trials

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 2,500 FPTs

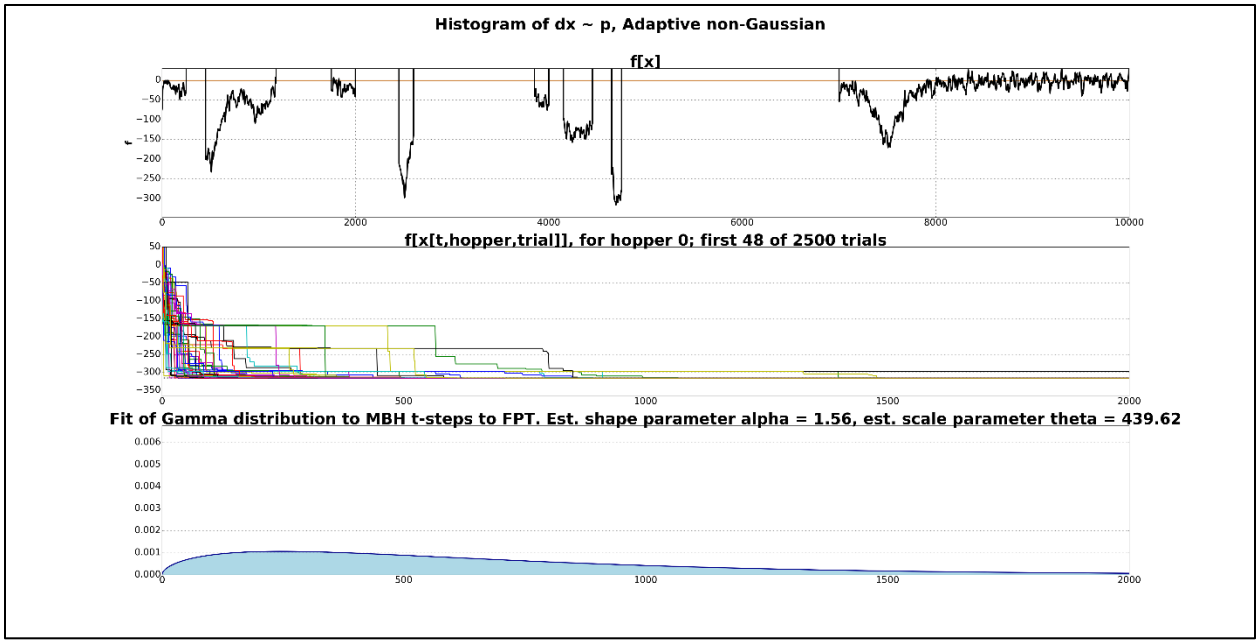


Figure IV.12:
Upper panel: Textured prototypical f having a disconnected and sparse feasible domain, upon which MBH operated using adaptive p
Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 2500 trials
Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 2,500 FPTs

Finally, it can be shown that when the MBH is operating on the 1-dimensional example of Gibsonian f, using a well-chosen p or adapting p to $\widehat{q[t]}$, does not by itself speed-up the MBH effectively. That is because, when operating on Gibsonian f, regardless of the shape of p, the probability of $\boldsymbol{x}[t]$ being and remaining within the wide basin containing the local minimum is much higher than the probability of $\boldsymbol{x}[t]$ being within or hopping to the narrow, distant basin containing the global minimum. Therefore, at each MBH time-step, the probability of drawing an $\boldsymbol{\Delta x}$ that will cause $\boldsymbol{x}[t]$ find its way into the narrow basin and then travel down to f^* is small. The reason that adaptive p is particularly ineffective in speeding-up MBH operating on Gibsonian f is that the wide, smooth, locally convex basin containing the nearly as deep local minimum causes

faulty (mis-timed) estimates of the scale parameter of $\hat{q}[t]$ and, as a result, a mal-adaption of $p[t]$ to $\hat{q}[t]$.

However, the ineffectiveness of speeding up an MBH operating on the 1-dimensional example of Gibsonian by biasing the shape of the hop length distribution p motivates Chapter V where a different method for speeding up MBH convergence on Gibsonian f is shown to be highly effective.

IV.o The special case of adapting p to $\hat{q}$ when $\min_{\text{global}}(f)$ is known to be small and non-negative

In many applications the physics of the problem are such $f[\boldsymbol{x}[0]]$ for almost any randomly chosen $\boldsymbol{x}[0]$ in $\boldsymbol{X}^{\mathbb{F}}$is large but $\min_{\text{global}}(f)$ is known to be small and non-negative (e.g., when f is energy or the sum or integral of change in velocity corresponding to the consumption of propellant). In such cases, the adaptation of p to $\hat{q}$ can be simplified as follows: Assume the existence of a distribution q such that $\Delta x \sim q$. $\hat{q}[t]$ is estimable from the $\Delta x[t]$. As in sections IV.m and IV.n, the goal is to match the scale parameter of adapted $p[t]$ to the scale parameter of $\hat{q}[t]$, but in this case one can use $f[\boldsymbol{x}[i,t]]$ of $i = 1, 2, 3, \ldots, I$ concurrent hoppers (whether they are operating independently or cooperating in a manner that will be described as Multiple Communicating Hoppers in Chapter V). $I > 1$ hoppers are used simply to establish needed statistics regarding $f[\boldsymbol{x}[i,t]]$. $I > 4$ is used to reduce the variance in the needed statistics. In most applications, $I > 16$ is unnecessary. This case of adapting p to $\hat{q}$ applies to, and was employed in, the use-case in Chapter VI.

Mean($f[:,t]$), where : is across the i concurrent hoppers, is their mean incumbent value of f at MBH time-step t. Max($f[:,t]$), where : is across the i concurrent hoppers, is their max incumbent value of f at MBH time-step t. Max($f[:,0]$), where : is across the i concurrent hoppers, is their max

initial incumbent value of f at initial MBH time-step t_0. $s[t]$ = mean(f [x[:,t-1]])/max(f [x[:,0]]); $s[t] \rightarrow \varepsilon$ as mean(f [x[:,t]]) → min(f). $\lambda[t] = (a \cdot s[t])+((1-a) \cdot 1.0)$; where $a \cong 0.8$ and $a<1$ minimizes Prob(detrimental adaptation process) by assuring that the scale parameter of the distribution generating the hops never goes to zero. $p[t] = p_\lambda(0, \lambda[t])$ where $p_\lambda(0, \lambda[t])$ is the non-Gaussian distribution illustrated in Figure II.4 but with variable scale parameter $\lambda[t]$. Figures IV.13, IV.14, and IV.15 illustrate the use of this special case of the adaptation of p to $\hat{q}$. Figure IV.13 shows MBH operating on an f using a fixed Laplace(0,1) p that is not similar to $\hat{q}$. Figure IV.14 shows MBH operating on the same f using a fixed p_λ (fixed in the sense that λ in $p_\lambda(0, \lambda)$ is time-invariant). Figure IV.15 shows MBH operating on the same f using $p_\lambda(0, \lambda[t])$ adapted to $\hat{q}[t]$, where $\lambda[t]$ was generated as described above. In Figure IV.15, 16 independent hoppers are used (I = 16).

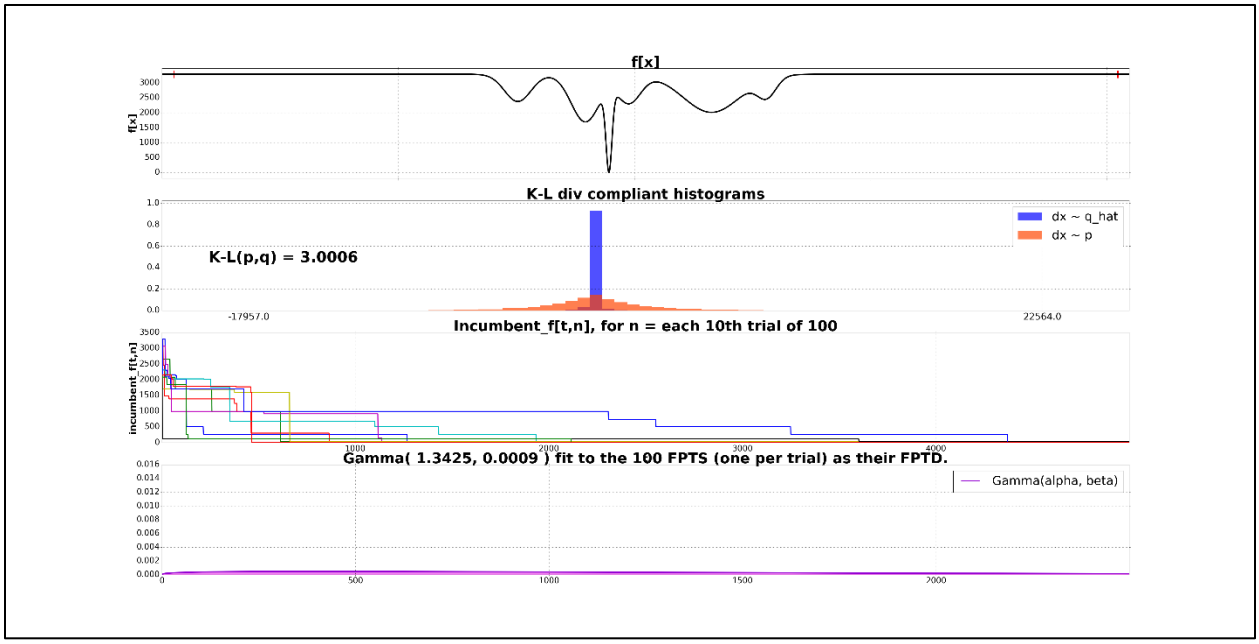


Figure IV.13:

Upper panel: f upon which MBH operated using a fixed Laplace(0,1) p

Second panel: $D_{K\text{-}L}(p,\hat{q})$ where $\hat{q}$ is written as q only because of limitations in graphics fonts

Third panel: $f[\boldsymbol{x}[t,n]]$ for t being 5000 MBH time-steps, n being every 10^{th} of 100 trials

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 100 FPTs

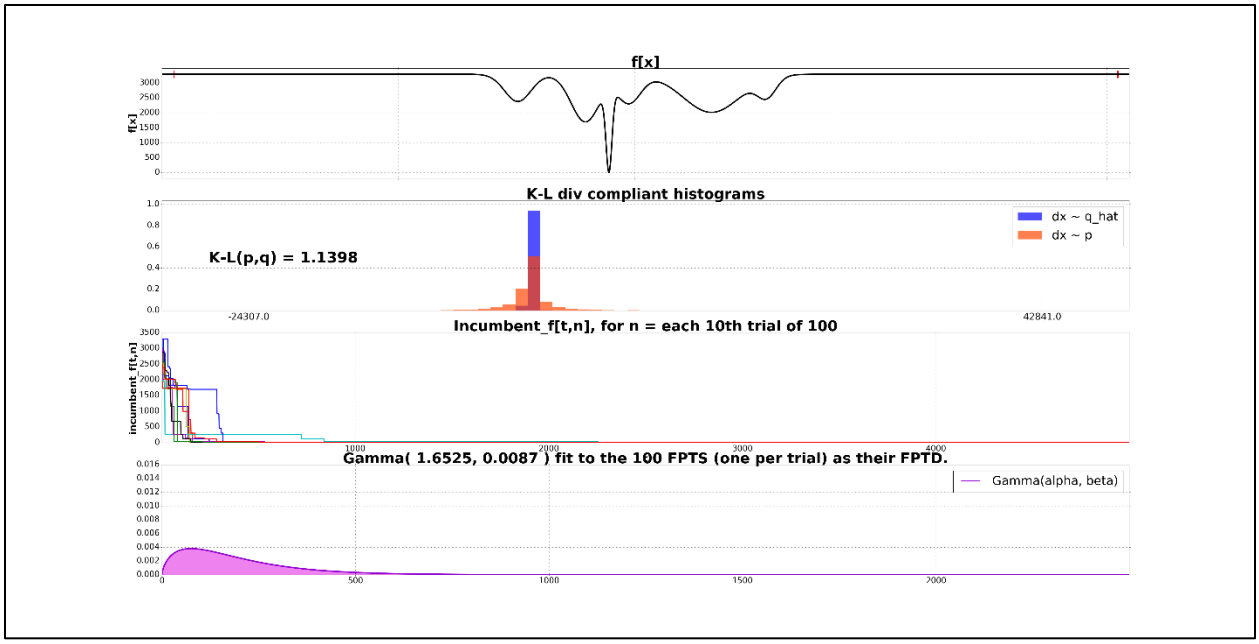


Figure IV.14:

Upper panel: f upon which MBH operated using a fixed p_λ

Second panel: $D_{K\text{-}L}(p,\hat{q})$ where $\hat{q}$ is written as q only because of limitations in graphics fonts

Third panel: $f[\boldsymbol{x}[t,n]]$ for t being 5000 MBH time-steps, n being every 10th of 100 trials

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 100 FPTs

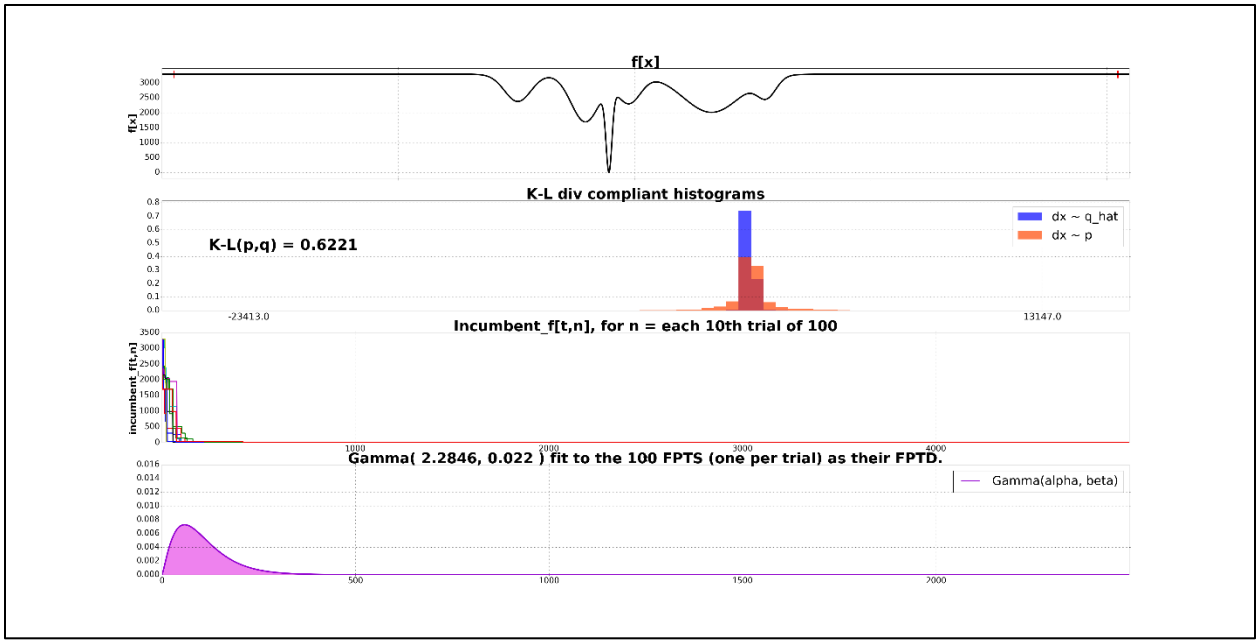


Figure IV.15:
Upper panel: f upon which MBH operated using $p_\lambda(0, \lambda[t])$ adapted to $\hat{q}[t]$
Second panel: $D_{K\text{-}L}(p,\hat{q})$ where $\hat{q}$ is written as q only because of limitations in graphics fonts
Third panel: $f[\boldsymbol{x}[t,n]]$ for t being 5000 MBH time-steps, n being every 10th of 100 trials
Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 100 FPTs

IV.p Chapter Summary

The central theme of this chapter is that whereas p, the distribution from which the $\boldsymbol{\Delta x}$ are drawn, is chosen or designed by the MBH engineer, or is shape-adapted by a method described in this chapter, distribution q, which is comprised of $\boldsymbol{\Delta x}$ that caused candidate hops $\boldsymbol{\xi}[t]$ to be accepted as the next incumbent hop $\boldsymbol{x}[t]$, is a property of f , $\boldsymbol{X}^{\mathbb{F}}$ and the path $\boldsymbol{x}$[0:t-1]. Therefore, while p is known by construction, q is an abstraction. At best, an approximation to q, namely $\hat{q}$ can be estimated using a Monte Carlo method. However, this chapter has shown that to the extent that the shape of p is similar to q as measured by the Kullback-Leibler divergence $D_{K\text{-}L}(p, \hat{q})$, the convergence of the MBH operating on an f and $\boldsymbol{X}^{\mathbb{F}}$ will be sped-up. Moreover, it was explained that $\hat{q}$ is time-varying and so any choice or design of a fixed p is at best a compromise. With that

in mind, a method for adapting p to $\hat{q}$ is provided as a way of making the shape of p similar to the shape of $\hat{q}[t]$ and thereby speed-up the convergence of MBH. The limited effectiveness in speeding-up MBH on very poorly-behaved f by (or only by) biasing the shape of the hop length distribution q motivates Chapter V where other highly effective methods for speeding-up MBH operating on very poorly-behaved f – including Gibsonian f – are provided.

IV.q Answers to the three questions with which this chapter began, and the tools used

This chapter began with three questions. In order to answer these questions, new tools were added to the analytic framework. The following table lists each of the three questions, summarizes their answers, and lists the concepts and tools that were added to the analytical framework to develop those answers.

TABLE I

Question	Answer	Concept or tool used
Given an f and $\boldsymbol{X}^{\mathbb{F}}$, what is the impact of the shape of distribution p on the MBH convergence time?	Empirically, it has been shown that that the shape of p directly impacts the expected efficiency of the MBH search and therefore the MBH convergence time. This was first reported in Englander and Englander (2014). Intuitively, it is understandable that p must provide a sufficient number of long hop distances in order to escape from basins containing a local minimum but not the global minimum, and short hop distances in order to converge to the global minimum as the MBH approaches it. The formalization of that intuition requires Questions and Answers 2 and 3, and Appendix C.	Expected MBH search efficiency as defined in this chapter
For a set of f and $\boldsymbol{X}^{\mathbb{F}}$, e.g., some appropriately defined equivalence class $\mathcal{C} \equiv \{(f_i, \boldsymbol{X}_i^{\mathbb{F}})\}$ that have similar local minima structure, is there a p that is universal in the sense that its use results in faster convergence on all $(f_i, \boldsymbol{X}_i^{\mathbb{F}}) \subset \mathcal{C}$ compared to different p?	Yes. An iso-directional p having a zero mean, a very tall mid-section ("head") at its mean, and very long, thin tails, is effective for a wide range of f and especially for f that is non-convex and/or non-smooth, and/or if $\boldsymbol{X}^{\mathbb{F}}$ is disconnected and sparse	• Conjecture of the existence of distribution q • Estimate $\hat{q}$ • $D_{K\text{-}L}(p, \hat{q})$ as a measure of similarity between p and $\hat{q}$ • MBH FPTs and fitting of FPTs to Gamma to FPTs as their (as MBH FPTDs) • Use of FPTDs to measure the speed-up of MBH that is achieved by making p is similar to $\hat{q}$
Given a particular unknown f and $\boldsymbol{X}^{\mathbb{F}}$, is there a method for adaptively shaping p so that MBH convergence can be sped-up?	Yes. The adaptation of p to $\hat{q}[t]$	• Timely estimate of $\hat{q}[t]$ • Methods for adapting p to $\hat{q}[t]$ and assumptions made in constructing those methods

V. ACCELERATING MBH CONVERGENCE BY BIASING THE "HOP FROM" LOCATION

This chapter is complementary to Chapter IV both conceptually and practically. It is complementary in concept by addressing the speed-up of MBH by biasing the "hop from" location rather than the biasing the shape of the distribution from which incremental hop distances are drawn. Like the methods provided in Chapter IV, the methods provided here are novel.

The methods described in this chapter speed-up the MBH by accelerating the descent of f[$\boldsymbol{x}$[t]] into g[d], the bottom of which is f*, regardless of whether the search drives $\boldsymbol{x}$[t] further way from $\boldsymbol{x}$* in the short-term. This chapter is complementary in practice by providing MBH speed-up methods that are highly effective on poorly-behaved f for which the speed-up methods in Chapter IV were shown to be minimally effective or ineffective – the most dramatic example being the case of Gibsonian f.

V.1 Primary theme of this chapter

The primary theme of this chapter is: Speeding-up the descent of f[$\boldsymbol{x}$[t]] = d into g[d], regardless of whether the descent is sped-up into a basin containing a local minimum or the basin containing the global minimum, speeds-up the convergence to the global minimum. This is accomplished by a two-layer search process in which the inner layer seeks to descend more deeply into g[d] into some basin within every MBH time-step.

The reason this is effective is that the MBH algorithm prevents f[$\boldsymbol{x}$[t]] from increasing, therefore the descent of f[$\boldsymbol{x}$[t]] = d into g[d] is a non-increasing process. Therefore, the faster the MBH descends even into a "wrong" basin, i.e., a basin that contains a local minimum but not the global minimum, and which is necessarily less deep than the basin containing the global minimum,

the lower the probability that the MBH will remain in the wrong basin. Appendix C provides analytical as well as geometrically constructed support for this assertion.

The methods described in this chapter can be used in combination with the methods provided in Chapter IV. One such combination is demonstrated in the use-case described in Chapter VI. Further, evidence in this chapter, Chapter IV and Chapter V shows that MBHs operating on different f and $\boldsymbol{X}^{\mathbb{F}}$ respond differently to different speed-up methods. Further, that suggests that response to speed-up methods may be a way for an on-going MBH to autonomously characterize and classify the f and $\boldsymbol{X}^{\mathbb{F}}$ upon which it is operating. That open question is taken up in Chapter VII after the supporting evidence has been presented.

This chapter provides two methods for speeding-up MBH by biasing the "hop from" location: Stochastic Constrained Local Search (SCLS) and Multiple Communicating Hoppers (MCH). Both provide an inner layer to the basic MBH algorithm (per Chapter II) during each MBH time-step. SCLS provides opportunities for improvement to $\mathcal{G}[t]$ and thereby $f[\mathcal{G}[t]]]$, and MCH provides opportunities improvement to $\boldsymbol{x}[t\text{-}1]$ and thereby $f[\boldsymbol{x}[t\text{-}1]]]$, before the test of whether $f[\mathcal{G}[t]]] < f[\boldsymbol{x}[t\text{-}1]]]$ is applied in order to determine whether $\boldsymbol{x}[t]$ be will set equal to $\mathcal{G}[t]$ or remain equal to $\boldsymbol{x}[t\text{-}1]$, and $f[\boldsymbol{x}[t]]$ will be set equal to $f[\mathcal{G}[t]]$ or remain equal to $f[\boldsymbol{x}[t\text{-}1]]$.

Because Chapter IV ended with clear evidence that the methods of Chapter IV, biasing the shape of the distribution from which incremental hop distances are drawn, are ineffective on Gibsonian f, whereas SCLS is highly effective on Gibsonian f, this chapter begins with SCLS.

V.2 Stochastic Constrained Local Search (SCLS)

Before presenting the SCLS algorithm incorporated into the basic MBH algorithm, and then showing its effectives on Gibsonian f , and on textured prototypical f having a disconnected, sparse feasible domain, some historical context is helpful. Recall from Chapter I that early MBH

researchers and users believed that some form of local search needed to be incorporated into MBH when working on problems for which f is globally non-convex but convex in small local neighborhoods. The idea was that the combination of some form of local search and global random search could produce a set of "putative" global minima that could be later compared to determine which is actually the global minimum. There was no understanding that incorporating some form of local search into global random search could speed-up an MBH directly to the global minimum. To the present author's knowledge, the first MBH incorporation of deterministic local search into global random search specifically for the purpose of accelerating MBH directly to the global minimum was Jacob Englander, with improvements later made by Ellison and Ozemik [8, 9]. Their applications involved high-dimensional spacecraft trajectory optimization problems, and their local search method involved a deterministic gradient search method using a Non-linear Programming NLP solver named Sparse Nonlinear OPTimizer (SNOPT) [56]. That method has many issues and challenges that are summarized in the table at the end of this chapter, in which it is referred by the present author as Deterministic Constrained Local Search (DCLS). DCLS frequently generates singularities or otherwise fails numerically when it is operating in a neighborhood of $\boldsymbol{X}$ in which f is too flat or too rough, and it provides little benefit when $\boldsymbol{X}^{\mathbb{F}}$ is severely disconnected and sparse. As a result, DCLS fails to provide any benefit during many MBH time-steps and wastes a great deal of CPU execution time depending upon the failure rates and modes that are dependent upon f and $\boldsymbol{X}^{\mathbb{F}}$.

Despite the drawbacks of DCLS, Englander and Ellison found by experimentation that they could not solve their MBH application problems in allowable time without using it [57]. They did not investigate why it made the difference between application "success" or failure" within allowable time, why it worked despite failing during a majority of MBH time-steps, or whether a

stochastic version might be an improvement. Those topics are addressed in this chapter. They did make important contributions to their use of SNOPT, and through their close relationship with the SNOPT development, to SNOPT itself.

SCLS was developed by the present author as a way of achieving some of the benefits of DCLS without its liabilities.

In order to benefit from a close proximity to evidence in Chapter IV that biasing the shape of the distribution from which incremental hop distances are drawn is ineffective on Gibsonian *f*, the application of SCLS to Gibsonian *f* is presented here, before the algorithm is defined and then applied to textured prototypical *f* over a disconnected, sparse domain. Figure V.1 depicts Gibsonian *f* upon which MBH operated using fixed non-Gaussian p and SCLS for which a maximum of 32 local steps per MBH time-step was allowed. Note that the speed-up of convergence provided by SCLS enables $f[\boldsymbol{x}[t,n]]$ for Gibsonian *f* to be plotted on a horizontal axis in the range of [0, 2000] as was with prototypical *f*, such as in Figures II.5, II.6 and II.7, rather than [0,650000] used in Figure III.3. Also note in Figure V.1 that in all of the 48 trials, the MBH quickly found the local minimum in the wide basin that is not the global minimum in the narrow basin and found the global minimum in many of the 48 trials but not all of them. Then note that in Figure V.2 that, not only plotted on a horizontal axis [0, 2000], but also, in all of the 48 trials, the MBH quickly found the local minimum in the wide basin that is not the global minimum in the narrow basin – and found the global minimum in all of them before MBH time-step 2,000. Compare this to Figure II.3 in which, without SCLS, convergence to the lobal minimum required hundreds of thousands of MBH times-steps, and for one trial, 500,000. Finally, note that Figure V.3 is not only plotted on a horizontal axis [0, 2000], but also, in all of the 48 trials, the MBH

quickly found the local minimum in the wide basin that is not the global minimum in the narrow basin – and found the global minimum in all of them before MBH time-step 1,000.

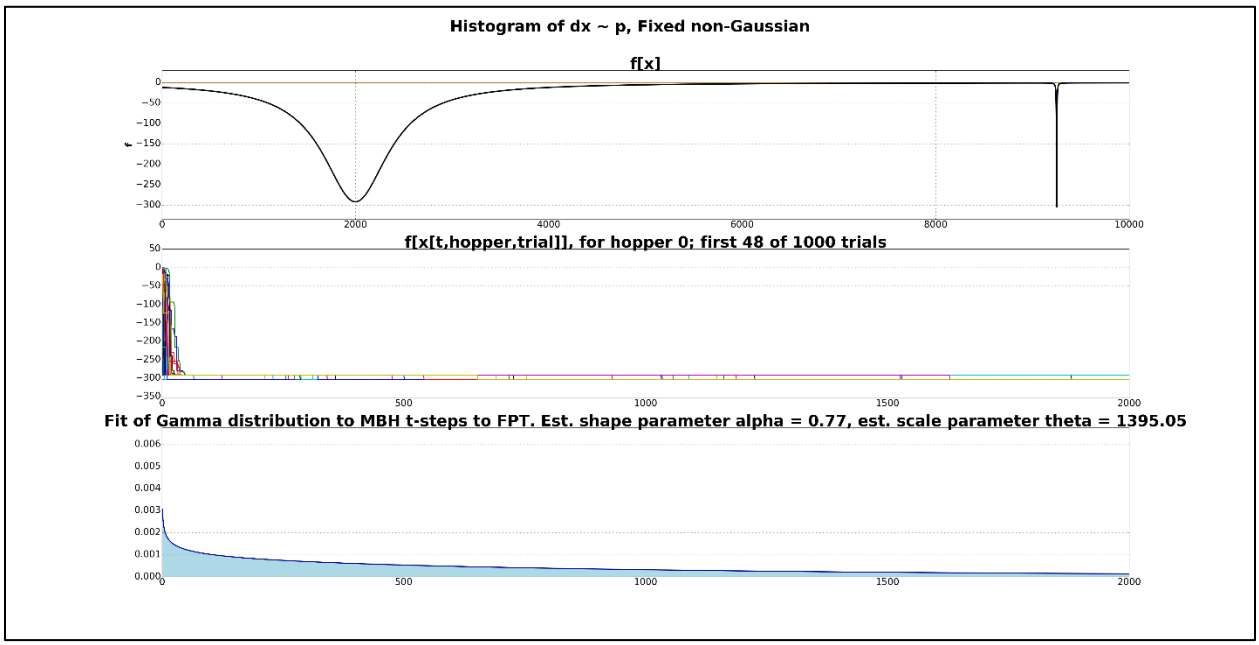


Figure V.1:

Upper panel: Gibsonian *f* upon which MBH operated using fixed non-Gaussian p and SCLS for which a maximum of 32 local steps per MBH time-step was allowed

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 1000 trials.

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 1,000 FPTs.

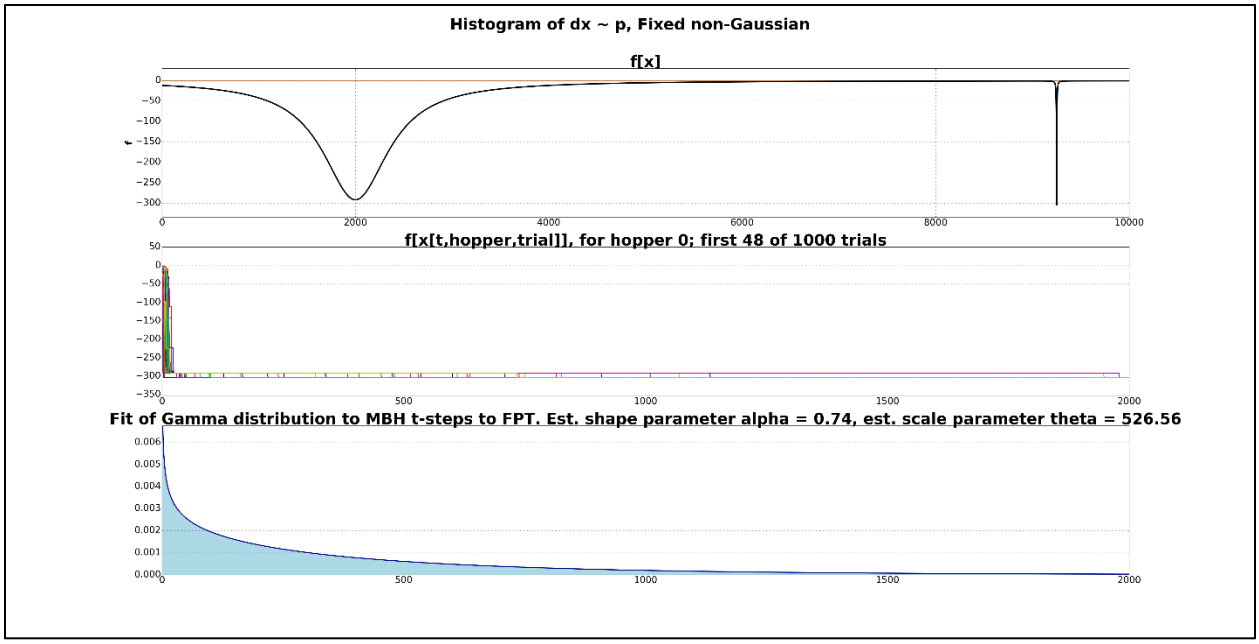


Figure V.2:

Upper panel: Gibsonian *f* upon which MBH operated using fixed non-Gaussian p and SCLS for which a maximum of 64 local steps per MBH time-step was allowed

Middle panel: $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 1,000 trials.

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 1,000 FPTs.

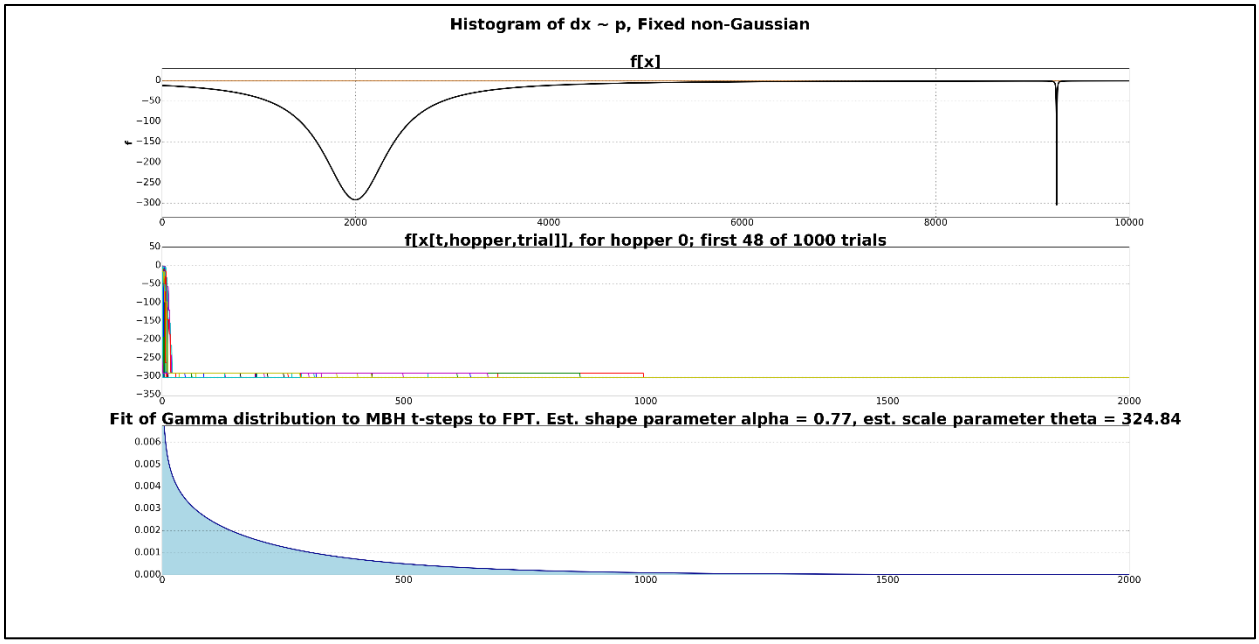


Figure V.3:
Upper panel: Gibsonian *f* upon which MBH operated using fixed non-Gaussian *p* and SCLS for which a maximum of 96 local steps per MBH time-step was allowed
Middle panel: $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 1,000 trials
Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 1,000 FPTs

SCLS is a stochastic version of DCLS that is related to the work of Robbins and Munro in that it retains the notion of the gradient descent (in this case stochastic gradient descent) except that the number of allowable local search steps is limited, forcing the stochastic gradient descent to be on a tether of a prescribed length [58, 59, 60, 61]. SCLS is related to the work of Robbins and Monro, as well as Kiefer and Wolfowitz, in that SCLS entails (in this case local) convex optimization. However, in the present work, stochastic gradient descent is used only as an inner-layer of constrained local search within an otherwise random search for a global minimum of a non-convex objective function.

The MBH algorithm incorporating SCLS is specified as follows:

1. At every MBH time-step t, $\boldsymbol{x}[t\text{-}1]$ was defined by the previous time-step t-1 and was necessarily in $\boldsymbol{X}^{\mathbb{F}}$. Likewise, $f[\boldsymbol{x}[t\text{-}1]]$ was evaluated in the previous time-step t-1.
2. Then, within the current MBH time-step t, draw $\Delta\boldsymbol{x} \sim \boldsymbol{p}$ and generate $\xi[t] = (\boldsymbol{x}[t] + \Delta\boldsymbol{x})$.
3. For j = 1, 2, 3, …, J where J is the number of potential sub-time-step allowed:
    a. $\xi[t] = (\xi[t] + \delta\mathbf{x})$ if $f[(\xi[t] + \delta\mathbf{x})] < f[(\xi[t])]$, otherwise $\xi[t] = \xi[t]$.
4. Determine whether $\xi[t]$ is the feasible subspace $\boldsymbol{X}^{\mathbb{F}}$ of $\boldsymbol{X}$. If it is not, draw another $\Delta\boldsymbol{x}$ and, thereby, form another $\xi[t]$.
5. Evaluate $f[\xi[t]]$ and compare $f[\xi[t]]$ to $f[\boldsymbol{x}[t]]$.
6. If $f[\xi[t]] < f[\boldsymbol{x}[t]]$, then replace $f[\boldsymbol{x}[t]]$ with $f[\xi[t]]$ and $\boldsymbol{x}[t]$ with $\xi[t]$. If $f[\xi[t]] \geq f[\boldsymbol{x}[t]]$, then $f[\boldsymbol{x}[t]]$ and $\boldsymbol{x}[t]$ remain unchanged.
7. Advance the iteration counter t and return to Step 1.

Regarding step 3.a: $\xi[t]$ is a constrained conditional random walk that stochastically attempts to descend the downward local gradient around $(\boldsymbol{x}[t] + \Delta\boldsymbol{x})$. The range in $\boldsymbol{X}$ of that constrained conditional random walk is dependent on the scale factor of the distribution from which the $\delta\boldsymbol{x}$ are drawn, namely $p_{\delta\boldsymbol{x}}$, and on J. The scale parameter of $p_{\delta\boldsymbol{x}}$ is chosen to be small and J is chosen to be a large as possible without adversely impacting the CPU time that is added by the use of SCLS. Distribution $p_{\delta\boldsymbol{x}}$ is not the same as distribution $p_{\Delta\boldsymbol{x}}$. Distribution $p_{\delta\boldsymbol{x}}$ has a small scale parameter and does not need to have a narrow mid-section or long-tails because the search (conditional random walk) that it drives is highly local. Simulation evidence indicates that the speed-up of MBH due to the use of SCLS is not sensitive to the shape of $p_{\delta\boldsymbol{x}}$. However, that warrants further investigation. In contrast to DCLS, if f is severely non-smooth or flat in the local neighborhood around $(\boldsymbol{x}[t] + \Delta\boldsymbol{x})$, SCLS may not provide any benefit, however the amount of time

it will waste is bounded by J. Further, the amount of time wasted is consistent and not aggravated by numerical complications such as singularities of matrices that cannot be inverted and the error messages and recovery mechanisms that such pathologies involve.

By comparing the specification above to the specification of MBH in Chapter II, one can see that SCLS is not difficult to implement. The effectiveness of SCLS on Gibsonian *f* was illustrated in Figures V.1, V.2 and V.3 above. Its effectiveness on textured *f* having a disconnected, sparse feasible domain is illustrated in Figures V.4, V.5, V6 and V.7 below. The texture used often poses challenges to DCLS.

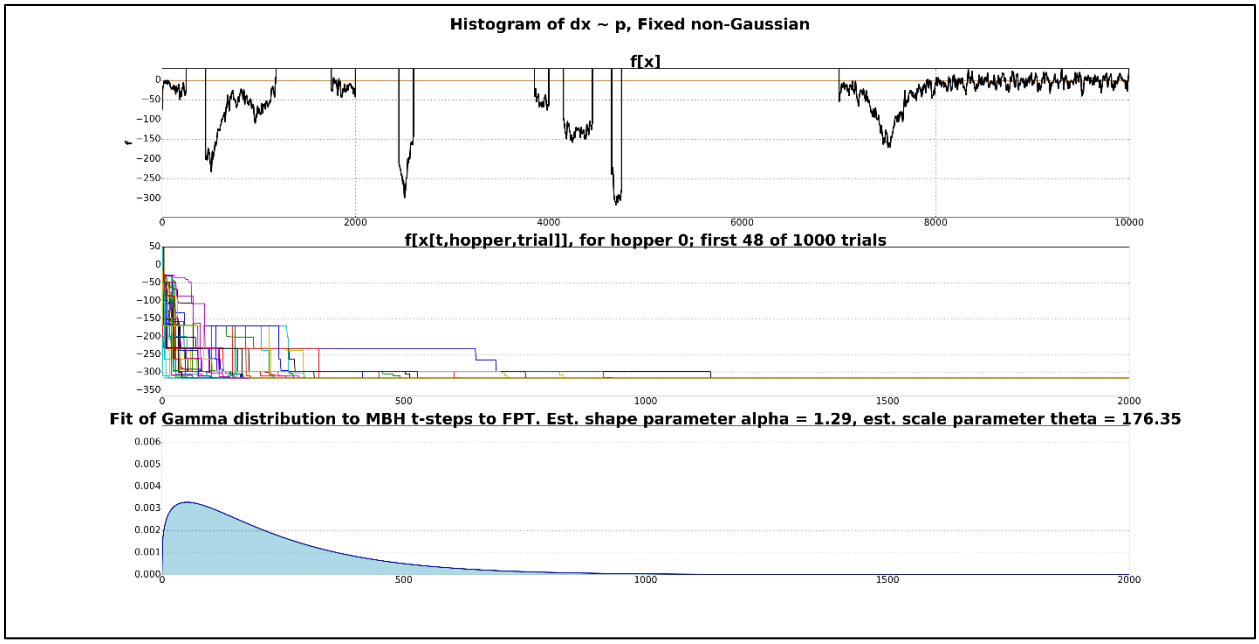


Figure V.4:

Upper panel: Textured prototypical *f* having a disconnected and sparse domain, upon which MBH operated using fixed non-Gaussian *p* and SCLS in which a maximum of 16 local steps per MBH time-step was allowed

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 1,000 trials

Bottom panel: $\Gamma(\alpha,\theta)$ fit as the FPTD of the 1,000 FPTs.

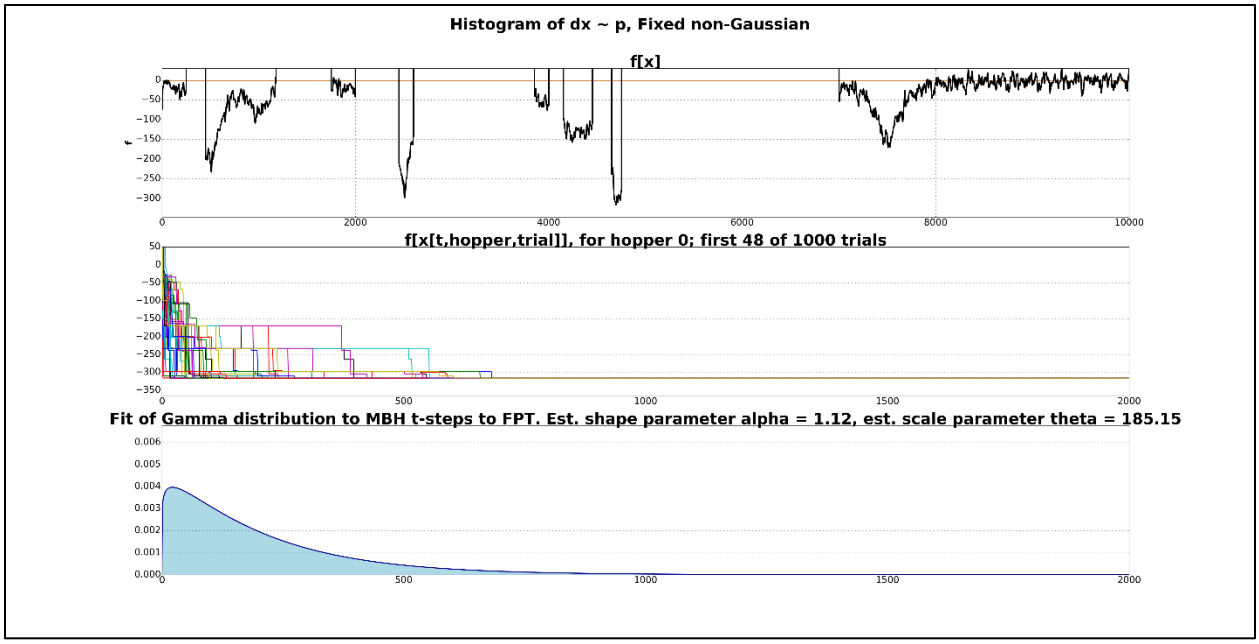


Figure V.5:

Upper panel: Textured prototypical *f* having a disconnected and sparse domain upon which MBH operated using fixed non-Gaussian *p* and SCLS for which a maximum of 32 local steps per MBH time-step was allowed

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 1,000 trials

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 1,000 FPTs.

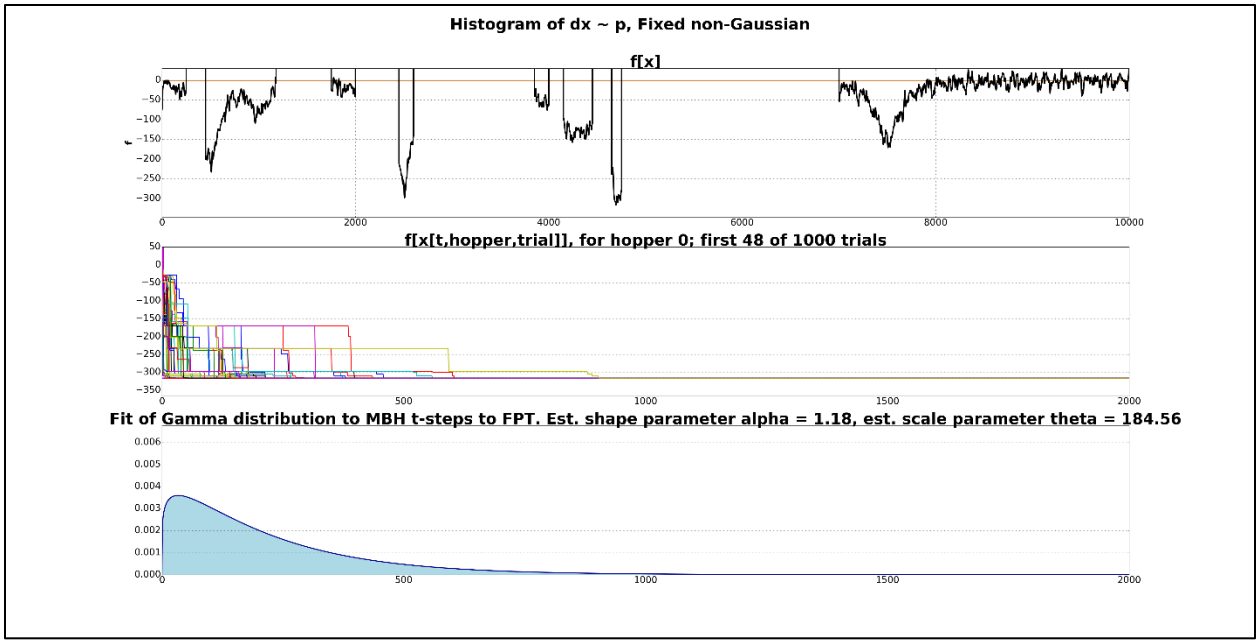


Figure V.6:

Upper panel: Textured prototypical f having a disconnected, sparse domain upon which MBH operated using fixed non-Gaussian p and SCLS for which a maximum of 64 local steps per MBH time-step was allowed

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being the first 2,000 of 10,000 MBH time-steps, n being the first 48 of 1,000 trials

Bottom panel: $\Gamma(\alpha,\theta)$ fit as the FPTD of the 1,000 FPTs.

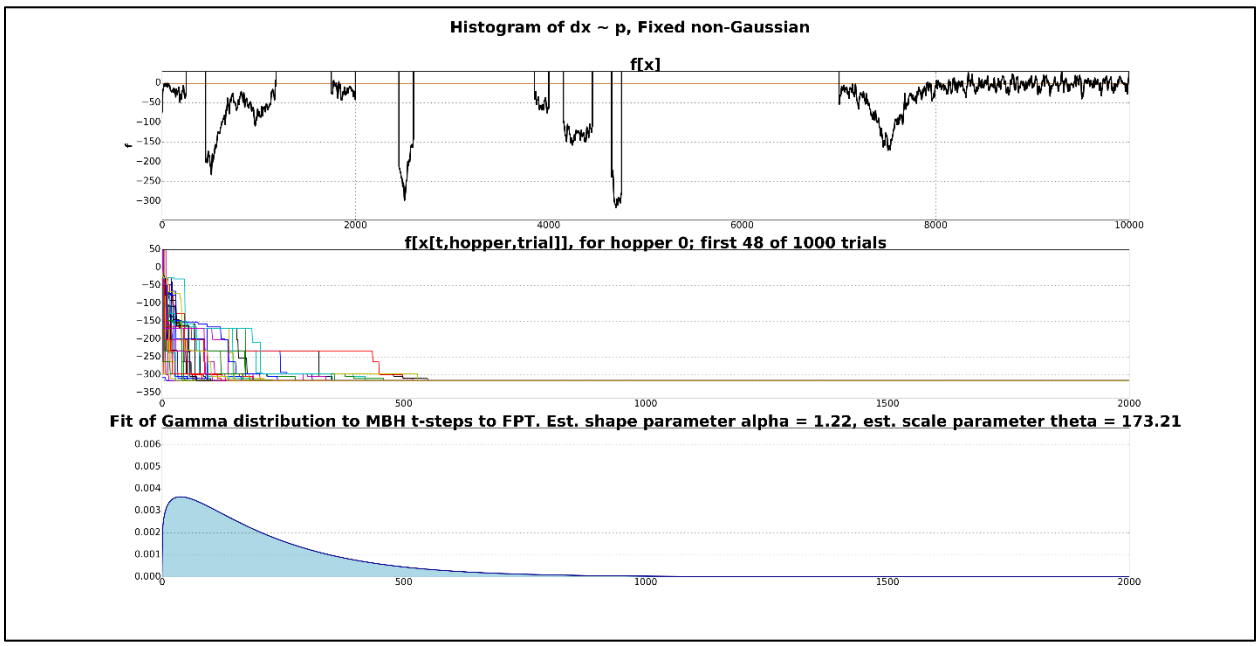


Figure V.7:
Upper panel: Textured prototypical *f* having a disconnected and sparse domain upon which MBH operated using fixed non-Gaussian *p* and SCLS for which a maximum of 96 local steps per MBH time-step was allowed
Middle panel: $f[\boldsymbol{x}[t,n]]$ for *t* being the first 2,000 of 10,000 MBH time-steps, *n* being the first 48 of 1,000 trials
Bottom panel: $\Gamma(\alpha,\theta)$ fit as the FPTD of the 1,000 FPTs.

By comparing the sequence of Figures V.4, V.5, V.6 and V.7, it is apparent that increasing the number of allowable local steps per MBH time-step does not make a significant difference for an MBH operating on textured *f* having a disconnected, sparse domain when fixed non-Gaussian *p* is used. Further, by comparing the sequence of Figures V.1, V.2 and V.3, to the sequence of Figures V.4, V.5, V.6 and V.7, it is apparent that MBH operating on different *f* respond differently to the same speed-up method. Here, the MBH operating on Gibsonian *f* continues to benefit significantly by increasing the number of allowable local SCLS hops per MBH time-step when using fixed non-Gaussian *p*, whereas the MBH operating on textured *f* having a disconnected, sparse domain does not.

V.3 Multiple Communicating Hoppers (MCH)

MCH is used in the Pioneer 11 trajectory optimization use-case described in Chapter VI. MCH involves M = {1, 2, 3, …} simultaneous hoppers. The notation for specifying it requires that $\boldsymbol{x}[t\text{-}1]$ be rewritten as $\boldsymbol{x}[t\text{-}1,m] = \boldsymbol{x}[t\text{-}1,\aleph]\ \forall\ m,$ where $\aleph$ is the "best" $\boldsymbol{x}$ found by the collection of M hoppers at time-step *t-1* and "best" is defined as the most *f*-minimizing $\boldsymbol{x}[t\text{-}1,m]$. Then, $\boldsymbol{\xi}[t,m] = (\boldsymbol{x}[t\text{-}1,\aleph] + \Delta\boldsymbol{x})$ and $\mathbf{x}[t,m]$ is set equal to $\boldsymbol{\xi}[t,m]\ \forall\ m$ if and only if $f[\boldsymbol{\xi}[t,m]] < f[\boldsymbol{x}[t\text{-}1,m]]$ meaning $f[(\boldsymbol{x}[t\text{-}1,\aleph] + \Delta\boldsymbol{x})] < f[\boldsymbol{x}[t\text{-}1,\aleph]]$. For a single-CPU machine, increasing M increases the speed-up of MBH convergence measured in MBH time-steps. Empirical results suggest that 8 < M < 16 and consumes little additional CPU time compared to smaller choices of M, choosing M larger than 32 increases the CPU time required per MBH time-step to an extent that is undesirable.

Just as SCLS is related to the work of Robbins and Monro, as well as Kiefer and Wolfowitz, which is a valid comparison because SCLS entails (in this case local) convex optimization, MCH is related to the work of Spall [62] as a form of multi-agent (in this case local) convex optimization . However, as in the case of the relationship between SCLS and the work of Robbins and Monro, as well as Kiefer and Wolfowitz, in the present work, MCH is different from the work of Spall in that MCH is only as an inner-layer within an otherwise random search for a global minimum of a non-convex objective function.

The MBH algorithm incorporating MCH is specified below:

1. At every MBH time-step t, $\boldsymbol{x}[t\text{-}1,m] = \boldsymbol{x}[t\text{-}1,\aleph]\ \forall\ m$ was defined in the previous time-step, where $\aleph$ is the "best" $\boldsymbol{x}$ found by the collection of M hoppers at time-step *t-1* and "best" is defined as the most f-minimizing $\boldsymbol{x}[t\text{-}1,m]$. $\boldsymbol{x}[t\text{-}1,\aleph]$ was necessarily in $\boldsymbol{X}^{\mathbb{F}}$. Likewise, $f[\boldsymbol{x}[t\text{-}1,\aleph]]$ was evaluated at the previous time-step.
2. Then, within the current time-step t, for multiple M simultaneous hoppers $\{1, 2, 3, \ldots, m\}$:
    a. Draw $\Delta\boldsymbol{x}[m] \sim \boldsymbol{p}$ and form $\xi[t,m] = (\boldsymbol{x}[t\text{-}1,\aleph] + \Delta\boldsymbol{x}[m])$
    b. Test whether $\xi[t,m] \subset \boldsymbol{X}^{\mathbb{F}}$. If $\xi[t,m] \not\subset \boldsymbol{X}^{\mathbb{F}}$, draw a different $\Delta\boldsymbol{x}[m] \sim \boldsymbol{p}$ and form $\xi[t,m]$ as many times as is necessary to form $\xi[t,m] \in \boldsymbol{X}^{\mathbb{F}}$
    c. Evaluate $f[\xi[t,m]]$
    d. Test whether $f[\xi[t,m]] < f[\boldsymbol{x}[t\text{-}1,\aleph]]$
    e. If $f[\xi[t,m]] < f[\boldsymbol{x}[t\text{-}1,\aleph]]$ then set $\boldsymbol{x}[t,m] = \xi[t,m]$. Otherwise set $\boldsymbol{x}[t,m] = \boldsymbol{x}[t\text{-}1,\aleph]$
3. At the end of time-step t, compare $f[\xi[t,:]]$ where : is across all m ∈ M. Label $\text{argmin}_m(f[\boldsymbol{x}[t,m]])$ as m_{best}
4. For multiple M simultaneous hoppers $\{1, 2, 3, \ldots, m\}$:
    a. If m is not m_{best}, set $\boldsymbol{x}[t,m] = \boldsymbol{x}[t, m_{best}]$ and $f[\boldsymbol{x}[t,m]] = f[\boldsymbol{x}[t, m_{best}]]$
5. Advance the iteration counter t and return to Step 1.

The speed-up of MBH by MCH is illustrated in Figures V.8 through V.16 below and in Chapter VI where MCH was applied to the Pioneer 11 trajectory optimization use-case. In Figures V.8 through V.11, the effectiveness of MCH on a 2-dimensional f that is Gibsonian when projected onto an axis that is diagonal to the x and y axes in the sense of being the linear combination of equal x and y components. For brevity, this f is referred to loosely as 2-dimensional Gibsonian f.

Figure V.8 shows the heatmap of f. Figure V.9 shows the convergence speed when no MCH is used. Figure V.10 shows the convergence speed when MCH is applied using 4 simultaneous hoppers. Figure V.11 shows the convergence speed when MCH is applied using 8 simultaneous hoppers. In Figure V.9, all trials converge to the global minimum, and the last trial to converge required approximately 2,250,000 MBH time-steps. In Figure V.10, all trials converge to the global minimum, and the last trial to converge required approximately 600,000 MBH time-steps. In Figure V.11, all trials converge to the global minimum, and the last trial to converge required approximately 300,000 rather than or 2,500,000 or even 600,000 MBH time-steps. It is noteworthy that, in this case, f is 2-dimensional and while the speed-up due to MCH is approximately linear with respect to the number M of MCH hoppers used, it is independent of dimensionality. The cost of using MCH on a low dimensional problem is roughly the same as the cost of using MCH on a high dimensional problem. This is very different from SCLS, where the computational cost increases as a power of dimensionality. Moreover, code being developed to implement MCH in real engineering settings queries the hardware as to the number of parallelizable CPUs available, and automatically parallelizes the MCH operations accordingly, thereby further increasing the speed-up achieved by MCH.

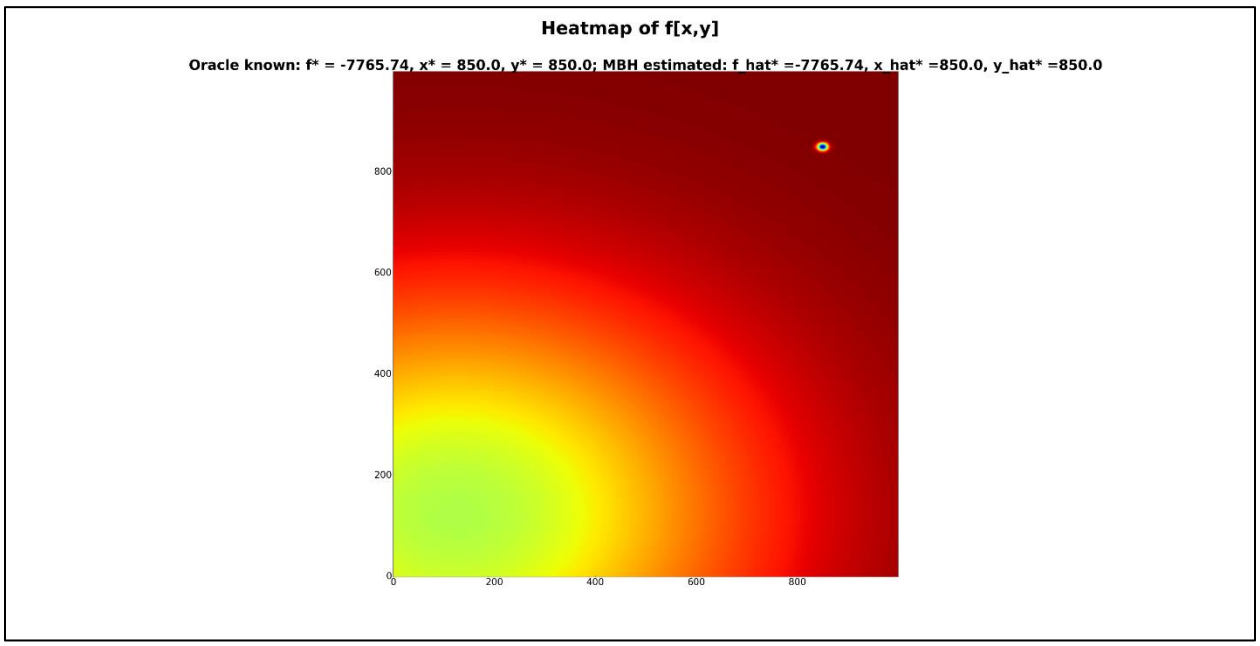


Figure V.8: Heatmap of a 2-dimensional Gibsonian *f*. Blue indicates the lowest values of *f*. Red indicates high values of *f*. Yellow indicates intermediate values of *f*.

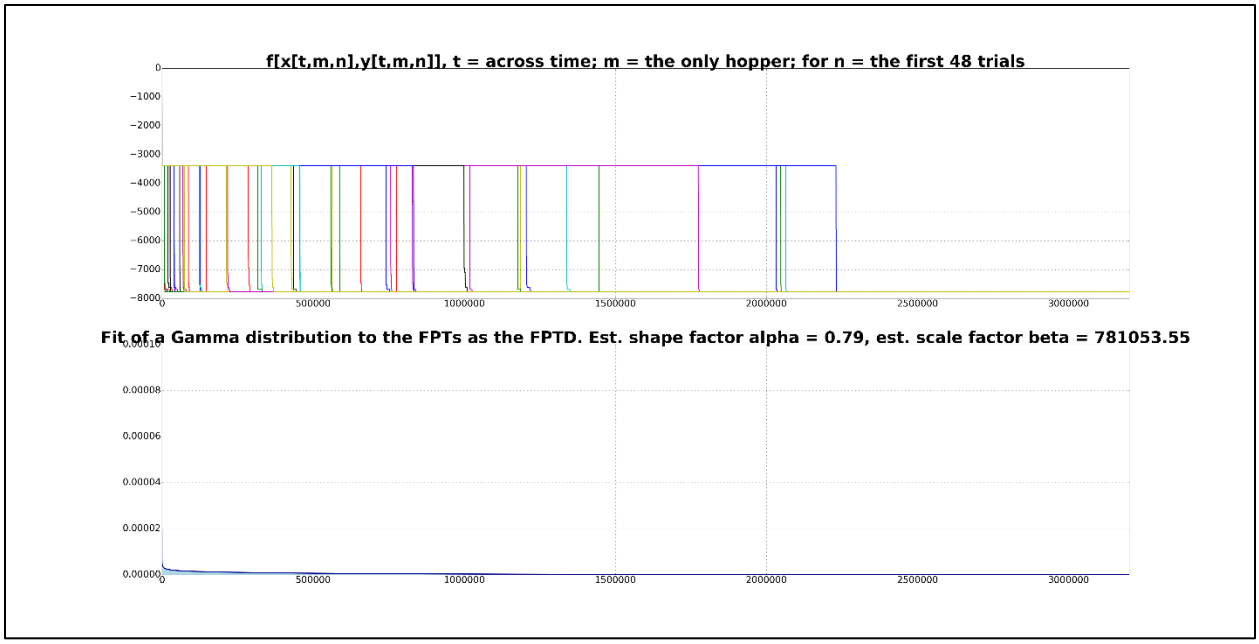


Figure V.9: Convergence performance of MBH operating on 2-dimensional Gibsonian f using 2-dimensional fixed non-Gaussian p that generated i.i.d incremental hop distances, no SCLS, and no MCH.

Upper panel: $f[\boldsymbol{x}[t,n],\boldsymbol{y}[t,n]]$ for t being each of 3,200,000 MBH time-steps, n being the first 48 of 100 trials

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 100 FPTs

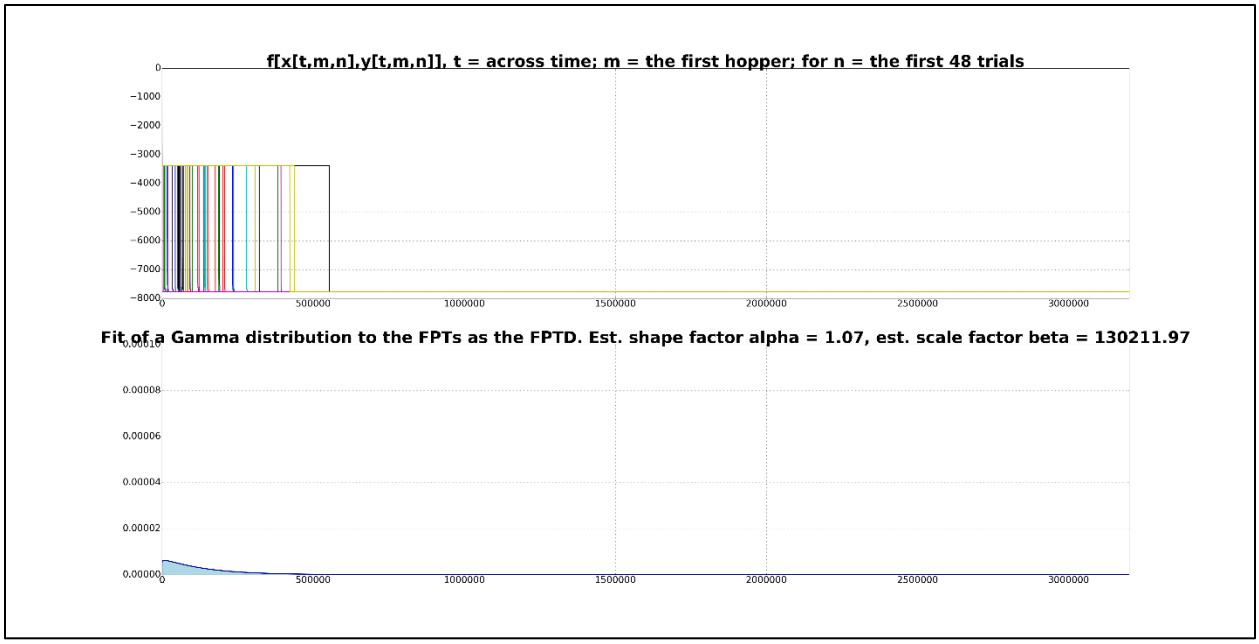


Figure V.10: Convergence performance of MBH operating on 2-dimensional Gibsonian *f* using 2-dimensional fixed non-Gaussian *p* that generated i.i.d incremental hop distances, no SCLS, but MCH with 4 simultaneous communicating hoppers.

Upper panel: $f[\boldsymbol{x}[t,n],\boldsymbol{y}[t,n]]$ for t being each of 3,200,000 MBH time-steps, n being the first 48 of 100 trials

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 100 FPTs.

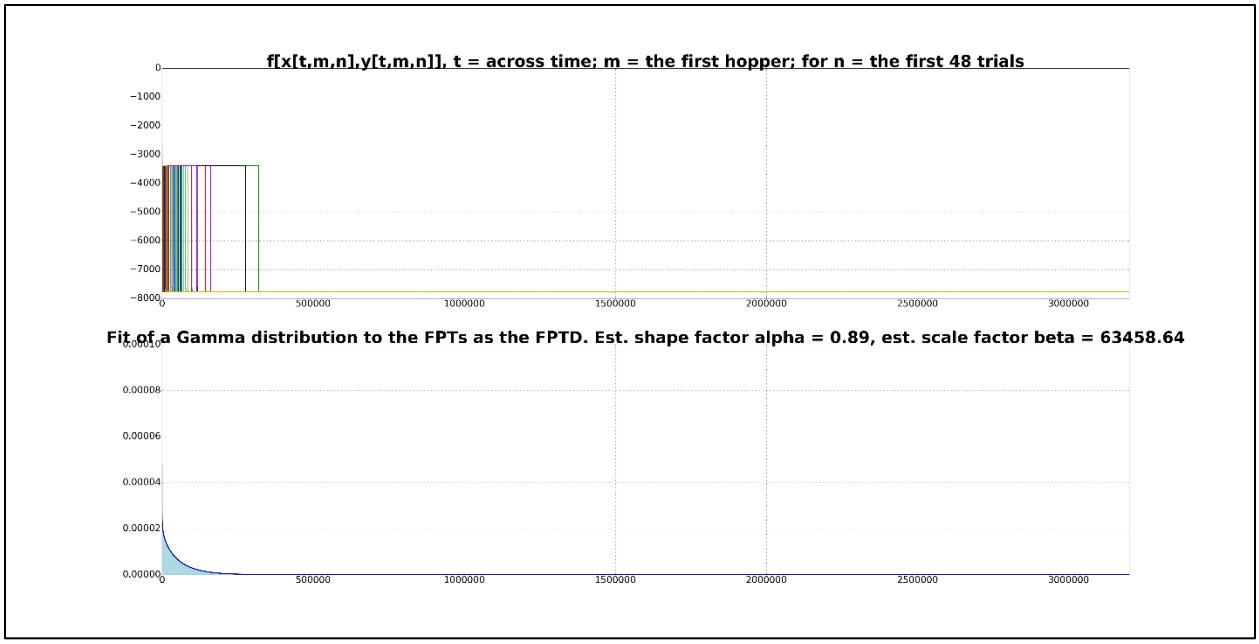


Figure V.11: Convergence performance of MBH operating on 2-dimensional Gibsonian *f* using 2-dimensional fixed non-Gaussian *p* that generated i.i.d incremental hop distances, no SCLS, but MCH with 8 simultaneous communicating hoppers.
Upper panel: $f[\boldsymbol{x}[t,n],\boldsymbol{y}[t,n]]$ for *t* being each of 3,200,000 MBH time-steps, *n* being the first 48 of 100 trials
Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 100 FPTs

In Figures V.12, V.13 and V.14, the effectiveness of MCH on 1-dimensional prototypical *f*, for which $\boldsymbol{X}^{\mathbb{F}}$ is the entirety of $\boldsymbol{X}$, is illustrated.

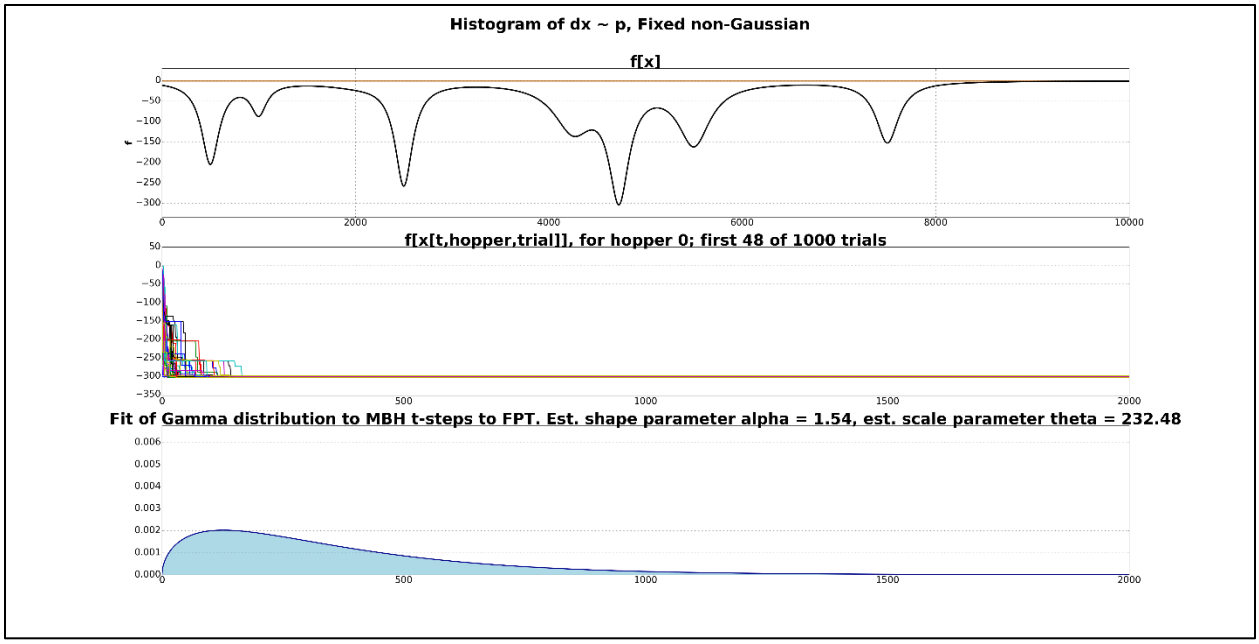


Figure V.12:

Upper panel: Prototypical f for which $\boldsymbol{X}^{\mathbb{F}}$ is the entirety of $\boldsymbol{X}$. The light brown horizontal line passes through maximum feasible f

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being each of 2,000 MBH time-steps, n being the first 48 of 100 trials , using 2 MCH and fixed non-Gaussian p having the histogram shown in Figure II.4

Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 1,000 FPTs.

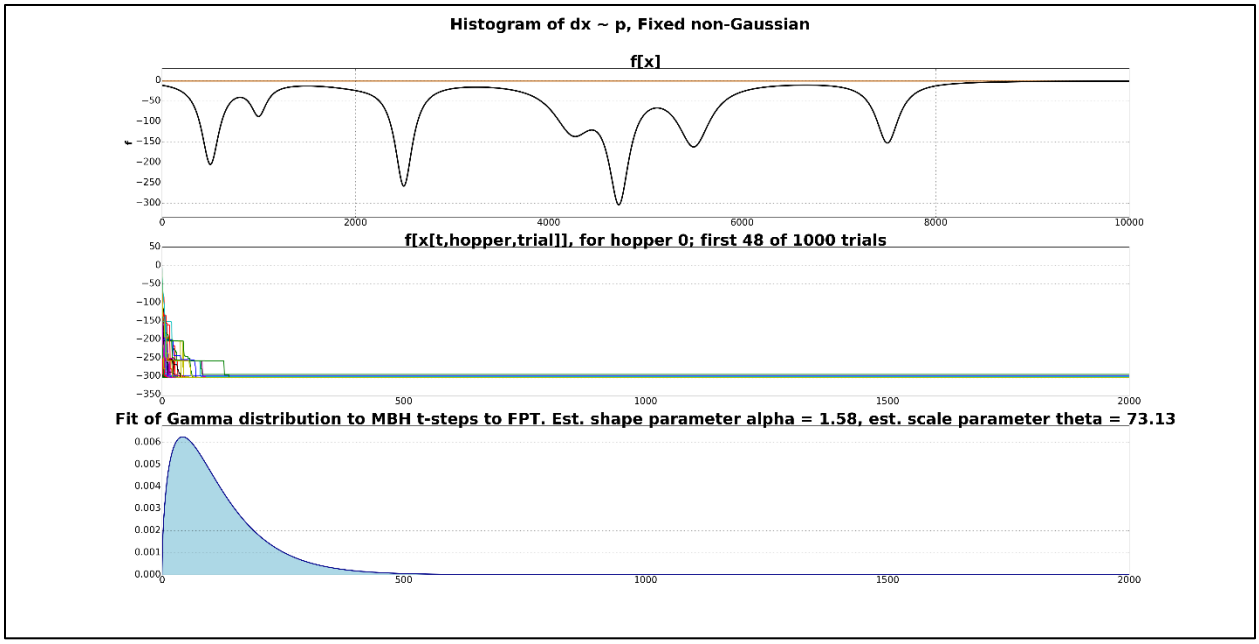


Figure V.13:
Upper panel: Prototypical f for which $\boldsymbol{X}^{\mathbb{F}}$ is the entirety of $\boldsymbol{X}$. The light brown horizontal line passes through maximum feasible f
Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being each of 2,000 MBH time-steps, n being the first 48 of 100 trials , using 6 MCH and fixed non-Gaussian p having the histogram shown in Figure II.4
Bottom panel: $\Gamma(\alpha, \theta)$ fit as the FPTD of the 1,000 FPTs.

The effectiveness of MCH on prototypical f with a disconnected sparse feasible domain, and on textured prototypical f with a disconnected sparse feasible domain, is similar but weaker than it is on prototypical f for which $\boldsymbol{X}^{\mathbb{F}}$ is the entirety of $\boldsymbol{X}$. That can be seen in Figures V.14 and through V.16. In Figure V.14, it is apparent that by using 12 MCH hoppers, in the middle panel all of 48 trials converged within the first 125 MBH time-steps, and in the lower panel the height of the Gamma-fit FPTD does not even fit on the standard y-axis used on other charts. In Figure V.15, it is apparent that using 12 MCH hoppers, in the middle panel a few of the 48 trials do not converge within the first 2,000 MBH time-steps, and in the lower panel the height of the Gamma-fit FPTD does not extend much higher than the upper limit of the y-axis. In Figure V.16, it is

apparent that by using 12 MCH hoppers, in the middle panel many of the 48 trials do not converge within the first 2,000 MBH time-steps, and in the lower panel the height of the Gamma-fit FPTD does not reach the upper limit of the y-axis.

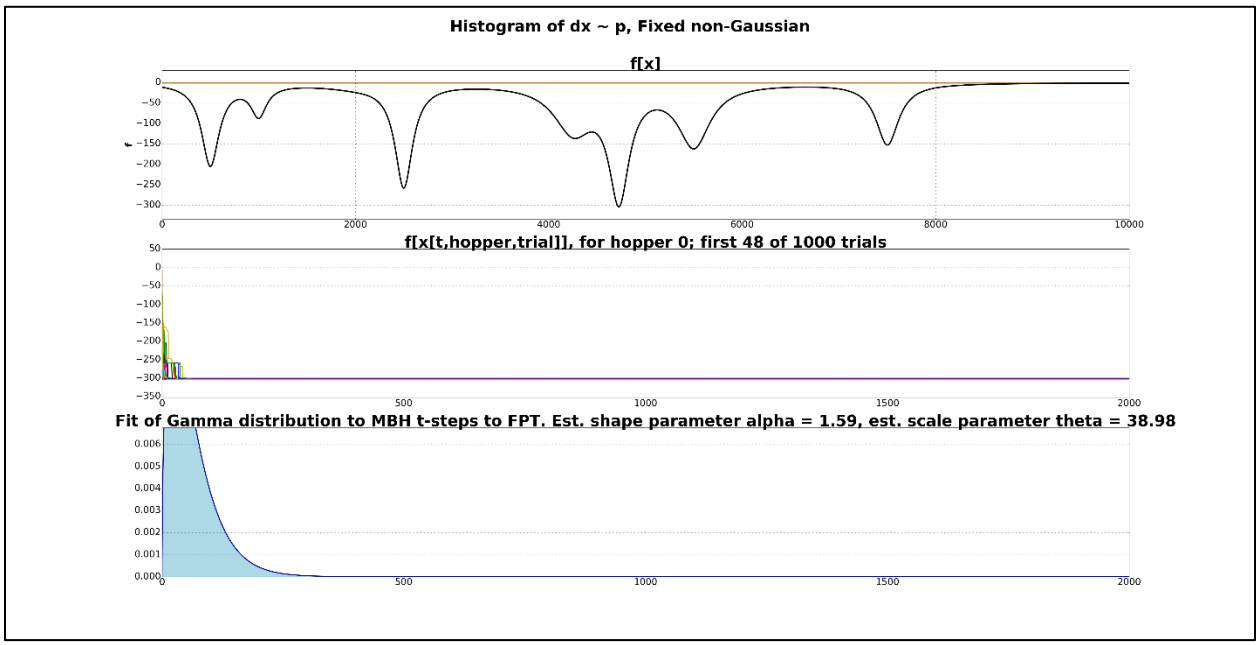


Figure V.14:

Upper panel: Prototypical f for which $\boldsymbol{X}^{\mathbb{F}}$ is the entirety of $\boldsymbol{X}$ upon which MBH operated using fixed non-Gaussian p and 12 MCH hoppers. The light brown horizontal line passes through maximum feasible f.

Middle panel: $f[\boldsymbol{x}[t,n]]$ for t being each of 2,000 MBH time-steps, n being the first 48 of 100 trials , using 12 MCH and fixed non-Gaussian p having the histogram shown in Figure II.4

Bottom panel: $\Gamma(\alpha,\theta)$ fit as the FPTD of the 1,000 FPTs.

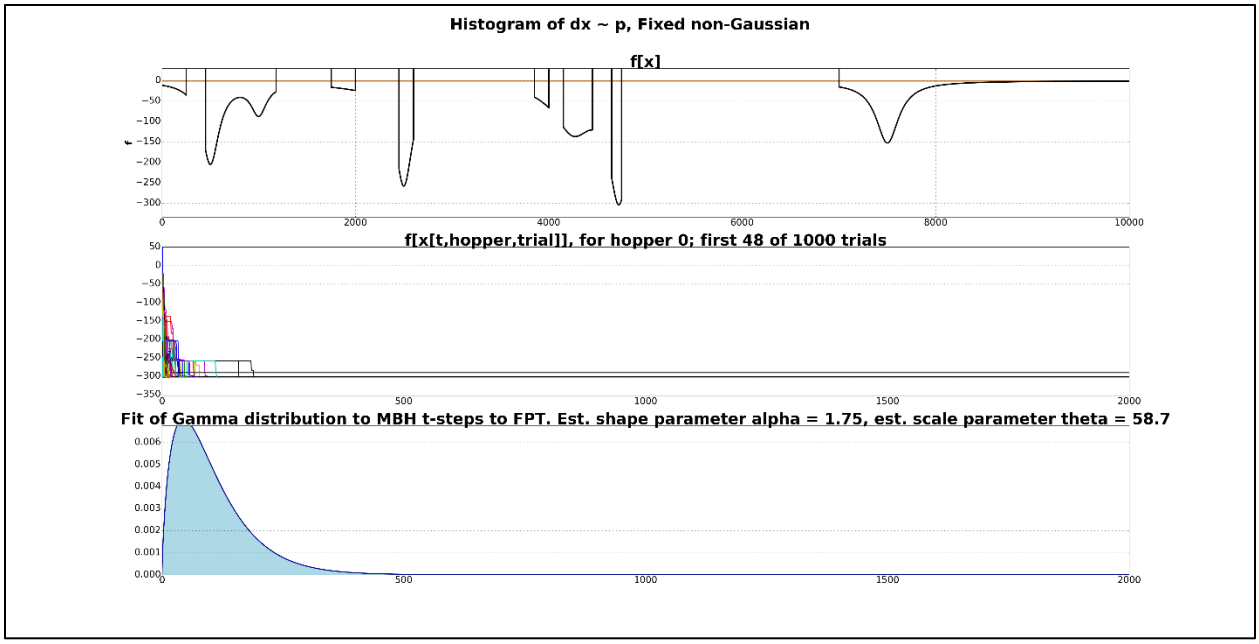


Figure V.15:

Upper panel: Prototypical *f* having a disconnected, sparse domain upon which MBH operated using fixed non-Gaussian *p* and 12 MCH hoppers. The light brown horizontal line passes through maximum feasible *f*

Middle panel: $f[\boldsymbol{x}[t,n]]$ for *t* being each of 2,000 MBH time-steps, *n* being the first 48 of 100 trials , using 12 MCH and fixed non-Gaussian *p* having the histogram shown in Figure II.4

Bottom panel: $\Gamma(\alpha,\theta)$ fit as the FPTD of the 1,000 FPTs

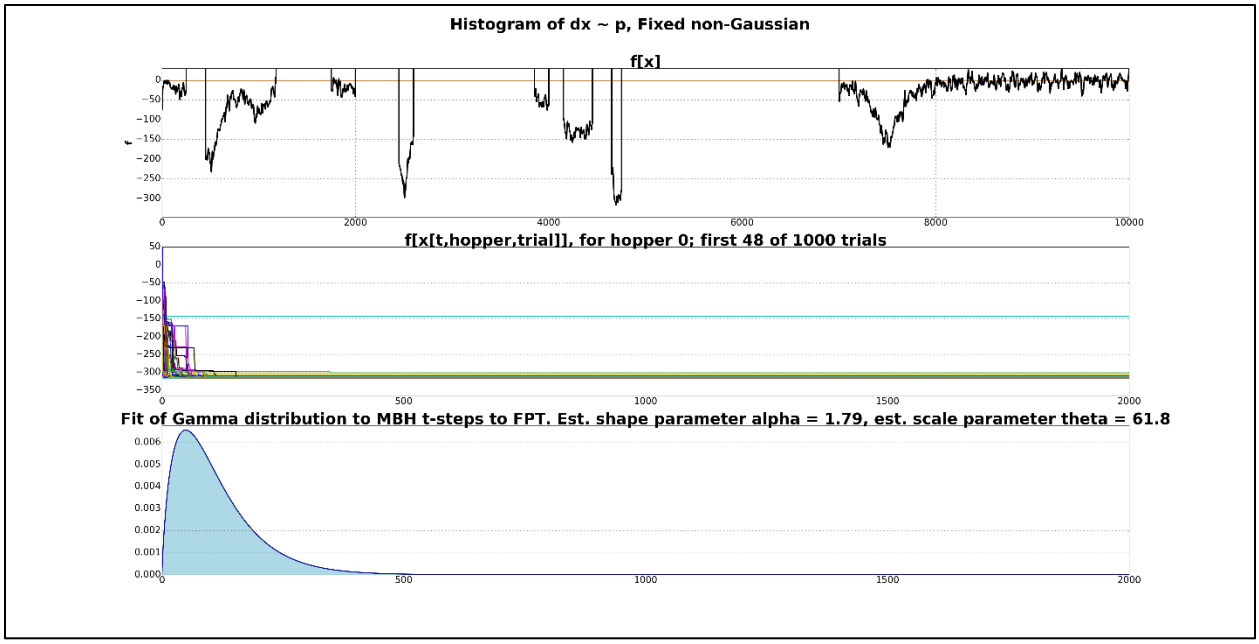


Figure V.16:
Upper panel: Textured f having a disconnected and sparse domain upon which MBH operated using fixed non-Gaussian p and MCH using 12 communicating hoppers. The light brown horizontal line passes through maximum feasible f
Middle panel: $f[\mathbf{x}[t,n]]$ for t being each of 2,000 MBH time-steps, n being the first 48 of 100 trials , using 12 MCH and fixed non-Gaussian p having the histogram shown in Figure II.4
Bottom panel: $\Gamma(\alpha,\theta)$ fit as the FPTD of the 1000 FPTs

The effectiveness of MCH will also be illustrated in Chapter VI where it is applied to the Pioneer 11 trajectory optimization use-case.

V.4 Combinations of methods

The methods provided in this chapter and Chapter IV can be combined. For example, adapting p to $\hat{q}$ combines naturally with MCH because the multiple hoppers used for MCH can also be used to reduce the variance in the estimate of the time-varying scale parameter of $\hat{q}[t]$, as is done in the use-case described in Chapter VI.

V.5 Computational costs of SCLS versus MCH

Depending upon how one defines computational costs, SCLS and MCH incur those costs differently. One natural way to measure computational costs is in units of queries to the "oracle" or physics model of the objective function *f*. In SCLS, where a maximum of *S* local constrained search steps are allowed per MBH time-step for the one hopper, that would be the number of queries to the "oracle" or physics model of the objective function $f[\xi[t,s]]$. In MCH, where a maximum of *M* simultaneous communicating hoppers are used, that would be the number of queries to the "oracle" or physics model of the value $f[\boldsymbol{x}[t,m]]$.

The number of queries of the objective function $f[\xi[t,s]]$ is dependent upon the maximum of *S* local constrained search steps that is allowed per MBH time-step for the one hopper, raised to the power of the dimensionality of $\boldsymbol{X}$. If $\boldsymbol{X}$ is one dimensional and $S = 50$, the number of queries required is 50. But if $S = 50$ and $\boldsymbol{X}$ is 2-dimensional, the number of queries required is $50^2 = 2{,}500$ If $S = 50$ and $\boldsymbol{X}$ is 3-dimensional, the number of queries required is $50^3 = 125{,}000$. If $S = 50$ and $\boldsymbol{X}$ is 100-dimensional, the number of queries is $50^{50} = 8.882$ E+84. Many real MBH applications entail 50 or even 100 dimensions. Fortunately, in real high-dimensional applications of MBH, the queries of the objective function are not, by themselves, time-consuming operations. However, this suggests that *S* should not be large as the dimensionality of $\boldsymbol{X}$ grows.

The number of queries of the value $f[\boldsymbol{x}[t,m]]$ is dependent upon the number *M* of simultaneous communicating hoppers that are used. Whereas the number of queries to $f[\xi[t,s]]$ is dependent upon the maximum of *S* local constrained search steps are allowed per MBH time-step for the one hopper, raised to the power of the dimensionality of $\boldsymbol{X}$, the number of queries to $f[\boldsymbol{x}[t,m]]$ is proportional to *M* times the dimensionality of $\boldsymbol{X}$. If *M* is increased from 8 to 16, and $\boldsymbol{X}$ is 1-dimensional, the number of queries to $f[\boldsymbol{x}[t,m]]$ doubles. If *M* is increased from 8 to 16, and

$\boldsymbol{X}$ is 2-dimensional, the number of queries to $f[\boldsymbol{x}[t,m]]$ quadruples. If M is increased from 8 to 16, and $\boldsymbol{X}$ is 3-dimensional, the number of queries to $f[\boldsymbol{x}[t,m]]$ increases by a multiplicative factor of 6. If M is increased from 8 to 16, and $\boldsymbol{X}$ is 100-dimensional, the number of queries to $f[\boldsymbol{x}[t,m]]$ increases by a multiplicative factor of 200 rather than 2^{100}.

This should not be construed to suggest that using MCH is always better than using SCLS because the effectiveness on the speed-up of an MBH, by MCH versus SCLS depends on f, $\boldsymbol{X}^{\mathbb{F}}$, and p, and tend to be non-linear in M or S respectively. Therefore, the speed-up of MBH on some combinations of f, $\boldsymbol{X}^{\mathbb{F}}$, and p, achieved by SCLS may be more significant than the speed-up provided by MCH and outweigh the cost of additional queries even when $\boldsymbol{X}$ is high-dimensional provided that S is kept sufficiently small. The use-case in Chapter VI shows that MCH provides the additional benefit of providing M concurrent samples of "accepted" $\boldsymbol{\Delta x}$ that can be used to reduce the variance in the estimation of $\hat{q}[t]$ as will be shown.

V.6 Both SCLS and MCH, in different ways, reduce the variance between FPTs

Figures V.17 and V.18 below show as simulation-based evidence that both SCLS and MCH reduce the variance between FPTs. In Figures V,17 and V.18, in each of 1,000 trials, the MBH is operating on the same f with the same p. The f is prototypical f illustrated in Figure 0.a, and distribution p is the fixed non-Gaussian p having the histogram in Figure II.4. Figure V.17 shows the relationship between the number of MCH communicating hoppers per MBH time-step and the standard deviation across 1,000 FPTs, using the same f with the same p. The implication of Figure V.17 is that SCLS reduces the variance across the FPTs. Figure V.18 shows the relationship between the number of MCH communicating hoppers per MBH time-step and the standard deviation across 1,000 FPTs, using the same f with the same p. The implication of Figure V.18 is that MCH reduces the variance across the FPTs.

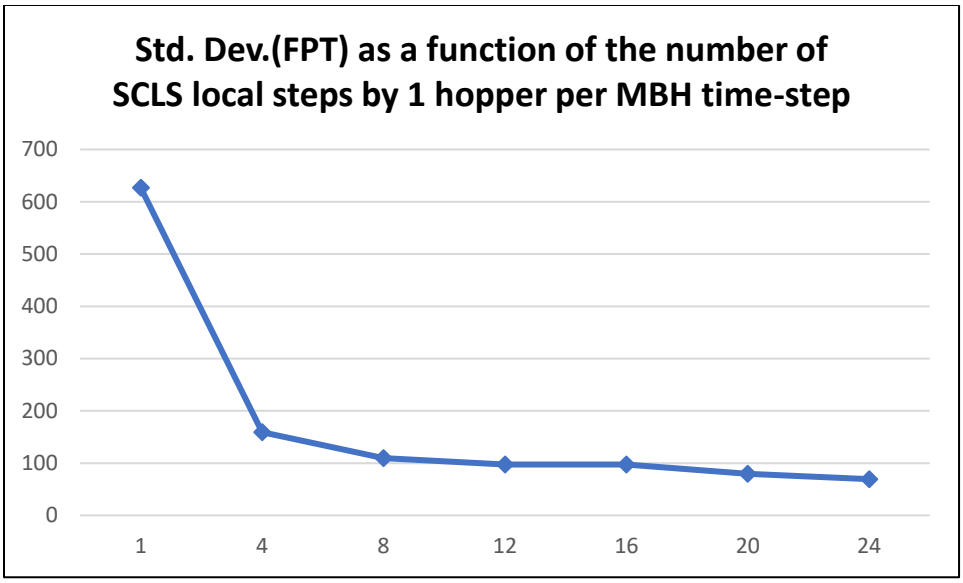


Figure V.17: The relationship between the number of SCLS local steps per MBH time-step and the standard deviation across 1,000 FPTs using the same *f* with the same *p*.

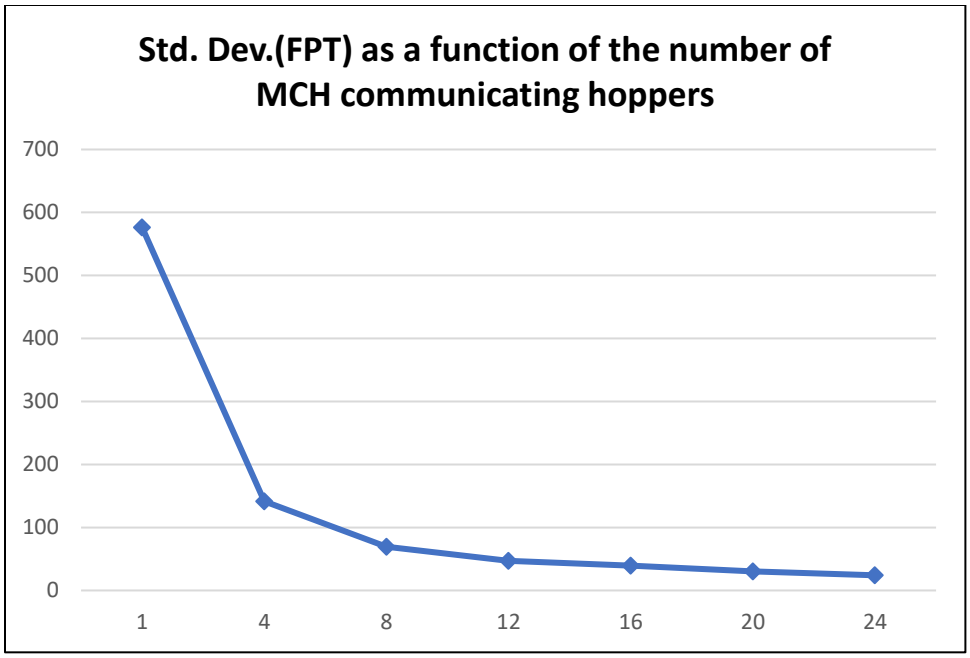


Figure V.18: The relationship between the number of MCH communicating hoppers per MBH time-step and the standard deviation across 1,000 FPTs using the same *f* with the same *p*.

V.7 Chapter summary

The central theme of this chapter is that speeding-up the descent of $f[\boldsymbol{x}[t]] = d$ into $g[d]$ speeds-up the convergence to f^* by an MBH operating on f and $\boldsymbol{X}^{\mathbb{F}}$, regardless of whether that speed-up of the descent of $f[\boldsymbol{x}[t]]$ further down into $g[d]$ causes, in the short-term, $\boldsymbol{x}$[t] to move farther from $\boldsymbol{x}^*$. That is because the descent of $f[\boldsymbol{x}[\mathrm{t}]]$ deeper into $g[d]$ is a non-increasing process. Speeding-up the descent of MBH into the "wrong" basin, i.e., the basin that contains a local minimum but the global minimum, speeds-up the decrease in the probability that the hopper will remain in the "wrong" basin. A formal explanation for this is given in Appendix C.

Two methods for accelerating the descent of $f[\boldsymbol{x}[t]] = d$ into $g[d]$, thereby the speed-up of the convergence of the MBH to f^*, were provided. SCLS was shown in to be effective on the 1-dimensional example of Gibsonian f in Figures V.1, V.2 and V.3, and MCH was shown to be effective on a 2-dimensional example of Gibsonian f as shown in Figures V.8 through V.11. These are methods complementary to, and can be used with, the speed-up methods provided and explained in Chapter IV. Further, for some f, the methods in Chapter V are more effective than the methods in Chapter IV, but when used to augment the methods in Chapter IV make the methods in Chapter IV more effective.

Finally, it was demonstrated by simulation-based evidence that the methods in Chapter V speed-up MBH differently depending upon f, $\boldsymbol{X}^{\mathbb{F}}$, and p.

The following table summarizes the methods described in this chapter:

TABLE II

Method	Benefit	Drawbacks
DCLS	• Long history of successful use in the optimization of difficult high-dimensional trajectory optimization problems • Incorporates widely-used and respected SNOPT Non-linear solver • Improvement over several years and versions by J. Englander, D. Ellison, and M. Ozimek	• Numerically brittle where f is flat or severely non-smooth, or $X^{\mathbb{F}}$ is disconnected and sparse • High rates of failure per MBH time-step • Many of the failure modes are very time-consuming • Large variance in CPU clock-time consumed per MBH time-step • Time-inefficient – except that it enables convergence times that would otherwise be non-feasible with respect to project schedules
SCLS	• Simpler than DCLS or MCH • Faster than DCLS or MCH for low-dimensional problems • Inherently robust	• Recent development • Execution time increases with the number of dimensions if a constant number of SCLS local search steps is maintained as the problem dimensionality increases • Not yet tested on difficult high-dimensional trajectory optimization problems • Not yet published • Not yet validated by other researchers • Comparison to DCLS in high-dimensional problems not yet investigated
MCH	• Developed by the present author in 2017 • Improved and extended by the trajectory optimization team at the NASA Goddard Space Flight Center • Faster than DCLS or SCLS for high-dimensional problems • Robust • Simple • Demonstrated in the Pioneer 11 trajectory optimization use-case described in Chapter VI and published in 2020	• Execution time increases with the number of communicating hoppers but not with the dimensionality of the problem • Comparison to DCLS in high-dimensional problems not yet investigated

VI. THE PIONEER 11 TRAJECTORY OPTIMIZATION USE-CASE

This chapter describes an application of speeding-up MBH by biasing the shape of the distribution from which incremental hop distances are drawn, namely *p*, per Chapter IV and by biasing the location from which each next hop is taken using MCH per Chapter V.

The use-case is the re-optimization of the historical Pioneer 11 mission, using the European Space Agency (ESA) Advanced Concept Team's (ACT's) PyKep 2-body model. In addition, colleagues at NASA Goddard Space Flight Center advised on the development of some of the code needed for interfacing between PyKep and the code developed by the present author for the experiments and graphics provided below.

$\boldsymbol{X}$, the space of decision variables, which comprise the domain of f, is 3-dimensional. The first dimension represents the epoch (day, hour, minute, second,… millisecond, …) of launch from the Earth. The second dimension represents the travel time (Time of Flight, a.k.a. TOF) to the targeted closest (nearest "flyby") point to Jupiter at which point the trajectory is adjusted to head towards Saturn, and the third dimension represents the travel time (Time of Flight, a.k.a. TOF) to the closest point to Saturn at which point the trajectory is adjusted to head toward a target outside the Solar System. Like the epoch of launch, TOFs are resolved to finer than milliseconds. The objective function to be minimized, namely $f[\boldsymbol{x}]$ where $\boldsymbol{x} = [x_1, x_2, x_3]$, is the integral over the time-span of the trajectory of the spacecraft – from its detachment from the Earth-launch vehicle to the targeted closest (nearest "flyby") point to Saturn – of $|\Delta v|$, the change in velocity achieved by the consumption of on-board propellant. Practitioners simply refer to f as "Δv" rather than as the time integral of $\boldsymbol{\Delta v}$ because the meaning of the expression $\boldsymbol{\Delta v}$ is widely understood. Further, for practitioners, f is a surrogate for the consumption of propellant. The consumption of propellant

needs to be minimized because the mass required for propellant competes with "dry mass" that includes scientific instruments. By design, only very small amounts of propellant are available on-board to maximize the mass dedicated to instruments that support the science mission. The value of $f[\boldsymbol{x}]$ is obtained by a query to PyKep after specifying $\boldsymbol{x}$. Pykep returns $f[\boldsymbol{x}]$ based on its physics model which includes a specification of the Pioneer 11 spacecraft, orbital models based on Lambert's problem, and an ephemeris that describes the position and velocity of all significant and relevant celestial bodies at every point in time. The value of f and all three dimensions of $\boldsymbol{X}$ are modeled as being continuously defined, but in practice they are each modelled using the IEEE Standard 754 for floating point numbers. A description of PyKep is provided in papers by the ESA ACT team [21]. The way in which PyKep maps $[x_1, x_2, x_3]$ to $f[x_1, x_2, x_3]$, including its simplifying assumptions that limit it to being a low-to-mid fidelity model, is explained in Appendix E.

The objective function differently is poorly-behaved when projected upon each of the three dimensions (practitioners say, "f is different in each of the three dimensions"). Although f can be queried everywhere in $\boldsymbol{X}$, $\boldsymbol{X}^{\mathbb{F}}$ is sparse in the sense that most values of f are not feasible because of the tight constraints with respect to on-board propellant. In addition, $\boldsymbol{X}^{\mathbb{F}}$ is disconnected. Because $\boldsymbol{X}$ contains an enormous number of points, exhaustively searching for $\boldsymbol{X}^{\mathbb{F}}$, let alone $\min_{\text{global}}(f(\boldsymbol{X}^{\mathbb{F}}))$ is prohibitive. Some of the properties of f that adversely impact convergence include near-singularities due to penalty functions that have been introduced into f to prevent an NLP gradient-search solver (commonly used, but not used here) from encountering singularities that would cause errors and time-consuming fault recoveries.

The output of the MBH optimization is $\boldsymbol{x}^* = [x^*_1, x^*_2, x^*_3]$ that globally minimizes feasible $f[\boldsymbol{x}]$. Point $\boldsymbol{x}^*$ in the space of decision variables is used to generate the least propellent-consuming trajectory to reach Saturn. The $\log_{10}$ values of the objective function $f(\boldsymbol{X})$ in a small neighborhood

around $\boldsymbol{x}^*$ is depicted as three heatmaps in Figure VI.1. Log_{10} values of the objective function $f(\boldsymbol{X})$ are shown because, even in a small neighborhood around $\boldsymbol{x}^*$, the range of $f(\boldsymbol{X})$ is so large that the variations in $f(\boldsymbol{X})$ would be unreadable if a linear scale was used. Figure VI.2 shows linear $f(\boldsymbol{X})$ in the same neighborhood but clipped to show only $f(\boldsymbol{X}) < 10$ where 10 was the NASA Goddard estimate of the upper limit on so-called "Δv" that would make an f, therefore a trajectory, feasible. $f \geq 10$ implies that the mission is "un-flyable" because it would require too much on-board propellant.

The small neighborhood shown in Figures VI.1 and VI.2 is the MBH-found $\text{min}_{\text{global}}(f)$ within +/- 64 days in the date of launch from Earth, the time to the nearest flyby of Juniper, and the time to the nearest flyby of Saturn, respectively. In each heatmap, $\text{min}_{\text{global}}(f)$ is at the center. It is apparent that mis-designing the trajectory by even one day with respect to any of the three decision variables can lead to a sub-optimal solution. Figure VI.2 shows that such a small mis-design of the trajectory can result in a non-feasible solution, which in this case means an "un-flyable" mission. Figure VI.2 also shows that $\boldsymbol{X}^{\mathbb{F}}$ is disconnected and sparse. In both sets of heatmaps, dark blue corresponds to the lowest value of f.

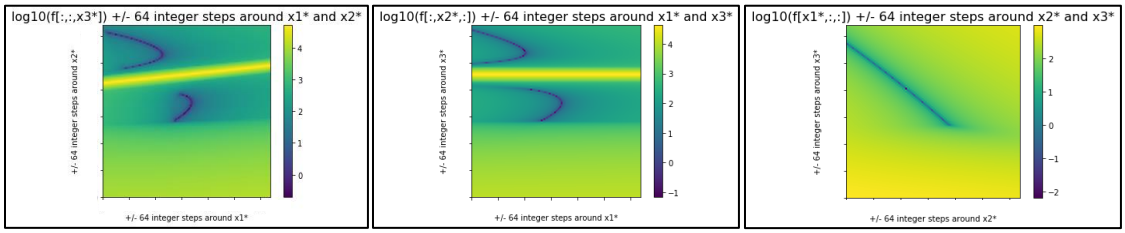


Figure VI.1**:** Heatmaps of $\text{Log}_{10}(f[x_1, x_2, x_3])$ in the +/- 64 3-dimensional neighborhood around x_n^*. The three heatmaps above illustrate the complexities of the hyper-geometry of f even in a small 3-dimensional neighborhood around $\text{argmin}_{\text{global}}(f)$.

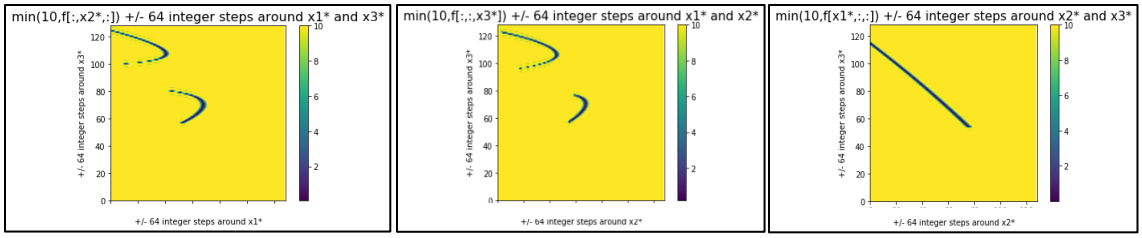


Figure VI.2**:** Heatmaps of min(10,(*f*[x_1, x_2, x_3])) in the +/- 64 3-dimensional neighborhood around x_n*.

The table shown after Figure VI.3 below documents that the methods used in the use-case consistently minimized *f* to values between 0.0003 and 0.0250. Figure VI.3 below shows the *f** = 0.003 Pioneer 11 trajectory found by the use-case MBH, rendered using PyKep. For a sense of perspective, the value of *f* exceeds 15,000 at most points in ***X***.

In Figure VI.3, the z-axis of the optimal trajectory is magnified 5x, making the trajectory to Saturn appear to be more "out-of-plane" with respect to the sub-trajectory from Earth to Jupiter than it really is. Drawing the trajectory in this way is a common aerospace engineering convention used to make it easier to visually inspect for any lack of "smoothness" in the transition from one "leg" of a mission to the next. NASA colleagues who know how to visually inspect such plots reported to the present author that this solution is very "smooth", which is consistent with a very small use of propellant. In this case the 5x magnification of the z-axis is built into the PyKep trajectory rendering software. It is a meaningless coincidence that the trajectory is in three dimensions and the space of decision variables, ***X***, is also in three dimensions.

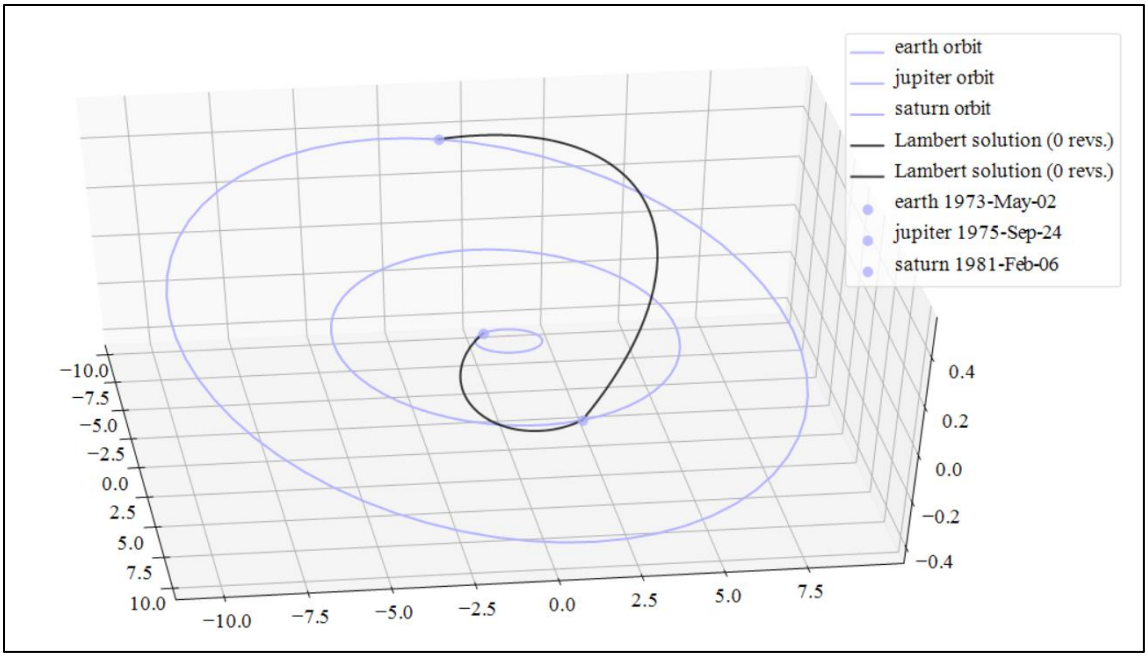


Figure VI.3: The f^* = 0.003 Pioneer 11 trajectory, found by the use-case MBH, rendered using the ESA ACT's PyKep toolkit

The challenging behavior of f and $\boldsymbol{X}^{\mathbb{F}}$, and the 3-dimensionality of X, make this a suitable use-case for the present work. The physics and modeling issues that are the source of the of the challenging behavior of f and $\boldsymbol{X}^{\mathbb{F}}$, as well as the historic nature of the Pioneer 11 mission, makes the present work directly applicable to MBH practitioners in the astronautical engineering community. Members of that community will note that no form of Constrained Local Search including DCLS, the use of which commonly taken for granted, is used at all. Instead, the use-case only involves the adaptation of p to $\hat{q}[t]$ per Chapter IV and MCH per Chapter V.

Pioneer 11 was launched by NASA in April of 1973. Its scientific mission was to study the asteroid belt, as well as solar wind and cosmic rays, around Jupiter and Saturn. Pioneer 11 was the first probe to encounter Saturn, and the second to fly through the asteroid belt and perform a flyby of Jupiter. Pioneer 11 then achieved escape velocity that enabled it to leave the Solar System. Due to power constraints and the vast distance to the spacecraft, the last routine contact with the

spacecraft was on September 30, 1995. The last transmission of scientifically useful data from Pioneer 11 was received on November 24, 1995. Here, Pioneer 11's trajectory refers to the path from Earth to Jupiter, and then Jupiter to Saturn.

Pioneer 11 was equipped with a very small tank for propellant. Therefore, the optimal trajectory is known *a priori* to be one that used very little propellant, which implies a very small integrated total change in velocity due to thrusting, so-called "Δ*v"* in meters per second.

The precision and fidelity of the model used in the current work is not fine enough to determine whether the solution provided here uses more or less fuel than the historic mission. However, the significance of the performance achieved with this use-case, as shown in TABLE III below, is that, because the problem's f and $\boldsymbol{X}^{\mathbb{F}}$ are poorly-behaved, the use-case is a good test of the analytical framework and methods proposed in Chapters IV and V. Those poor-behaviors in f and $\boldsymbol{X}^{\mathbb{F}}$ problem arise from: quasi-periodicities due to the relative positions of the probe, Earth, Jupiter, and Saturn, throughout the mission; penalty functions imposed by the PyKep team to steer NLP solvers (not used here) from encountering singularities in the model (transcription) due to Lagrange's solution to Lambert's problem in orbital mechanics being undefined at transfer angles of 180 degrees. Those poor-behaviors in f are visible in the small neighborhoods around the solution's estimated minimum f, namely $\widehat{f^*}$, shown in the heatmaps in Figure VI.1 above. Here, the transfer angle is the angular change in the z-plane relative to the x/y plane defined by the elliptical orbit from Earth to Jupiter, that is required transfer to the plane that defines the elliptical orbit from Jupiter to Saturn.

TABLE III: Comparison of f^* for various p using the PyKep model for Pioneer 11

p	Number of hoppers	Number of hops	Number of PyKep queries	min(f^*)	max(f^*)	average(f^*)	std. dev(f^*)
Adaptive	16	500	8,000	0.0003	0.0250	0.0067	0.0082
Fixed Laplace-like (leptokurtic)	1	500	500	0.5991	17.5510	4.3793	5.0067
	1	10,000	10,000	0.0242	0.8234	0.2245	0.2541
Fixed Gaussian	1	500	500	2.5302	242.3553	90.8899	83.0177
	1	10,000	10,000	0.7494	15.2358	6.7022	5.0392

A paper on this use-case, co-authored by Arnold C. Englander, Jacob A. Englander, and Michael J. Carter, was presented to the Astrodynamics Specialist Conference of the American Astronautical Society and the American Institute of Aeronautics and Astronautics (AAS/AIAA, both of which are international organizations despite their names). The paper is available through NASA [7].

Among the practical benefits of using European Space Agency's Advanced Concepts Team (ACT) model is that it is server-based, written in C++, and exposed to Python via an Application Programming Interface (API). The implementation described in this paper was coded in Python and only needed to query the server-based ACT model relatively infrequently: a maximum of 500 times per typical MBH optimization of the use-case, for each of 16 simultaneous hoppers, totaling a maximum of 8,000 queries. This enabled an implementation of a 3-dimensional model that has extremely fine granularity (IEEE Floating Point precision) in f and $\boldsymbol{X}$ to perform each MBH in less than a few minutes on a conventional laptop.

The inputs into PyKep are three decision variables, each of which represents a date and time in PyKep coordinates. Each of the three decision variables is bounded so as to stay within a range centered around the corresponding dates of the historical Pioneer 11 mission. The calendar coordinates for the center-points in each dimension are: Launch from Earth on April 6, 1973, 02:11:00 UTC; flyby of Jupiter on December 3, 1974; and flyby of Saturn on September 1, 1979.

The range on x_1 is (-9861.25, -9494.50); the range on x_2 is (0.25, 1000.00); and the range on x_3 is (0.25, 2000.0). All dimensions use units of integer plus fractional days. The bounds on x_1 are set by PyKep. The negative value for x_1 is the result of PyKep using January 1, 2000, 00:00:00 UTC as its zero reference and Pioneer having launched from Earth on April 6, 1973.

Using the adaptive p combined with MCH employing 16 hoppers that communicate as specified in the PSEUDO-CODE below, f^* in the range from 0.0003 to 0.0285, with 10-trial average of 0.0067, was found within fewer than 500 MBH time-steps. Many tens of repeated and independent optimizations, in which all 16 hoppers were independently initialized by a uniform random distribution over $\boldsymbol{X}$, suggest that the performance is not sensitive to the initial positions of the 16 communicating hoppers. The same is suggested by all simulations of MCH illustrated in Chapter V.

PSEUDO-CODE FOR THE PIONEER 11 TRAJECTORY RE-OPTIMIZATION PROBLEM

==========

---

if $t = 0$:

t is the MBH time-step

for m in range (1, Num_hoppers):

Randomly initialize of $\boldsymbol{x}[m,0]$ and evaluate $f[\boldsymbol{x}[m,0]]$

Num_hoppers = 16; $\boldsymbol{x}[m,t]$ is an N-dimensional vector; $f[\boldsymbol{x}[m,t]]$ is a scalar

for t in range(1,Terminus):

for m in range (1, Num_hoppers):

evaluate $f[\boldsymbol{x}[m,t\text{-}1]]$

for n in range (1, Num_dimensions):

draw $\Delta x_{p_{n,m}} \sim p[\lambda[t\text{-}1]$ and generate $\xi[m,t]_n = (x[m,t] + \Delta x_p[m,t])_n$.

$\xi[m,t]_n$ and $(x[m,t] + \Delta x_p[m,t])_n$ are the n^{th} dimensional component of vectors $\boldsymbol{\xi}[m,t]$ and $(\boldsymbol{x}[m,t] + \Delta x_p[m,t])$, respectively

$\Delta x_{p_{n,m}} = \lambda[t\text{-}1] \cdot k_n \cdot \prod_1^3 \delta$ where k_n is a scale factor applied to each of the three dimensions as to their respective ranges, and δ is an independent draw from $\mathcal{N}(0,1)$. $\boldsymbol{\xi}[m,t]$ is an N-dimensional vector comprised of elements $\xi[m,t]_n$ bounded so that $\xi[m,t]_n \subset X$

evaluate $f[\boldsymbol{\xi}[m,t]]$ # $f[\xi[t]]$ is a scalar

if $f[\boldsymbol{\xi}[m,t]] < f[\boldsymbol{x}[m, t\text{-}1]]$:

set $f[\boldsymbol{x}[m,t]] = f[\xi[m,t]]$ and $\boldsymbol{x}[m,t] = \boldsymbol{\xi}[m,t]$

if $f[\boldsymbol{\xi}[m,t]] \geq f[\boldsymbol{x}[m,t\text{-}1]]$:

set $f[\boldsymbol{x}[m,t]] = f[\boldsymbol{x}[m,t\text{-}1]]$ and $\boldsymbol{x}[m,t] = \boldsymbol{x}[m,t\text{-}1]$

$\text{hopper}_{\text{best}} = \text{argmin}_m(f[\boldsymbol{x}[:,t]])$ # : denotes across the M hoppers

for all m<> $\text{hopper}_{\text{best}}$:

$\boldsymbol{x}[m,t] = \boldsymbol{x}[\text{hopper}_{\text{best}}, t]$

$f[\boldsymbol{x}[m,t]] = f\ \boldsymbol{x}[\text{hopper}_{\text{best}}, t]$

$\lambda[t] = ((\max(f[\boldsymbol{x}[:,t]]))/(\max(f[\boldsymbol{x}[:,t-1)]]) + \varepsilon)$

: denotes across the m hoppers and ε is a small positive constant that prevents the denominator from going to zero if $\max(f[x[:,t-1)]]$ ever goes to zero

$t = t + 1$

Because of the low-to-mid level of fidelity of PyKep (due to its simplifying assumptions explained in Appendix E), the present work did not evaluate the performance of the use-case by comparing it to the PyKep f^* for the historical Pioneer 11 mission. Instead, the use-case used PyKep as it was intended by its developers: as a generator of a challenging f, and as a bench-marking tool for comparing different optimization methodologies – here, the adaptive shaping of $p[t]$ per Chapter IV and the biasing of "hop from" locations per Chapter V.

Figures VI.4 and VI.5 illustrate that as the adaptation of $p[t]$ to $\hat{q}[t]$ drives $p[t]$ to become similar to $\hat{q}[t]$, convergence becomes more accurate and speeds up, strongly suggesting that the increased similarity of $p[t]$ and $\hat{q}[t]$ is the source of the MBH performance improvement. Although in Chapter IV, similarity between distributions, in this case $p[t]$ and $\hat{q}[t]$, is formalized using $D_{K\text{-}L}(p, \hat{q})$, that is not possible in the use-case because the MBH approaches solution's estimated minimum f, namely $\widehat{f^*}$, in such a small number of MBH time-steps that the statistically meaningful data required to perform $D_{K\text{-}L}(p,\hat{q})$ could not be obtained. Nonetheless, based on the intuition provided in Chapter IV in Figures IV.3 and IV.5 that the similarity (or dissimilarity) between p and $\hat{q}$, as measured by $D_{K\text{-}L}(p,\hat{q})$, is apparent by visually comparing their histograms, it is clear that $p[t]$ and $\hat{q}[t]$ are not similar in Figure VI.4 but they are similar in Figure VI.5. In Figure V. 4, the visually dissimilar shapes of the histograms is apparent in the right panel, whereas in Figure V. 5, the visually similar shape of the histograms is apparent. In Figure V. 5, also note the speed at which the lower left panel converges to zero, reflecting the speed with which the MBH finds $\widehat{f^*}$.

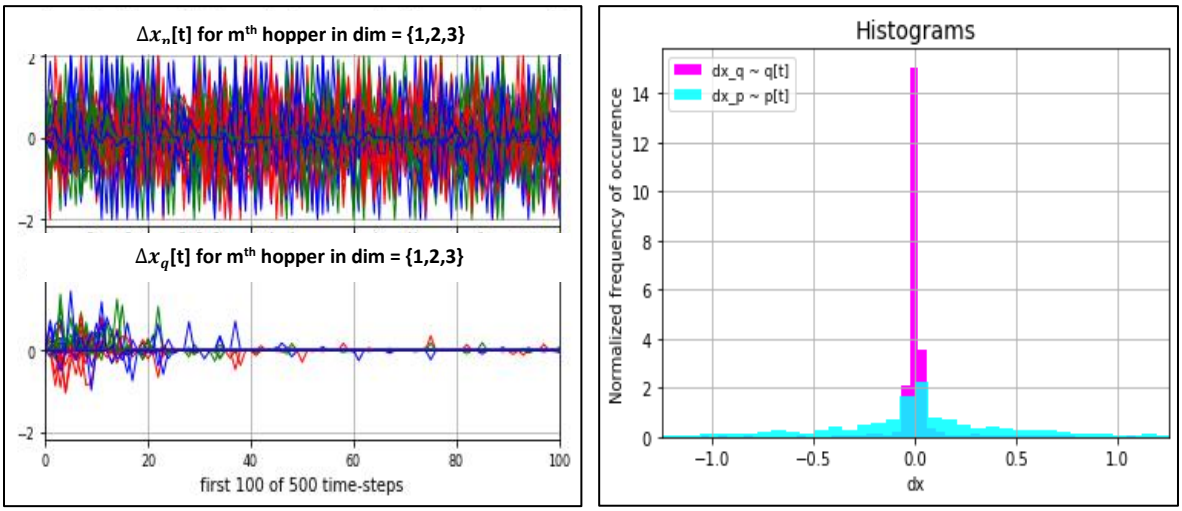


Figure VI.4:

Left upper panel: $\boldsymbol{\Delta x_p}$[t] ~ p[t] not adapted to $\hat{q}[t]$ (and not using MCH).

Left lower panel: $\boldsymbol{\Delta x_q}$[t] ~ $\hat{q}[t]$ (not using MCH).

In the left panel, the colors red, green, and blue correspond to the three dimensions of $\boldsymbol{X}$.

Right panel: Histograms of $\boldsymbol{\Delta x_p}[t]$ (cyan) and $\boldsymbol{\Delta x_q}[t]$ (magenta), respectively

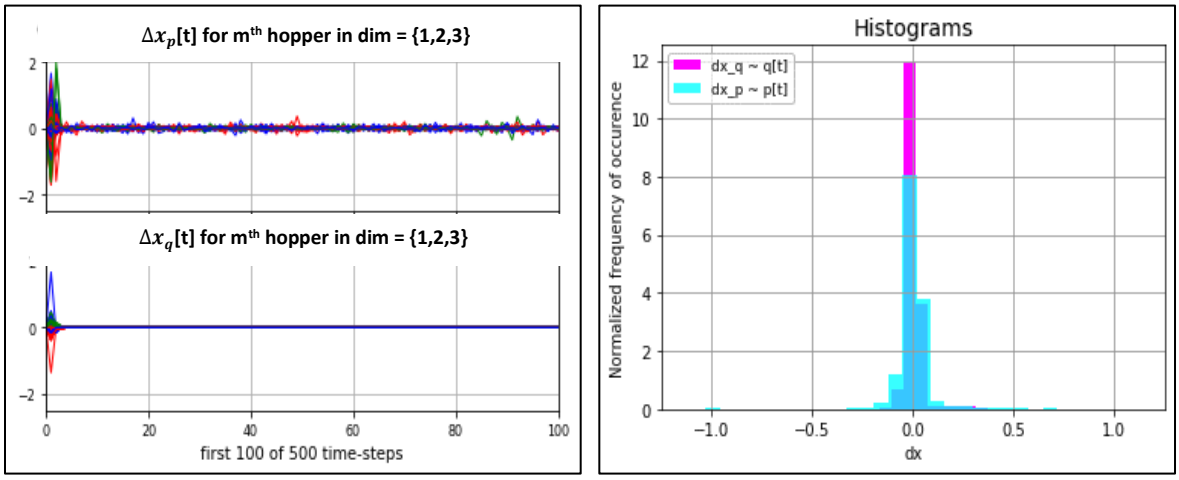


Figure VI.5:

Left upper panel: $\boldsymbol{\Delta x_p}$[t] ~ p[t] adapted to $\hat{q}[t]$ (while using MCH– the impact of which, in this graphic, is barely visible or significant). Left lower panel: $\boldsymbol{\Delta x_q}$[t] ~ $\hat{q}[t]$ (while using MCH – the impact of which, in this graphic, is barely visible or significant).

In the left panel, the colors red, green, and blue correspond to the three dimensions of $\boldsymbol{X}$

Right panel: Histograms of $\boldsymbol{\Delta x_p}$[t] (cyan) and $\boldsymbol{\Delta x_q}$[t] (magenta), respectively

Figure VI.6 illustrates the speed-up achieved by using adaptive *p* and MCH as opposed to using a fixed Laplace-like *p* and no MCH. Although the fixed Laplace-like *p* was designed to be well-suited to *f* in the sense explained in Chapter IV, the upper panel of Figure VI.6 shows that whereas, using fixed Laplace-like *p*, MBH does not drive $\widehat{f^*}$ below 16.066 (a non-feasible solution) until after 500 time-steps (500 PyKep queries), In contrast, the lower panel shows that by using adaptive *p* and 16-hopper MCH, the MBH converges to 0.0097 in 7 MBH time-steps (112 queries to PyKep). Thus, it is clear that using adaptive *p* and 16-hopper MCH not only speeds up MBH but also provides a much better (much smaller, feasible rather than non-feasible) solution.

When comparing the upper and lower plots of Figure VI.6, recall that the objective function *f* is exactly the same in both cases. Note too that not only does the search driven by adaptive *p* converge faster, but that the non-adaptive *p* results in $f\,[x[t>t_{Convergence}]] = 16.0655$, whereas the adaptive *p* results in $f\,[x[t>t_{Convergences}]] = 0.0097$. This indicates that the convergence in the case of the adaptive *p* is not only faster, it also results in a better solution.

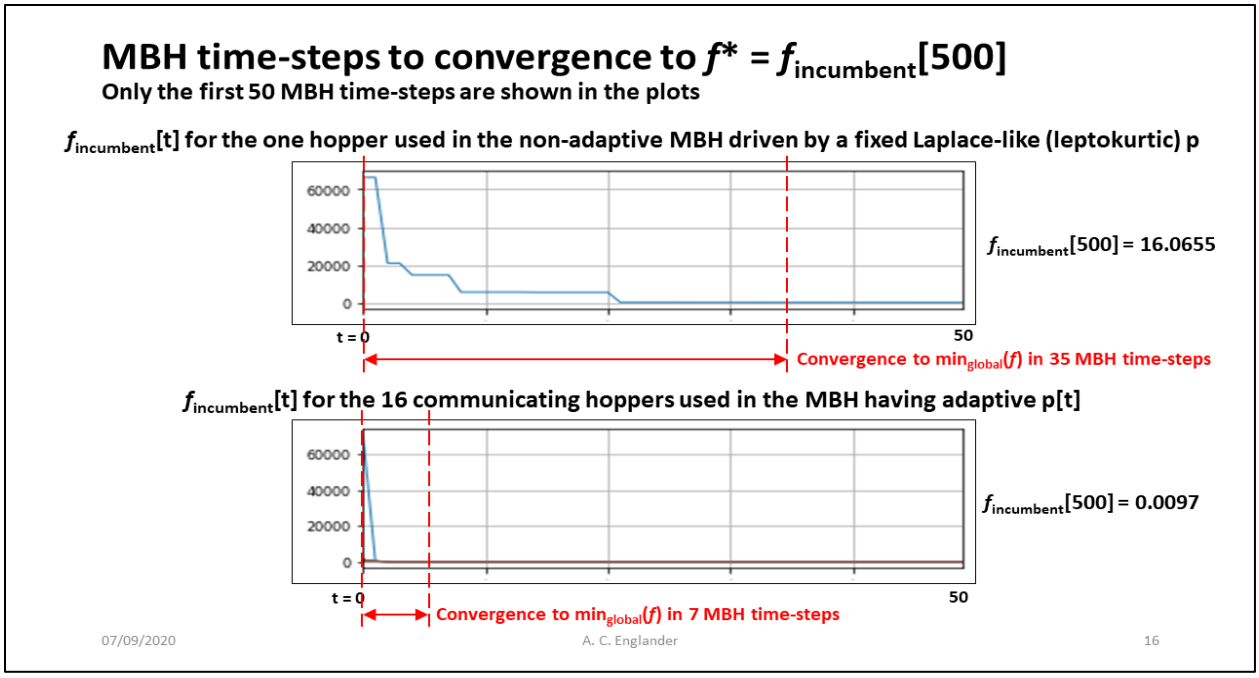


Figure IV.6. The progress of the MBH, in MBH time-steps, toward convergence to $\min_{global}(f)$
Upper plot: A single f_incumbent[t] also referred to as $f[\boldsymbol{x}[t]]$, being operated upon by an MBH using a fixed, non-adaptive Laplace-like p, converging toward $\min_{global}(f)$ and arriving in approximately 35 MBH time-steps
Lower plot: f_incumbent[m,t] also referred to as $f[\boldsymbol{x}[m,t]]$, for $m = \{1,2,3,\ldots,16\}$ hoppers, being operated upon by an MBH using the adaptive p, converging toward $\min_{global}(f)$ and arriving in approximately 7 MBH time-steps

When this use-case was presented to the Astrodynamics Specialist Conference of the American Astronautical Society and the American Institute of Aeronautics and Astronautics (AAS/AIAA) in early August 2020, participants found the performance results to be remarkable. During the second half of 2021, the present author looks forward to working with practitioners to empirically test the methods of the present work's Chapters IV and V on real high-dimensional inter-planetary spacecraft trajectory optimization problems.

VII. OPEN QUESTIONS AND HYPOTHESES

This chapter poses the following two open question and provides hypotheses that the present author hopes will prove useful when answers for them are pursued: (1) How to formally explain the strong fit of Gamma distributions as FPTDs of MBH FPTs; and (2) How to develop a method for detecting and characterizing "poorly-behaved" f and/or disconnected, sparse $\boldsymbol{X}^{\mathbb{F}}$ in near real-time, autonomously, by an ongoing MBH as a way of alerting the user that convergence may be slow and suggesting that additional speed-up methods be applied and/or that the project deadline be revised.

Regarding (1): Throughout the present work, Gamma distributions have been fit to MBH FPTs as FPTDs as a way of characterizing MBH convergence times and their variability on various f and $\boldsymbol{X}^{\mathbb{F}}$, as well as their speed-up by various methods. Moreover, in Appendix B the high quality of those fits to Gamma distributions are examined and the usefulness of those fits is expounded upon. That raises the natural question of why Gamma distributions fit so well as FPTDs of MBH FPTs.

This is a difficult question, if only because the variation in the FPTs depends on f, $\boldsymbol{X}^{\mathbb{F}}$, the distribution p from which the $\Delta\boldsymbol{x}$ are drawn, and the path of $\boldsymbol{x}[t]$. This deserves a level of analysis that the present author has been unable to achieve.

The present author's hypothesis is that MBH FPTs may be perturbed-Poisson distributed events, generally referred to as Renewal Processes [63]. The fact that they are not Poisson distributed can easily be seen by their mean not being equal to their variance, which is a requirement for Poisson-distributed random variables. If MBH FPTs were Poisson distributed, their FPTD (or FPT-PMF) being Gamma distributed can be shown by a derivation that involves a

large number of algebraic steps but is conceptually straight forward. But since MBH FPTs are not Poisson distributed, the Gamma distribution-like shape of MBH FPTDs cannot be attributed to the fact that Gamma distributions are the density function for Poisson-distributed random variables. Nonetheless, generalized Poisson-distributed processes, referred to as Renewal processes, are not required to have their mean equal their variance. The formal definition of a Poisson process is a set of random events (in this case arrivals) that are exponentially distributed. A Renewal processes is a set of random events (in this case arrivals) that may come from any distribution, including an exponential distribution to which Gaussian noise has been added as a perturbing process [64]. In that case, there are analytical and numerical results that show that the density function resembles a Gamma distribution – at least for reasonable large values of the independent variable. i.e., *t* in the case of the random variables being FPTs. That literature can be used to shed some light on why Gamma distributions fit so well as the FPTD of collections of MBH FPTs to the extent that one is willing to model MBH FPTs as a Renewal Process that is a Poisson-distributed process perturbed by additive Gaussian noise. The present author hypothesizes that it is not unreasonable to model MBH FPTs as mixture models of a Poisson distributed process and additive Gaussian noise. However, the empirical evidence in Figure VII.2 below, compared to the empirical evidence in Appendix B that uses actual MBH FPT data, suggests that modeling MBH FPTs as a Poisson process with additive Gaussian noise is not entirely appropriate.

Figure VII.1 below shows empirical evidence from numerical simulations that illustrate random variables from a Poisson-distributed process used as surrogates for MBH FPTs, their histogram, and their fit to Gamma distribution as their density function. It is well-known that a Gamma distribution is the density function for Poisson-distributed random variables, and it can be shown algebraically. Figure VII.2 below shows empirical evidence from a numerical simulation

that illustrates random variables from a Renewal process used as surrogates for MBH FPTs, their histogram, and their fit to Gamma distribution as their density function. The Renewal process used was a mixture model composed of a Poisson distributed process and a perturbing process is comprised of additive Gaussian noise. It is clear that the Gamma distribution fit to the random variables used as surrogates for MBH FPTs resembles the histogram. However, here the Gamma distribution does not resemble the histogram as closely as Gamma distributions fit histograms of MBH FPTS throughout the present work, as is especially evident in Appendix B.

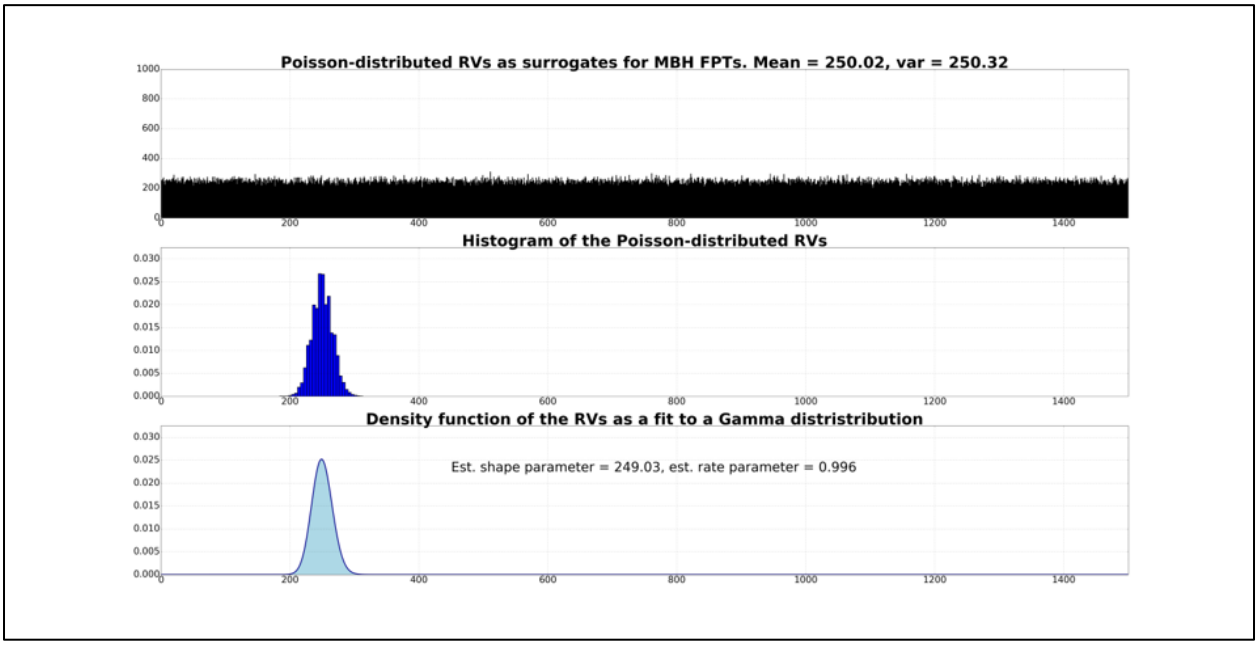


Figure VII.1:
Upper panel: 1,600 Poisson-distributed random variables used as surrogates for FPTs
Middle panel: The histogram (PMF) of the 1,600 Poisson-distributed random variables
Lower panel: The fit of a Gamma-distribution to the 1,600 Poisson-distributed random variables as their approximated FPTD

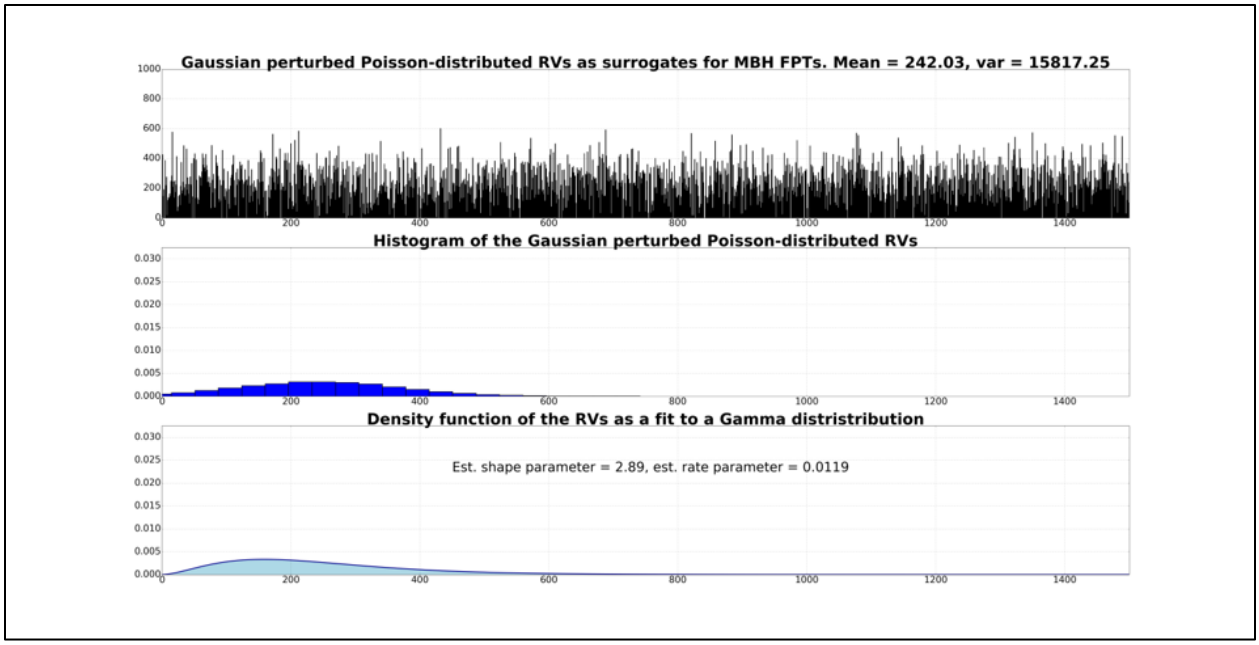


Figure VII.2:
Upper panel: 1,600 Poisson-distributed random variables perturbed by Gaussian noise, used as surrogates for FPTs
Middle panel: The histogram (PMF) of the 1,600 Poisson-distributed random variables perturbed by Gaussian noise
Lower panel: The fit of a Gamma-distribution to the 1,600 Poisson-distributed random variables perturbed by Gaussian noise as their approximated FPTD

Regarding open question (2): The present work provides multiple methods for speeding-up MBH and shows that different methods are more and less effective on different f and $\boldsymbol{X}^{\mathbb{F}}$. That raises the possibility that different methods could be systematically applied, autonomously by an on-going MBH, and used to classify different f and $\boldsymbol{X}^{\mathbb{F}}$ by how effectively the MBH operating upon them is sped-up. This would require the construction of equivalence classes of f that are classified according to how they respond to various MBH speed-up methods, but the present work provides some evidence that suggests such equivalence classes may exist and be separable to a useful extent. Pattern recognition methods written about by Duda and Hart and more recently by Bishop, could be used to formulate the classification process [65, 66]. Such an approach to

diagnosis by trials of various treatments is used in medicine, albeit within an empirical framework [67]. In this case, the features would be performance scores representing the extent to which methods in Chapter IV and V resulted in speeding-up the convergence rate (the minimization of f) during a sample period while the MBH is in progress. Those performance scores would need to be realizable, such as can be derived from $f[\boldsymbol{x}[t]]$, with or without SCLS or MCH, even though the scores could not be based on FPTS and FPTDs because the MBH has not yet converged and, more important, because f^* is unknown.

In addition, Appendix C, which addresses the probability of advancement for an MBH operating on a known 1-dimensional Gibsonian f while using a known p, raises the question: Since the probability of advancement can be determined at each depth d given knowledge of d, f and p, could d be inferred from the running statistics of the advancement of an on-going MBH if f is known to belong to an equivalence class of convergence time-impacting f ? Likewise, can the equivalence class of convergence time-impacting f to which the f being operated belongs, be inferred from running statistics empirically describing the probability of advancement of an on-going MBH? While these questions could not be addressed within the scope of the present work, the questions are intriguing to practitioners because they point to possible approaches to developing some kind of "progress indicator" or indicator of "remaining time to convergence" for MBH. Unlike MBH researchers whose work is analytical and/or simulation-based, and FPTs and FPTDs can be derived because f, $\boldsymbol{X}^{\mathbb{F}}$, and f^* are known by construction, MBH practitioners constantly face the possibility of stopping an MBH process either too soon or long after it has already found the optimal (or near-optimal) solution. Therefore, to practitioners, some kind of "progress indicator" or indicator of "remaining time to convergence" would be of great practical value.

VIII. SUMMARY

The present work has provided the first systematic investigation of the convergence rates of MBH and ways to speed them up. Two approaches to speeding up MBH have been provided: biasing the shape of the probability distribution from which hop distances are drawn; and biasing the location from which each next hop is taken. Further, two stochastic methods are provided for biasing the location from which each next hop is taken: SCLS and MCH. While the methods for speeding-up MBH are variably effective on different f and $\boldsymbol{X}^{\mathbb{F}}$, nonetheless some combination of methods is shown to be highly effective on each of the f and $\boldsymbol{X}^{\mathbb{F}}$ investigated in the present work by simulation experiments and the Pioneer 11 spacecraft trajectory optimization use-case.

Formal abstract concepts including the accepted hop distribution and the remaining productive search volume $g[d]$ have been introduced and developed, as have been algorithmically realizable concepts including the estimation of $\hat{q}$ and $\hat{q}[t]$, and the adaptation of p to $\hat{q}[t]$. The analytical framework used includes $D_{K\text{-}L}(p, \hat{q})$, MBH FPTs and the fit, thereby the parameterization, of Gamma distributions to FPTs as their FPTD.

The novel analytic contributions are the building blocks used to state and explain the impact on the convergence rate of an MBH operating on an f and $\boldsymbol{X}^{\mathbb{F}}$, of biasing the shape of p, the distribution from which hop distances are drawn, and biasing the location from which each next hop is taken. These statements and explanations involved novel formalisms and/or novel uses of existing formalisms. In addition, these statements and explanations resulted in novel contributions to engineering practice.

The explanation of the impact of biasing the shape of p involved*:* Conjecturing the existence of a distribution q comprised of hop distances $\boldsymbol{\Delta x}$ such that $f[\boldsymbol{x}[t]+\boldsymbol{\Delta x}] < f[\boldsymbol{x}[t]]$ and are

therefore “accepted” by the MBH algorithm in the sense that they move the process toward the global minimum of f rather than forcing the process to wait-in-place; estimating, by means of a Monte Carlo method, $\hat{q}$; using the Kullback-Leiber divergence $D_{K\text{-}L}(p, \hat{q})$ to measure the similarity between a p and $\hat{q}$; using the correlation between MBH FPTs, and their approximate FPTD, and $D_{K\text{-}L}(p, \hat{q})$ to show that making p more similar to $\hat{q}$ speeds-up the convergence of the MBH; and, because $\hat{q}$ is time-varying over the course of the MBH process, adapting $p[t]$ to $\hat{q}[t]$ in near real-time can result in making p similar to $\hat{q}$ not only across the MBH process but within various stages of the MBH process, thereby further speeding up the convergence of the MBH. As a guide for developing the linkages between each of these building blocks, the present author developed $g[d]$ as a measure of the remaining volume, at each objective function depth d, that contains yet smaller values of f. This is the contribution of Chapter IV.

The explanation of the impact of biasing the location from which each next hop is taken, is explained colloquially in Chapter V and rigorously in Appendix C. Chapter V provides two methods for speeding-up the convergence of the MBH by biasing the location from which each next hop is taken. They have different properties in some regards, e.g., their computational requirements as dimensionality grows, and they have similar properties in other regards, e.g., they both minimize the variance across FPTs corresponding to independent trials of MBH operating on the same f and, using the same p, as well as speeding-up the convergence time of the MBH operating on that same f and using that same p.

The approaches and methods provided in Chapters IV and V are also differently effective on different f and $\boldsymbol{X}^{\mathbb{F}}$. This is easy to show empirically by simulation experiments, especially by using Gibsonian f for which MBH is sped-up significantly by biasing the location from which

each next hop is taken but insignificantly by biasing the shape of *p*, the distribution from which hop distances are drawn. The analytical explanation, however, is provided by Appendix C.

The present work's focus on speeding-up MBH is enabled and informed by the proof of asymptotic convergence of global optimization based on random search, by Baba et. al (1997), shown to be applicable to MBH by the present author. This application of the proof by Baba *et. al* provided the MBH research and practitioner community with the previously missing guarantee that if mild conditions are satisfied, the question is not whether MBH will converge, but when. In addition, it focused the present author on the question of why convergence takes as long as it takes, and how to speed it up.

The novel contributions to engineering practice include two approaches (three methods) for speeding up MBH. The three methods can be used separately or in combination. Their effectiveness has been demonstrated in simulations and in the Pioneer 11 spacecraft optimization use-case. In the use-case, the effectiveness of combining the first and third method was demonstrated on a real 3-dimensional problem.

While the present work is restricted to 1-dimensional, 2-dimensional, and 3-dimensional problems, colleagues of the present author are already exploring implications of the present work in their engineering practice using problems that are high dimensional. In addition, question two in Chapter VI is highly relevant to practitioners and was inspired by the present author's discussions with practitioners at NASA Goddard Space Flight Center over the past several years.

IX. REFERENCES

[1] Adams, D., Ozemik, M., Scot, C., Preliminary Interplanetary Mission Design and Navigation for the Dragonfly New Frontiers Mission Concept, August 2018, 2018 AAS/AIAA Astrodynamics Specialist Conference, Snowbird, Utah, https://www.researchgate.net/publication/327110307_Preliminary_Interplanetary_Mission_Design_and_Navigation_for_the_Dragonfly_New_Frontiers_Mission_Concept

[2] Afanasiev, A., Oferkin, I., Posypkin, M., Rubtsov, A., Sulimov, A., and Sulimov, V. (2011), A Comparative Study of Different Optimization Algorithms for Molecular Docking, 3rd International Workshop on Science Gateways for Life Sciences (IWSG 2011), 8-10 JUNE 2011

[3] Bak, P., Sneppen, K., (1993). Punctuated equilibrium and criticality in a simple model of evolution. Physical Review Letters. **71** (24): 0834086. doi:10.1103/PhysRevLett.71.4083. PMID 10055149.

[4] Jain, P., Non-convex Optimization for Machine Learning, Non-convex Optimization for Machine Learning (prateekjain.org). See also Jain, P., Kar, P., Non-convex Optimization for Machine Learning. Foundations and Trends in Machine Learning, vol. 10, no. 3-4, pp. 142-336, 2017

[5] Li, Y., Introduction to Non-convex optimization, a lecture based on "Neon2" by Zeyuan Allen-Zhu and Yuanzhi Li, [1711.06673] Neon2: Finding Local Minima via First-Order Oracles (arxiv.org)

[6] Englander, J. A. and Englander, A. C., (2014), Tuning Monotonic Basin Hopping: Improving the Efficiency of Stochastic Search as Applied to Low-Thrust Trajectory Optimization, *Proceedings of the 24th International Symposium on Space Flight Dynamics*, Laurel, MD

[7] Englander, A. C., Englander, J. A. and Carter, M. J, Hopping with an Adaptive Hop Probability Distribution, July 27, 2020, *Proceedings of the Astrodynamics Specialist Conference of the American Astronautical Society*, Document 20205005121, NASA Technical Reports Server (NTRS)

[8] Ellison, D. H., Conway, B. A., Englander, J. A. and Ozimek, M. T., (2018), Analytic Gradient Computation for Bounded-Impulse Trajectory Models Using Two-Sided Shooting. *Journal of Guidance, Control, and Dynamics*, Volume 41, No. 7, pages 1449 – 1462

[9] Ellison, D. H., Conway, B. A., Englander, J. A. and Ozimek, M. T., (2018), Application and Analysis of Bounded-Impulse Trajectory Models with Analytic Gradients, Journal of Guidance, Control, and Dynamics, Published Online:12 Apr 2018, https://doi.org/10.2514/1.G003078, https://arc.aiaa.org/doi/10.2514/1.G003078

[10] Wales, D. J. and Doye, J. P. K., Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 Atoms. *Journal of Physical Chemistry*. A, 101, 5111-5116 (1997).

[11] Baba, N., Showman,T., and Sawaragi,Y., (1977), A Modified Convergence Theorem for Random Optimization Algorithm, *Information Science*, No. 13, (1977)

[12] Leary, R. H., (2000), Global Optimization on Funneling Landscapes. *Journal of Global Optimization*, Volume 18, No. 4, pages 367 – 383

[13] Locatelli, M., and Schoen, F., (2003), Efficient Algorithms for Large Scale Optimization: Lennard-Jones Clusters. *Computational Optimization and Applications* 26 (2003): 173-190.

[14] Vasile, M., Minisci, E. and Locatelli, M., (2010), Analysis of Some Global Optimization Algorithms for Space Trajectory Design. *Journal of Spacecraft and Rockets*, Volume 47, No. 2, pages 334 – 444

[15] Yam, C.H., Lorenzo, D. D. and Izzo, D., (2011), Low-thrust trajectory design as a constrained global optimization problem. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Volume 225, No. 11, pages 1243 – 1251

[16] Olson, B., Hashmi, I., Molloy, K., Shehu, A., (2012), Basin Hopping as a General and Versatile Optimization Framework for the Characterization of Biological Macromolecules, *Advances in Artificial Intelligence* Volume 2012, Article ID 674832, 19 pages doi:10.1155/2012/674832

[17] Englander J.A., Conway B.A., An Automated Solution of the Low-Thrust Interplanetary Trajectory Problem. *Journal of Guidance, Control, and Dynamics*. 2017; 40(1):15-27. doi:10.2514/1.G002124

[18] Englander J.A., Conway B.A., Williams, T., Automated Mission Planning via Evolutionary Algorithms. *Journal of Guidance, Control, and Dynamics*. 2012; 35 (6):1878-1887

[19] Goddard Fellows Innovation Challenge (GFIC) 2017-2020 Proposal Selections, https://seniorfellows.gsfc.nasa.gov/gfic-selections.html

[20] Knittle, J., Hughes, K., Englander, J., and Sarli, B., (2017), Automated Sensitivity Analysis of Interplanetary Trajectories, *Proceedings of the International Symposium on Space Flight Dynamics*, June 8th, 2017, Japan; published in *Trans. JSASS Aerospace Tech*. Japan, Vol. 14, No. ists 31, pp. Pd1 - Pd8, 2017

[21] Izzo, D., (2019, February 22). esa/pykep: (Version v2.3). Zenodo. http://doi.org/10.5281/zenodo.2575462

[22] Solis, F. J., and Wets, R. J. B (1981), Minimization by Random Search Techniques, *Mathematics of Operations Research*, Vol. 6: 19 – 30 (1981)

[23] Aghassi, M., *Minimization by Random Search Techniques* by Solis and Wets, and Introduction to Sampling Methods, published slides from a lecture presented in 2003

[24] Luus, R., Jaakola, T.H.I. (1973). "Optimization by direct search and systematic reduction of the size of search region". AIChE Journal. 19 (4): 760–766. doi:10.1002/aic.690190413 (https://doi.org/10.1002%2Faic.690190413). 3.

[25] Bojkov, R.; Hansel, B.; Luus, R. (1993). "Application of direct search optimization to optimal control problems". Hungarian Journal of Industrial Chemistry. 21: 177–185. 4.

[26] Spaans, R., Luus, R. (1992). "Importance of search-domain reduction in random optimization". Journal of Optimization Theory and Applications. 75: 635–638. doi:10.1007/BF00940497 (https://doi.org/10.1007%2FBF00940497). MR 1194836 (https://www.ams.org/mathscinet-getitem?mr=1194836).

[27] Luus, R. (2010). "Formulation and Illustration of Luus-Jaakola Optimization Procedure". In Rangalah, Gade Pandu (ed.). Stochastic Global Optimization: Techniques and Applications in Chemical Engineering. World Scientific Pub Co Inc. pp. 17–56. ISBN 978-9814299206. 9. Nemirovsky & Yudin (1983, p. 7)

[28] Nair, G. Gopalakrishnan (1979). "On the convergence of the LJ search method". Journal of Optimization Theory and Applications. 28 (3): 429–434. doi:10.1007/BF00933384 (https://doi.o rg/10.1007%2FBF00933384). MR 0543384 (https://www.ams.org/mathscinet-getitem?mr=054 3384).

[29] Nemirovsky, A. S.; Yudin, D. B. (1983). Problem complexity and method efficiency in optimization. WileyInterscience Series in Discrete Mathematics (Translated by E. R. Dawson from the (1979) Russian (Moscow: Nauka) ed.). New York: John Wiley & Sons, Inc. pp.

[30] Davidon, W.C. (1991). "Variable metric method for minimization". *SIAM Journal on Optimization*. **1** (1): 1–17. CiteSeerX 10.1.1.693.272. doi:10.1137/0801001.

[31] Hooke, R.; Jeeves, T.A. (1961). ""Direct search" solution of numerical and statistical problems". *Journal of the ACM*. **8** (2): 212–229. doi:10.1145/321062.321069

[32] Torczon, V.J. (1997). "On the convergence of pattern search algorithms" (PDF). *SIAM Journal on Optimization.* **7** (1): 1–25. CiteSeerX 10.1.1.50.3173. doi:10.1137/S1052623493250780.

[33] Dolan, E.D.; Lewis, R.M.; Torczon, V.J. (2003). "On the local convergence of pattern search" (PDF). *SIAM Journal on Optimization.* **14** (2): 567–583. CiteSeerX 10.1.1.78.2407. doi:10.1137/S1052623400374495.

[34] Pincus, Martin (1970). "A Monte-Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems". *Operation Research.* Vol. 18 No. 6: 1225–1228. doi:10.1287/opre.18.6.1225 – via JSTOR.

[35] Kirkpatrick, S.; Gelatt Jr, C. D.; Vecchi, M. P. (1983). "Optimization by Simulated Annealing". *Science*. **220** (4598): 671 680. Bibcode:1983Sci...220..671K. CiteSeerX 10.1.1.123.7607. doi:10.1126/science.220.4598.671. JSTOR 1690046. PMID 17813860.

[36] Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M., N.; Teller, A. H.; Teller, E. (1953). "Equation of State Calculations by Fast Computing Machines". *The Journal of Chemical Physics*. **21** (6): 1087. Bibcode:1953JChPh..21.1087M. doi:10.1063/1.1699114.

[37] Hastings, W.K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". *Biometrika*. ***57*** *(1): 97–109.* *Bibcode:1970Bimka..57...97H*. *doi:10.1093/biomet/57.1.97*. *JSTOR 2334940*. *Zbl 0219.65008*.

[38] Rosenbluth M.N. (2003). "Genesis of the Monte Carlo Algorithm for Statistical Mechanics". *AIP Conference Proceedings*. **690**: 22–30. doi:10.1063/1.1632112.

[39] Schneider, J. J., and Kirkpatrick, S., *Stochastic Optimization*. Springer, New York, 2006

[40] Spall, J. *Introduction to Stochastic Search and Optimization*. Wiley, Hoboken, NJ. 2003

[41] Dill, K., Chan, H. From Levinthal to pathways to funnels. *Nat Struct Mol Biol* **4,** 10–19 (1997). https://doi.org/10.1038/nsb0197-10

[42] Dill, K., Bromberg, S. *Molecular Driving Forces.* 2nd Edition. Garland Science, Taylor and Francis Group, LLC., New York, NY. 2011

[43] Wales, D. J. *Energy Landscapes*. Cambridge University Press. Cambridge. UK and New York, NY. 2003.

[44] Chaplin, M. Water structure and Science (2010), http://www1.bbu.ac.uk/water/vibrat.html. Available through: Download Water Structure and Science by Martin Chaplin (engineering108.com)

[45] Huang, K. *Lectures on Statistical Physics and Protein Folding*. World Scientific Publishing Co. Pte. Ltd. Singapore, and Hackensack, NJ. 2005

[46] Samuelson, Paul A. (1950). "The problem of integrability in utility theory". *Economica (New Series)*. 17. pp. 355–385. doi:10.2307/2549499. See pages 359-360.

[47] Østman B, Adami C (2013). Predicting evolution and visualizing high-dimensional fitness landscapes. *In "Recent Advances in the Theory and Application of Fitness Landscapes*" (A. Engelbrecht and H. Richter, eds.). Springer Series in Emergence, Complexity, and Computation, 2013. Also see*:* http://pleiotropy.fieldofscience.com/2013/11/smooth-and-rugged-fitness-landscapes.html#sthash.vMFV620p.dpuf

[48] Sole, R., and Goodwin, B., *Signs of Life: How Complexity pervades biology*, 2000, Basic Books, New York, NY

[49] Balents, L., Bouchaud, J-P., and Mezard, M., The Large Scale Energy Landscape of Randomly Pinned Objects, arXiv:cond-mat/9601137v1 29 Jan 1996

[50] Cover, T.M., and Thomas, J.A., *Elements of information theory*. Wiley, New York, 1991

[51] Jonathon Shlens, Google Research, 2014, arXiv:14042000v1 [cs.IT] 8 Apr 2014

[52] Hogg, R.V. and Craig, A.T., (1978). *Introduction to Mathematical Statistics* (4th edition). New York: Macmillan

[53] Papoulis, A. *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill, pp. 103-104, 1984

[54] Ye, Z-S., and Chen N., (2017) Closed-Form Estimators for the Gamma Distribution Derived from Likelihood Equations, The American Statistician, 71:2, 177-181

[55] Louzada, F., Ramos, P. L. Ramos, E.. (2019) A Note on Bias of Closed-Form Estimators for the Gamma Distribution Derived From Likelihood Equations. The American Statistician 73:2, pages 195-199.

[56] P.E. Gill; W. Murray; M.A. Saunders (2005). "SNOPT: An SQP algorithm for large-scale constrained optimization". Available on-line via the likn provided here.

[57] Private communications between Jacob Englander and the present author, November 2020 through February 2021

[58] Robbins, H. and Monro, S., A stochastic approximation method. Ann. Math. Statist. 22, 400-407 (1951)

[59] Kiefer, E., and Wolfowitz, J., Stochastic estimation of the maximum of a regression function.. Ann. Math. Statist. 23, 462-466 (1952)

[60] Borkar, V. S., *Stochastic Approximation: A Dynamical Systems Viewpoint*, 2008, Cambridge University Press, Cambridge, and Hindustan Book Agency, New Delhi

[61] Bhatnagar, S., Prasad, H. L., and Prashanth, L. A., , *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*, 2013 Lecture Notes in Control and Information Sciences, Springer-Verlag, London

[62] Spall, J. C., Multivariate stochastic approximation using simultaneous perturbation gradient approximation. IEEE Trans. Auto. Cont. 37(3), 332-341 (1992)

[63] Doob, J. L. (1948). "Renewal Theory From the Point of View of the Theory of Probability" (PDF). Transactions of the American Mathematical Society. **63** (3): 422–438. doi:10.2307/1990567. JSTOR 1990567.

[64] Wang, W., Schulz, J. P., Deng, W., and Barkai, E. (2018). "Renewal theory with fat-tailed distributed sojourn times: Typical versus rare". Phys. Rev. E. **98** (4): 042139. arXiv:1809.05856. Bibcode:2018PhRvE..98d2139W. doi:10.1103/PhysRevE.98.042139.

[65] Duda R. O. and Hart, P.E., *Pattern Analysis and Scene Recognition*, 1973, Wiley using simultaneous perturbation gradient approximation. IEEE Trans. Auto. Cont. 37(3), 332-341 (1992)

[66] Bishop, C., *Pattern Recognition and Machine Learning*, 2013, Springer, Singapore. Pdf version available at Bishop - Pattern Recognition and Machine Learning.pdf (google.com)

[67] Maxwell, S. R. J., Rational prescribing: the principles of drug selection, *Clinical Medicine Journal*, Royal College of Physicians, October 2016, Available at: Rational prescribing: the principles of drug selection | RCP Journals

[68] Izzo, D., Revisiting Lambert's problem, *Celestial Mechanics and Dynamical Astronomy*, Volume 121, pages 1-15 (2015), https://doi.org/10.1007/s10569-014-9587-y

[69] Izzo, D., Hennes, D., Simoes, L. F., amd Martens, M., Designing Complex Interplanetary Trajectories for the Global Trajectory Optimization Competitions, ArXiv:1511.00821v3 [physics.space-ph] (2015)

[70] Di Lizia, P., Radice, G, and Izzo, D., Advanced Global Optimization Tools for Mission Analysis and Design, available on the ACT net (www.esa.int/act)

[71] Izzo, D., Global Optimization and Space Pruning for Spacecraft Trajectory Design, in *Spacecraft Trajectory Optimization*, Conway, B. (editor), Cambridge University Press, New York, NY, 2010

X. APPENDICES

A. Asymptotic convergence proof by Baba et al., applicable to MBH

The following is based on an asymptotic convergence proof by Baba, Shoman, and Sawaragi (1977) that was generalized by Solis and Wets (1981).

Conceptual Algorithm:

Here, all notation has been revised to align with the notation used in the analytical framework presented above, including replacing x^0 with $x[0]$, ξ^k with $\xi[t]$, x^k with $x[t]$, *S* with *X*, and probability measure μ^k with $p[t]$. In the present work, *t* is the discretized index of successive MBH operations, the same as *k* is in the asymptotic convergence proofs by Baba, Shoman, and Sawaragi, and Solis and Wets. *X* replaces *S*, not only to align the notation with the analytic support provided above, but because MBH is typically thought of as operating on an f having a domain that is hyper-cube rather than a hyper-sphere. No changes in meaning or loss of generality resulted from these changes in notation.

1. Choose $x[0] \in X$. Set $t := 0$
2. Generate $\xi[t] \in \mathbb{R}^n$ (random) from distribution $p[t]$. More precisely, in order to adapt this asymptotic convergence proof to MBH, generate $\xi[t] = (x[t\text{-}1] + \Delta x_p[t])$, $\Delta x_p[t]$ random, $\Delta x_p[t] \sim p[t]$, $\xi[t] \in X$, $f[\xi[t]] \in \mathbb{R}$
3. Set $x[t] = D(x[t\text{-}1], \xi[t])$. In order to adapt this asymptotic convergence proof to MBH, define D as the operation of choosing $\xi[t]$ if $f[\xi[t]] < f[x[t\text{-}1]]$, otherwise choose $x[t\text{-}1]$
4. Go to step 1

This provides both local search and global search.

Local search $\Longrightarrow$ *supp*(*p*[*t*]) is bounded and *v*(*X* $\cap$ *supp*(*p*[t]) < *v*(*X*), where *supp* denotes support and *v* is a Lebesque measure, or n-dimensional volume of the set.

Global search $\Longrightarrow$ *supp*(*p*[*t*]) is such that *v*(*X* $\cap$ *supp*(*p*[t]) = *v*(*X*)

Sufficient conditions for Convergence:

1. D s.t. { $f[x[\mathrm{t}]]$ } is non-increasing. This is satisfied by the MBH algorithm.
   a. $f[\mathrm{D}(x,\xi)] \leq f[(x[t-1]]$
   b. $\xi[t] \in X \Longrightarrow f[\mathrm{D}(x,\xi)] \leq min\{\ [\xi[\mathrm{t}]\ ],\ f[x[t\text{-}1]]\ \}$
2. Zero probability of repeatedly missing any positive volume subset of *X*
   a. Stated formally, $\forall\ A \subseteq X$, where *A* is some subspace (or the entirety) of *X*, s.t. $v(A) > 0,\ \prod_{t=0}^{\infty}(1 - p[t](A)) = 0$
   b. In other words, the sampling strategy based on *p*[*t*] cannot consistently ignore any part of *X* with positive volume

Example given by Baba *et al.*:

$$\mathrm{D} = \begin{cases} \xi[t],\ \xi[t] \in X,\ f[\xi[t]] < f[x[t-1]] \\ x[t-1], otherwise \end{cases}$$

p = $\boldsymbol{\mathcal{N}}(x[t-1], \mathbf{I})$ meaning Gaussian, centered on **x**[*t*-1] with unit variance in each coordinate

Sufficient Condition 1, D s.t. $\{f[x[\mathrm{t}]]\}_{t=0}^{\infty}$ is non-increasing, is satisfied by construction.

Sufficient Condition 2, Zero probability of repeatedly missing any positive volume subset of *X*, is satisfied because *X* is contained in the support of $\boldsymbol{\mathcal{N}}(x[t-1], \mathbf{I})$

This is MBH when the domain of f is the unit hyper-sphere *S* and *p* is Gaussian. The asymptotic convergence (occurring at some unspecified time prior to *t* = infinity) is proven.

In the case of MBH, the speed of convergence was not studied prior to the present work. The central theme and novelty of Chapter IV of the present work, that convergence is sped-up when p is chosen to be similar to $\widehat{q[t]}$, rather than when p is chosen for mathematical convenience, implies that $p = \boldsymbol{\mathcal{N}}(x[t-1], \mathbf{I})$ results in convergence times that are much slower than can be achieved when p is non-Gaussian.

B. Justification for fitting a Gamma distribution to MBH FPTs as their FPTD

Usefulness

Throughout the present work, Gamma distributions fit as density functions of MBH FPTs are used. The benefits of using Gamma distributions as FPTDs rather than histograms as FPT-PMFs include: Because Gamma distributions are defined by two parameters, estimates of those parameters enable mapping a set of FPTs from an MBH operating on the same f and $\boldsymbol{X}^{\mathbb{F}}$ with the same fixed p, or a p adapted in the same way, into a 2-dimensional parametric space of estimated FPTDs. Moreover, different methods of the accelerating that MBH operating on the same f and $\boldsymbol{X}^{\mathbb{F}}$ with the same fixed p, or a p adapted in the same way, can be compared by their relative positions in that 2-dimensional parametric space. In addition, by being continuous probability density functions, Gamma distributions facilitate the calculation of quartiles, which provides a tool for characterizing the estimated time-to-convergence that an MBH operating on a given same f and $\boldsymbol{X}^{\mathbb{F}}$ with the same fixed p, or a p adapted in the same way, will require. Finally, because FPTDs are not unique to specific f and $\boldsymbol{X}^{\mathbb{F}}$, they can be useful in establishing equivalence classes $\mathcal{C}_I = \{ f_i, \boldsymbol{X}^{\mathbb{F}}{}_i \}$ and thereby characterize the estimated time-to-convergence that an MBH operating on members of a $\mathcal{C}_I$ given a fixed chosen or designed p, or a p adapted in the same way, with or without various methods of MBH speed-up.

Empirical justification

The following figures show fits of Gamma distributions to FPTs as approximations of FPTDs, in comparison to FPT-PMFs of the same FPTs. The parametrization of the Gamma distributions were obtained using the estimators provided in Chapter IV. The f used was the prototypical 1-dimensional f shown in Figure 0.a. By inspection, one can see that the fits of the

Gamma distributions are very good. The *p* used was the non-Gaussian *p* having the histogram illustrated in Figure II.4. Figures Appendix B.1 through Appendix B.3 below shows the Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs (histograms shown as blue bars) when SCLS is used and S = {8, 16, and 24}. Likewise, Figures Appendix B.4 through Appendix B.6 below shows Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs histograms (blue bars) when using the same *f* but the Gaussian *p* having the histogram illustrated in Figure IV.8. SCLS is used and again S = {8, 16, and 24}. These figures not only illustrate the quality of fit between the Gamma distribution-approximated FPTD and the FPT-PMFs histograms; they once again show the MBH speed-p that is achieved by using a *p* that is well suited to *f* as defined in Chapter IV.

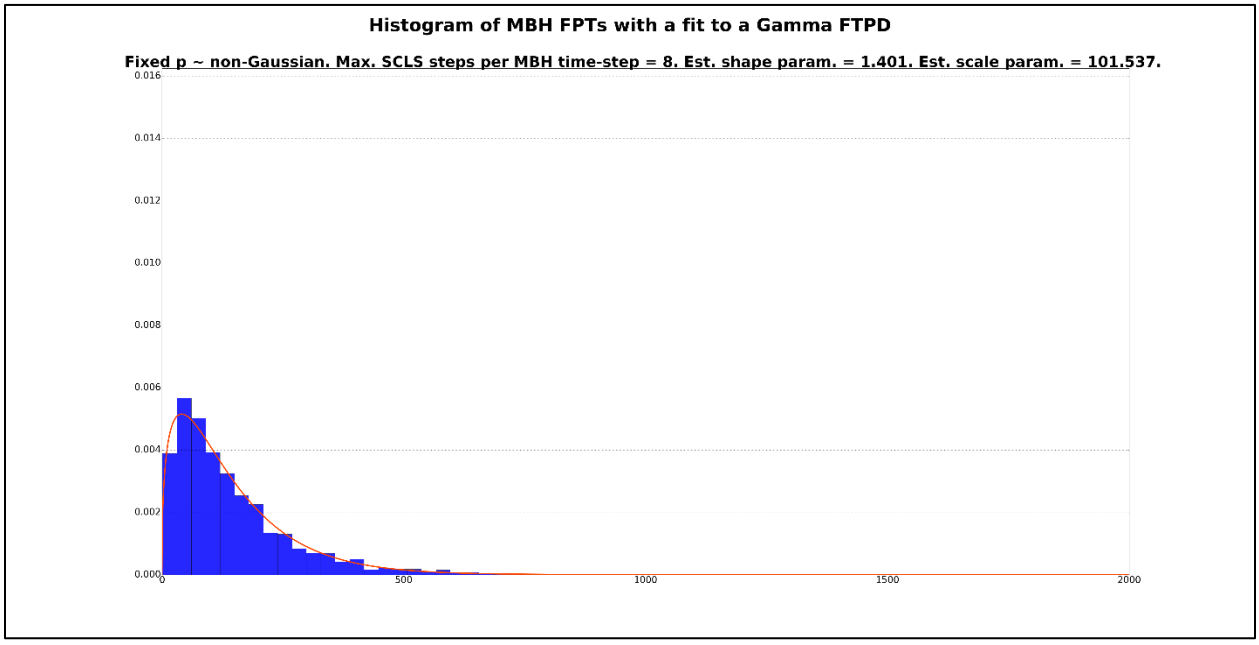


Figure Appendix B.1: The Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs (histograms shown as blue bars) when SCLS is used and S = 8. The *f* used was the prototypical 1-dimensional *f* shown in Figure 0.a. The *p* used was the fixed non-Gaussian *p* having the histogram illustrated in Figure II.4.

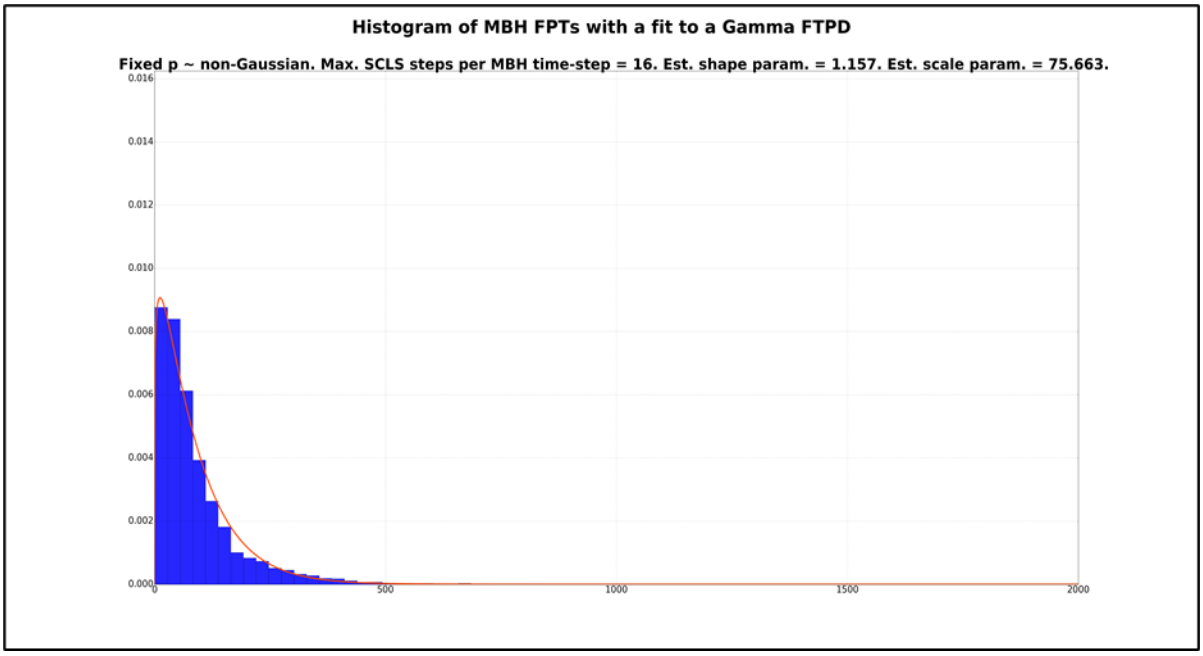


Figure Appendix B.2: The Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs (histograms shown as blue bars) when SCLS is used and S = 16. The *f* used was the prototypical 1-dimensional *f* shown in Figure 0.a. The *p* used was the fixed non-Gaussian *p* having the histogram illustrated in Figure II.4.

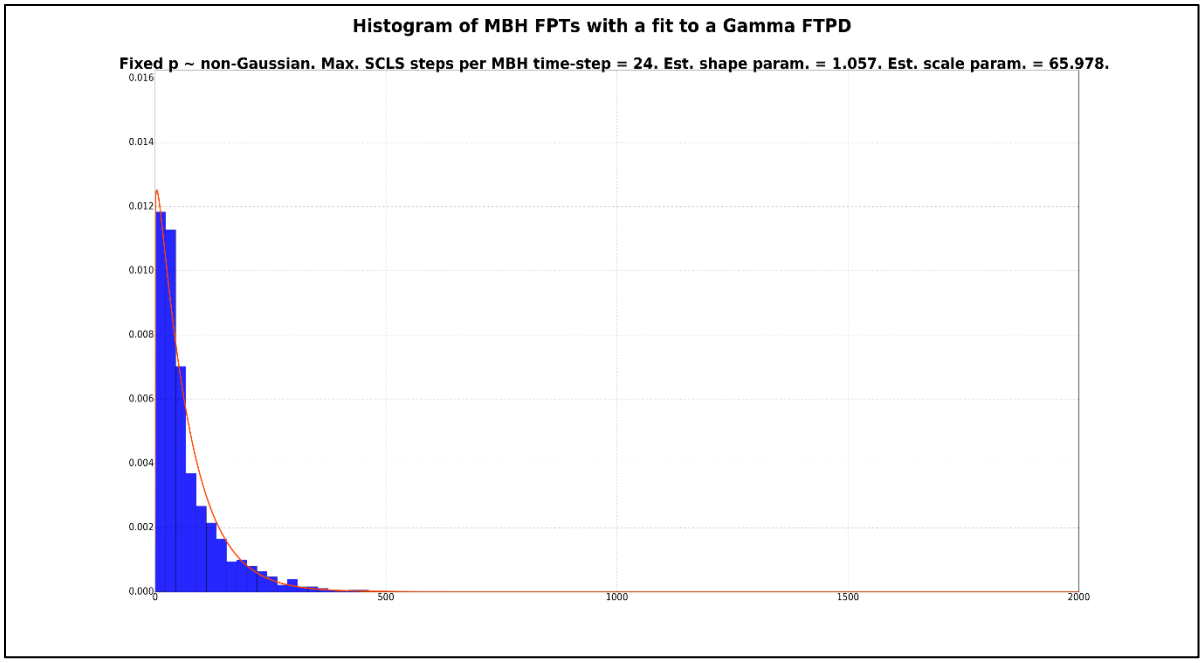


Figure Appendix B.3: The Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs (histograms shown as blue bars) when SCLS is used and S = 24. The *f* used was the prototypical 1-dimensional *f* shown in Figure 0.a. The *p* used was the fixed non-Gaussian *p* having the histogram illustrated in Figure II.4.

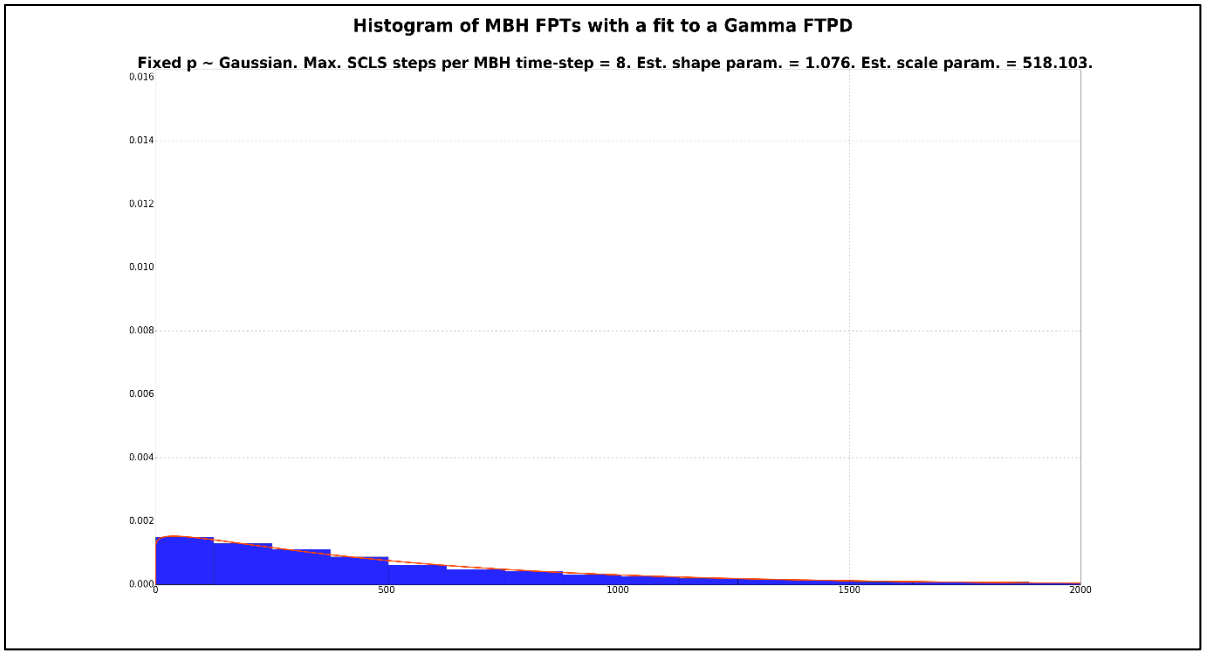


Figure Appendix B.4: The Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs (histograms shown as blue bars) when SCLS is used and S = 8. The *f* used was the prototypical 1-dimensional *f* shown in Figure 0.a. The *p* used was the fixed Gaussian *p* having the histogram illustrated in Figure IV.6

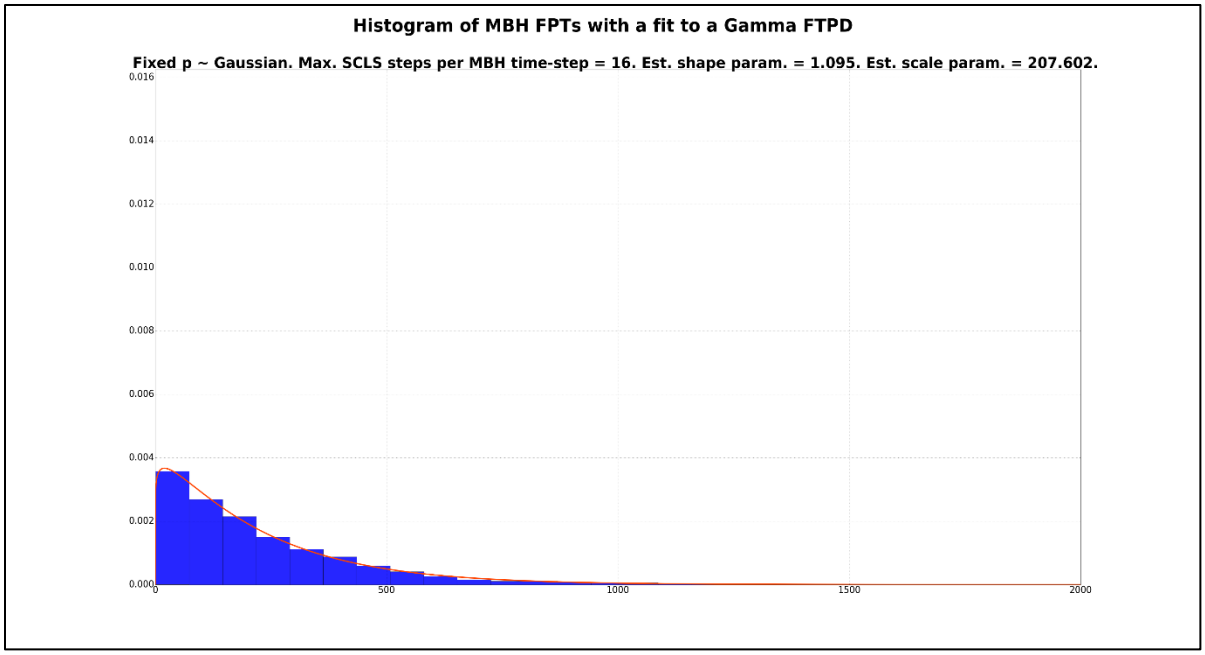


Figure Appendix B.5: The Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs (histograms shown as blue bars) when SCLS is used and S = 16. The *f* used was the prototypical 1-dimensional *f* shown in Figure 0.a. The *p* used was the fixed Gaussian *p* having the histogram illustrated in Figure IV.6

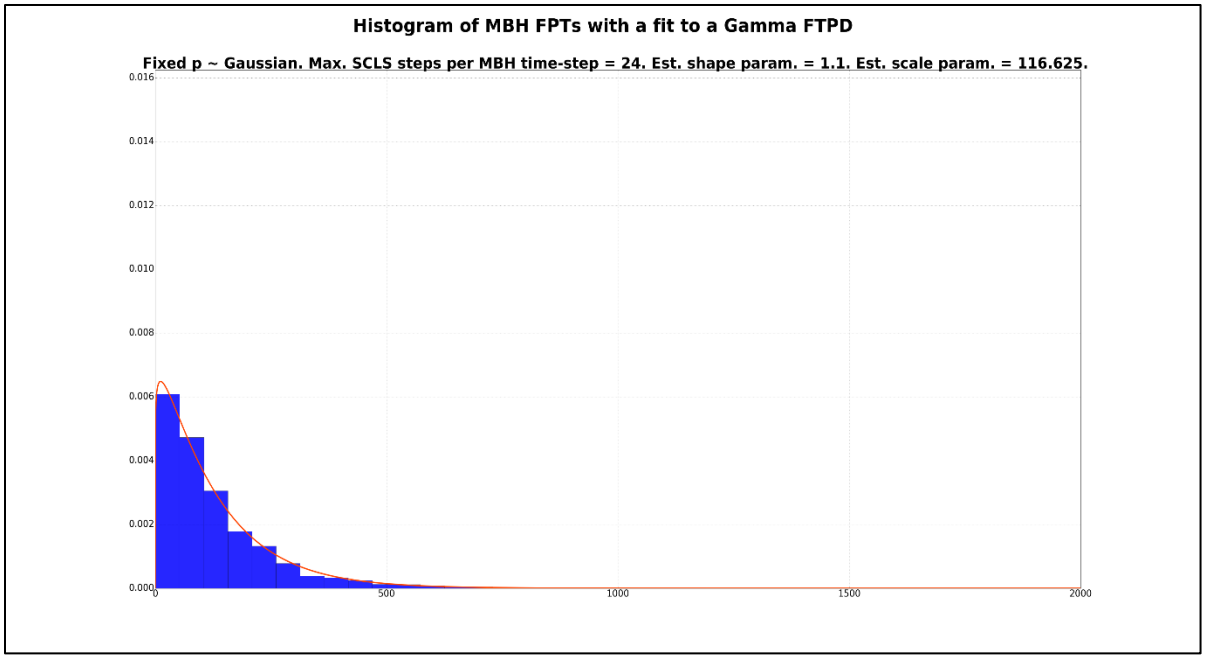


Figure Appendix B.6: The Gamma distribution-approximated FPTD (as a red line) drawn on top of the FPT-PMFs (histograms shown as blue bars) when SCLS is used and S = 8. The *f* used was the prototypical 1-dimensional *f* shown in Figure 0.a. The *p* used was the fixed Gaussian *p* having the histogram illustrated in Figure IV.6

The formal justification for the apparent goodness of fit is raised as an open question in Chapter VII. There, the present author's hypothesis is offered.

Demonstration of Use

The figures below illustrate how methods of the accelerating MBH operating on the same *f* and $\boldsymbol{X}^{\mathbb{F}}$ with the same or different fixed *p* can be compared by their relative positions in a 2-dimensional parametric space comprised of the shape parameter and the scale parameter that defines Gamma distributions fit to MBH FPTs. Because the figures involve the same *f* and the same speed-up method (SCLS) but different fixed *p*, they also illustrate the impact on convergence of using different *p* for the MBH operating on the same *f* and $\boldsymbol{X}^{\mathbb{F}}$.

The $\Gamma(\alpha, \theta)$ plane is useful for characterizing the impact of different *p* on the same *f*

Figure Appendix B.7 below shows the impact of different *p* on the same *f* shown as a scatter-plot on the $\Gamma(\alpha, \theta)$ plane. The green circular markers show the impact on the parameters of the Gamma-fit of using Gaussian *p* illustrated in Figure IV.**6**, and SCLS with 1, 8, 16, and 32 maximum allowable local steps per MBH time-step for one hopper operating upon on prototypical *f* illustrated in Figure 0.a. The red circular markers show the impact on the parameters of the Gamma-fit of using non-Gaussian *p* illustrated in II.4, and SCLS with 1, 8, 16, and 32 maximum allowable local steps per MBH time-step for one hopper operating upon prototypical *f* illustrated in Figure 0.a. The impact on MBH convergence speed of using of fixed non-Gaussian *p* and SCLS with 8, 16, 24 and 32 local steps per MBH time-step, versus fixed Gaussian *p* and SCLS with 8, 16, 24 and 32 local steps per MBH time-step, is illustrated by the location of the green versus red points in the $\Gamma(\alpha, \theta)$ plane.

In Figure Appendix B.7, note how nearly separable the data points for MBH operating on prototypical *f* using non-Gaussian *p* (red) are from MBH operating on prototypical *f* using

Gaussian p (blue). This illustrates that the same f responds differently to different p being used in combination with SCLS.

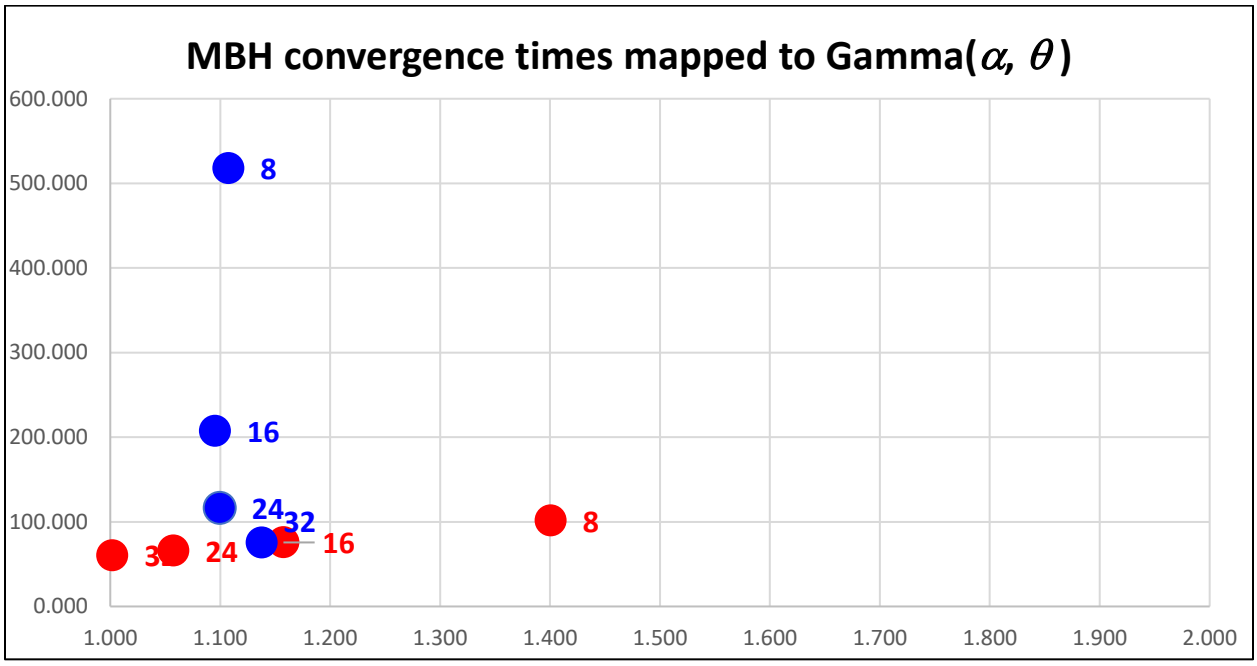


Figure Appendix B.7: Impact of different p on the same f when using SCLS is shown as a scatter-plot on the $\Gamma(\alpha, \theta)$ plane

x-axis: α

y-axis: θ

Blue circular markers: The position in the $\Gamma(\alpha, \theta)$ plane of S SCLS sub-steps for S = 8, 16, 24, and 32 maximum allowable local steps per MBH time-step when f is prototypical f and p is fixed Gaussian

Red circular markers: The position in the $\Gamma(\alpha, \theta)$ plane of S SCLS sub-steps for S = 8, 16, 24, and 32 maximum allowable local steps per MBH time-step when f is prototypical f and p is fixed non-Gaussian

The $\Gamma(\alpha, \theta)$ plane is also useful for characterizing the impact of the same SCLS methods on different f

Likewise, the $\Gamma(\alpha, \theta)$ plane can be used to character the impact of the same speed-up method (here SCLS) on different f (here, 1-dimensional prototypical f per Figure 0.a, versus 1-dimensional Gibsonian f per Figure III.2)**.** Figure Appendix B.8 below shows in purple diamond markers, the impact on the parameters of the Gamma-fit by SCLS for 8, 16, 32, 40, 48, 56, and 64

maximum allowable local steps per MBH time-step for one hopper when *f* is Gibsonian *f* as illustrated in Figure III.3. The blue circular markers show the impact on the parameters of the Gamma-fit by SCLS for 8, 16, 32, 40, 48, 56, and 64 maximum allowable local steps per MBH time-step for one hopper when *f* is prototypical *f* as illustrated in Figure 0.a. Note that the separation between purple diamond markers and the green circular markers suggests that the $\Gamma(\alpha, \theta)$ plane could be used as a space in which to position and classify equivalence classes for some *f* if the members of those classes impact convergence rates as differently as the 1-dimensional example of prototypical *f* compared to the 1-dimensional example of Gibsonian *f*.

In Figure Appendix B.8 note how separable the data points for prototypical *f* (green) vs. Gibsonian *f* (purple) appear to be, again indicating that different *f* respond differently to the speed-up methods provided in Chapters IV and V. Chapter VII provides a hypothesis that the ways in which different *f* respond differently to different speed-up methods provided in the present work might be used to assign an unknown *f* to a convergence time-impacting equivalence class within an on-going MBH.

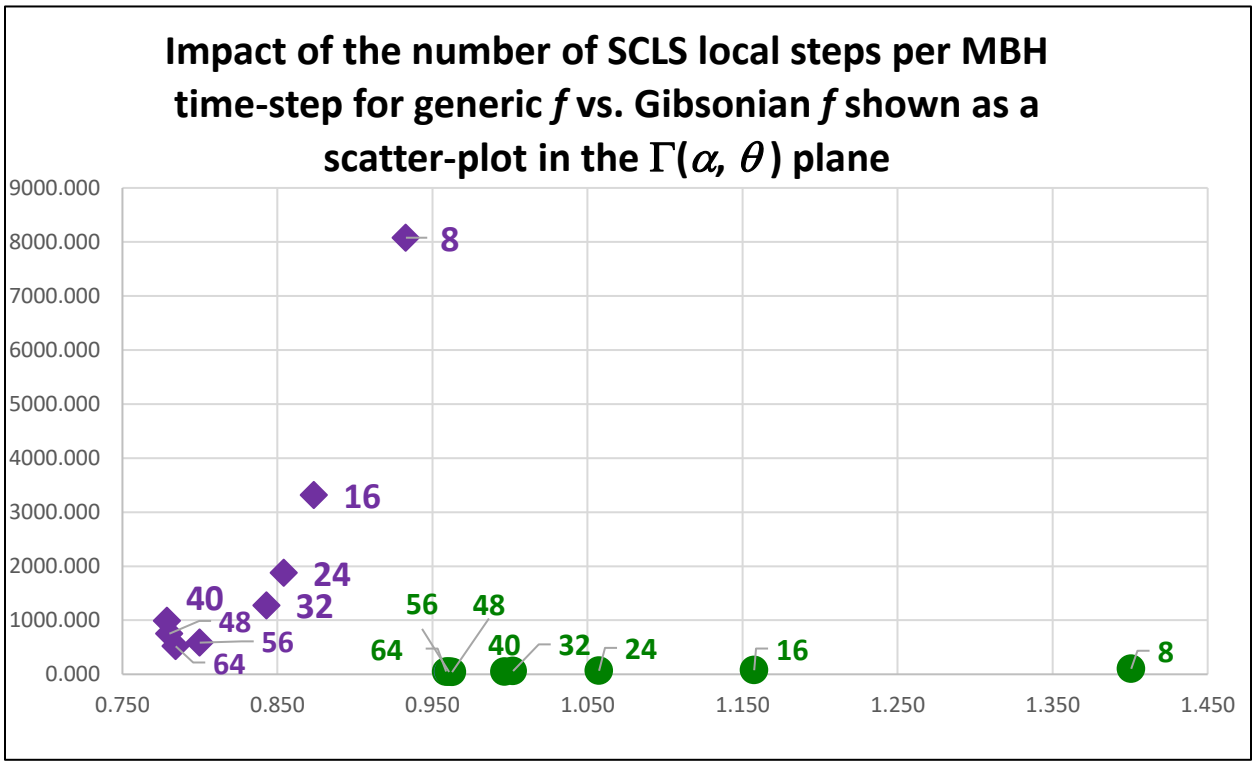


Figure Appendix B.8: Impact of number of SCLS local steps per MBH time-step on prototypical *f* vs. Gibsonian *f* as a scatter-plot on the $\Gamma(\alpha, \theta)$ plane

x-axis: α

y-axis: θ

Purple diamond markers: The position in the $\Gamma(\alpha, \theta)$ plane of S SCLS sub-steps for S = 8, 16, 32, 40, 48, 56, and 64 maximum allowable local steps per MBH time-step when *f* is Gibsonian *f*

Blue circular markers: The position in the $\Gamma(\alpha, \theta)$ plane of S SCLS sub-steps for S = 8, 16, 32, 40, 48, 56, and 64 maximum allowable local steps per MBH time-step when *f* is prototypical *f*

C. Probability of advancement on a 1-dimensional Gibsonian *f*

This appendix addresses the question: For a given f and p, at depth $f[\boldsymbol{x}[t]] = d$, what is the probability of a hopper at any point $\boldsymbol{x} = f^{-1}[d]$ descending more deeply into f, i.e., of drawing a $\Delta\boldsymbol{x}$ such that $f[\boldsymbol{x}[t]+\Delta\boldsymbol{x}] < f[\boldsymbol{x}[t]]$? As an example, choose f to be 1-dimensional Gibsonian shown in Figures III.2 and III.3 so that there are exactly four points $\boldsymbol{x} = f^{-1}[d]$ while $d > \min_{\text{local}}(f)$, each of which are necessarily on a different side of one of the two basins, respectively, and there are exactly two points $\boldsymbol{x} = f^{-1}[d]$ while $\min_{\text{global}}(f) < d < \min_{\text{local}}(f)$, each of which are necessarily on the left and right side of the narrow basin .

The answer explains the impact on the speed of MBH convergence of biasing the shape of p by showing how p affects the probability of the MBH advancing toward $\min_{\text{global}}(f)$ at each next MBH time-step t+1 given $\boldsymbol{x}[t] = f^{-1}[d]$. This provides an additional way to explain the material in Chapter IV.

The answer also explains the impact on the speed of MBH convergence of biasing the location from which each next hop is taken, and how $f^{-1}[d]$ at that biased location affects the probability of the MBH advancing toward $\min_{\text{global}}(f)$ at each next MBH time-step t+1 given $\boldsymbol{x}[\text{t}] = f^{-1}[d]$. This provides an additional way to explain the material in Chapter V, including why speeding up the descent of the hopper into the "wrong" basin, meaning the basin that contains $\min_{\text{local}}(f)$ but not $\min_{\text{global}}(f)$, speeds-up the convergence to $\min_{\text{global}}(f)$.

Further, the answer can be used to investigate several issues that are not included in the present work, but which have been encountered many times during the research that culminates in the present work. These issues include: The impact on MBH convergence speed of the distance

across X between $\text{argmin}_{\text{local}}(f)$ and $\text{argmin}_{\text{global}}(f)$; the distance across (the "width") of the wide basin that contains $\text{min}_{\text{local}}(f)$ but not $\text{min}_{\text{global}}(f)$, compared to distance across (the "width") of the narrow basin that contains $\text{min}_{\text{global}}(f)$ but not $\text{min}_{\text{local}}(f)$; the depth $\text{min}_{\text{local}}(f)$ but not $\text{min}_{\text{global}}(f)$, compared to the depth; and why the convergence of an MBH operating on a Gibsonian f is slower than the convergence of an MBH operating on a "nested' f – where nest f has its $\text{min}_{\text{global}}(f)$ in a basin that is very close to, or within, the basin that contains $\text{min}_{\text{local}}(f)$.

Figure Appendix C.1 illustrates f and the $\boldsymbol{x}[t] = f^{-1}[d]$. Figure Appendix C.2 provides the probabilities of descent at each depth d assuming that p is scaled to $len(\boldsymbol{X})$. After $d \leq \text{min}_{\text{non-global}}(f)$, the probability for an entry into the wide basin from the narrow basin goes to zero, although the probability of the MBH being force to wait-in-place at $\text{min}_{\text{non-global}}(f)$ remains non-zero until a $\boldsymbol{\Delta x}$ is drawn such that the hopper hops from $\text{min}_{\text{non-global}}(f)$ in the wide basin to somewhere lower in the narrow basin.

Figures Appendix C.3 through Appendix C.10 illustrate probabilities of descent down each basin, at each depth d, for each $\boldsymbol{x}[t] = f^{-1}[d]$, respectively, assuming that p is scaled to $len(\boldsymbol{X})$.

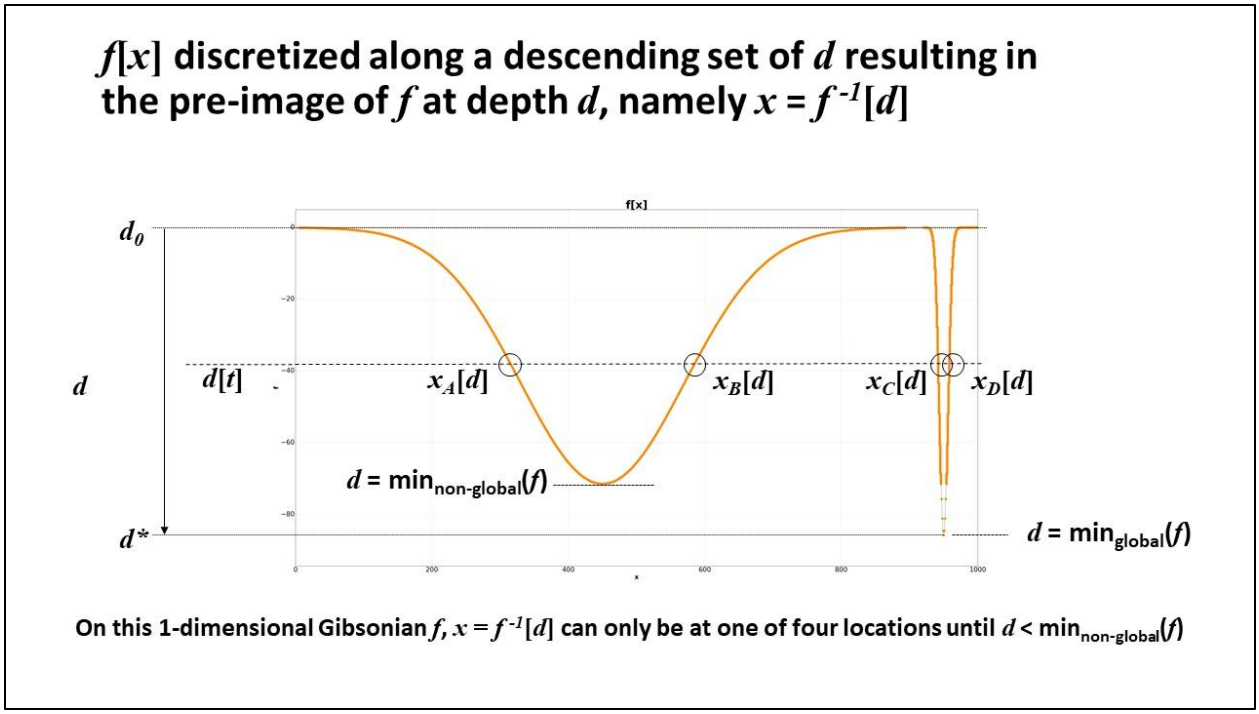


Figure Appendix C.1: f and the $\mathbf{x}[t] = f^{-1}[d]$

- Prob($x_A[d]$ descends the wide basin) = $\frac{1}{2}\int_{x_A[d-1]}^{x_B[d-1]} p(u)\,du$
- Prob($x_B[d]$ descends the wide basin) = $\frac{1}{2}\int_{x_A[d-1]}^{x_B[d-1]} p(u)\,du$
- Prob($x_C[d]$ descends the narrow basin) = $\frac{1}{2}\int_{x_C[d-1]}^{x_D[d-1]} p(u)\,du$
- Prob($x_D[d]$ descends the narrow basin) = $\frac{1}{2}\int_{x_C[d-1]}^{x_D[d-1]} p(u)\,du$

- Prob($x_A[d]$ descends the narrow basin) = $\frac{1}{2}\int_{x_A[d-\varepsilon]}^{x_D[d-1]} p(u)\,du - \frac{1}{2}\int_{x_A[d-1]}^{x_C[d-1]} p(u)\,du$
- Prob($x_B[d]$ descends the narrow basin) = $\frac{1}{2}\int_{x_B[d-1]}^{x_D[d-1]} p(u)\,du - \frac{1}{2}\int_{x_B[d-1]}^{x_C[d-1]} p(u)\,du$
- Prob($x_C[d]$ descends the wide basin) = $\frac{1}{2}\int_{x_A[d-1]}^{x_B[d-1]} p(u)\,du - \frac{1}{2}\int_{x_B[d-1]}^{x_C[d-1]} p(u)\,du$
- Prob($x_D[d]$ descends the wide basin) = $\frac{1}{2}\int_{x_A[d-1]}^{x_B[d-1]} p(u)\,du - \frac{1}{2}\int_{x_B[d-1]}^{x_D[d-1]} p(u)\,du$

Figure Appendix C.2: Transition probabilities while $d > \min_{\text{non-global}}(f)$ assuming that p is scaled to $len(\boldsymbol{X})$. After $d \leq \min_{\text{non-global}}(f)$, the probability for an entry into the wide basin from the narrow basin goes to zero although the probability of the MBH being force to wait-in-place at $\min_{\text{non-global}}(f)$ remains non-zero until a $\boldsymbol{\Delta x}$ is drawn such that the hopper hops from $\min_{\text{non-global}}(f)$ in the wide basin to somewhere lower in the narrow basin.

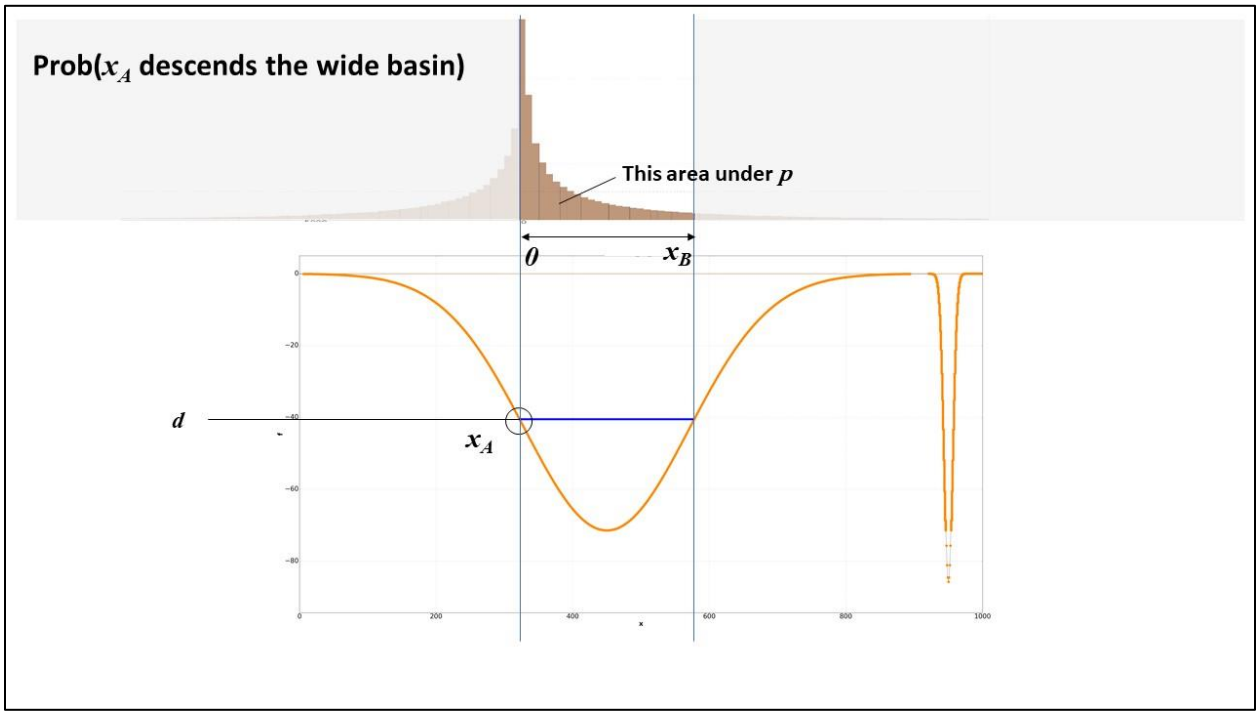


Figure Appendix C.3: Prob(x_A descends the wide basin)

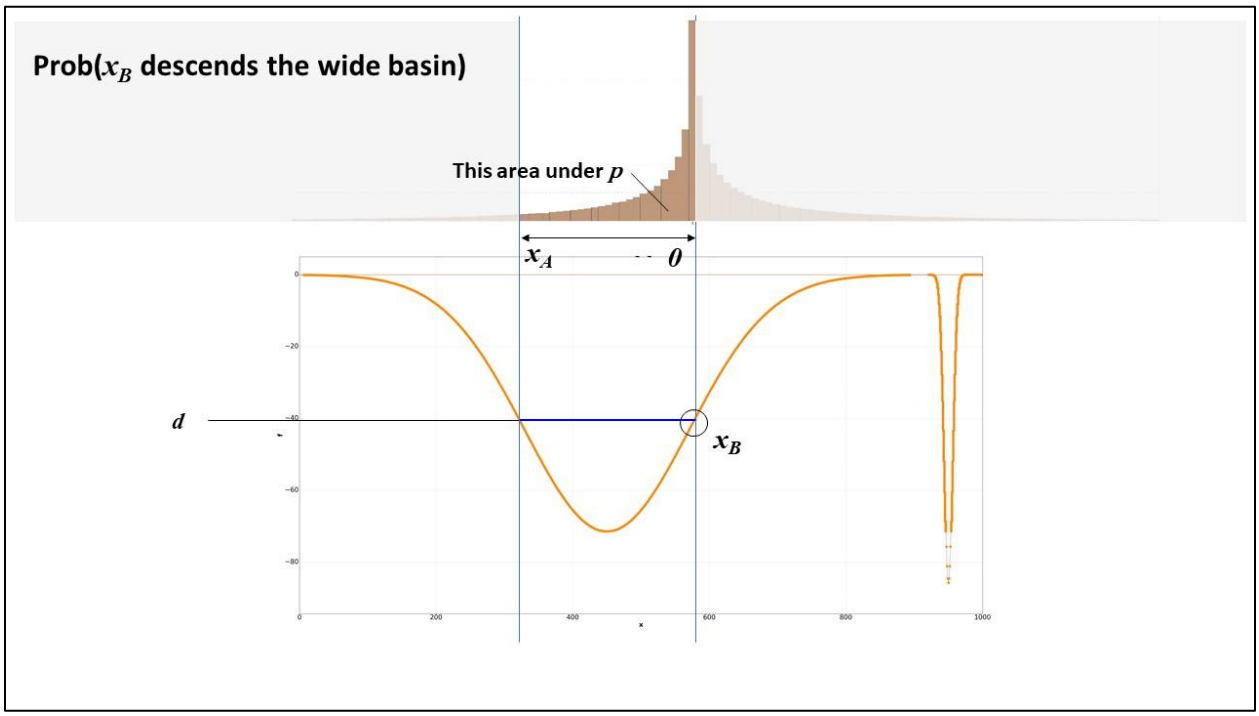


Figure Appendix C.4: Prob(x_B descends the wide basin)

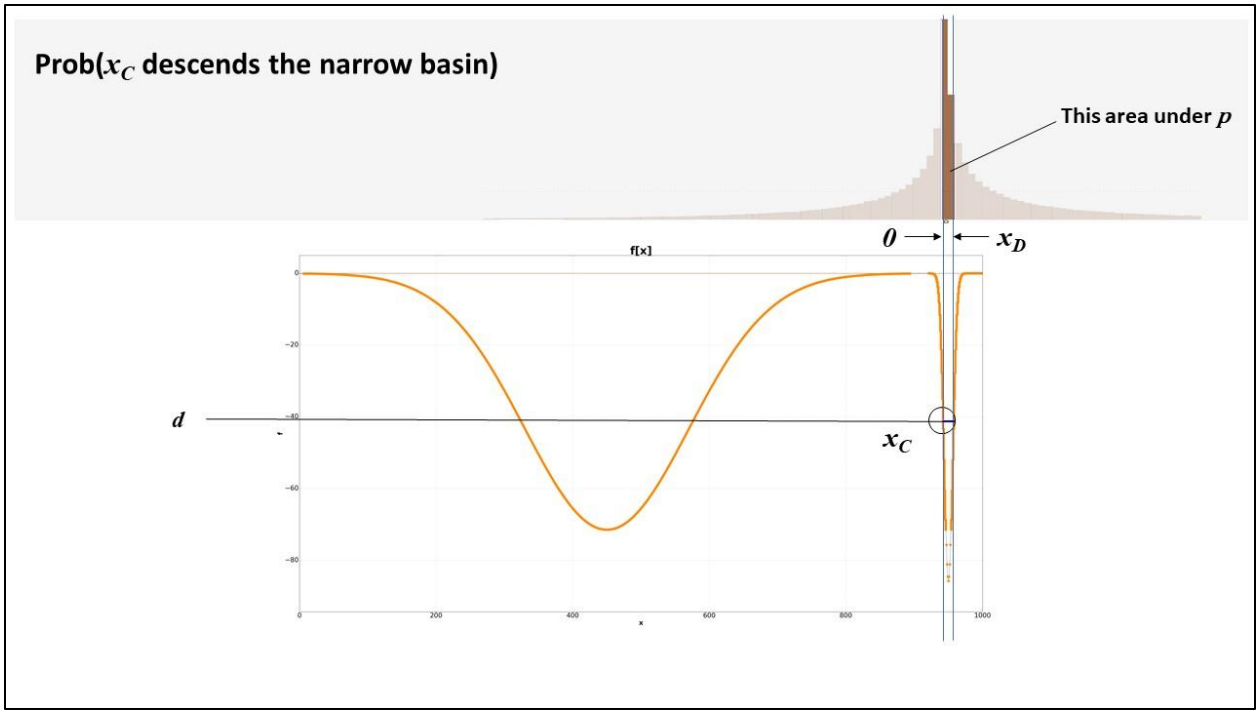


Figure Appendix C.5: Prob(x_C descends the narrow basin)

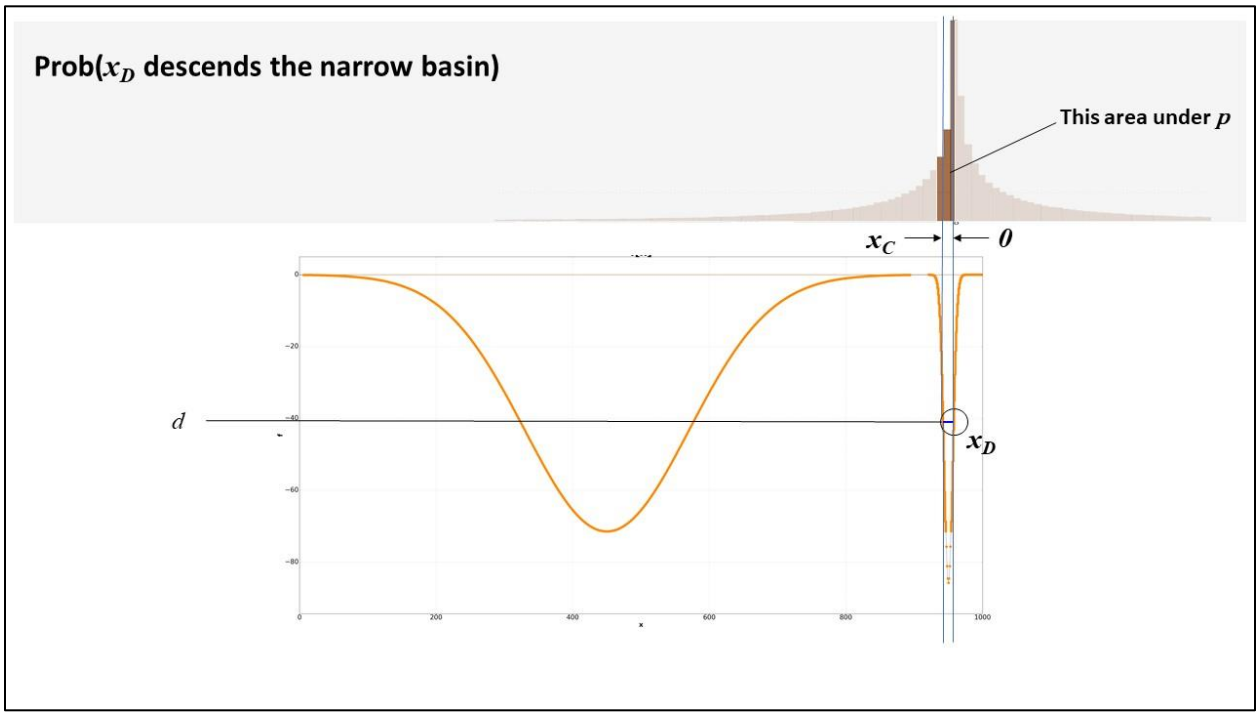


Figure Appendix C.6: Prob(x_D descends the narrow basin)

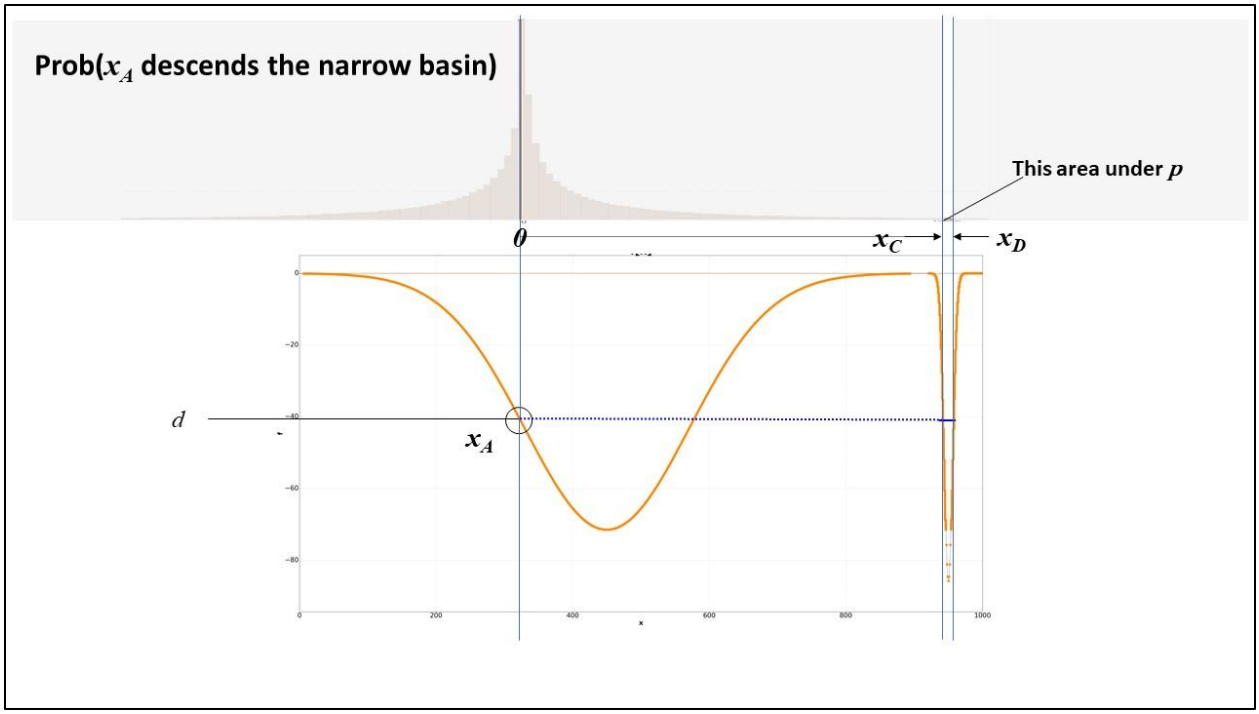


Figure Appendix C.7: Prob(x_A descends the narrow basin)

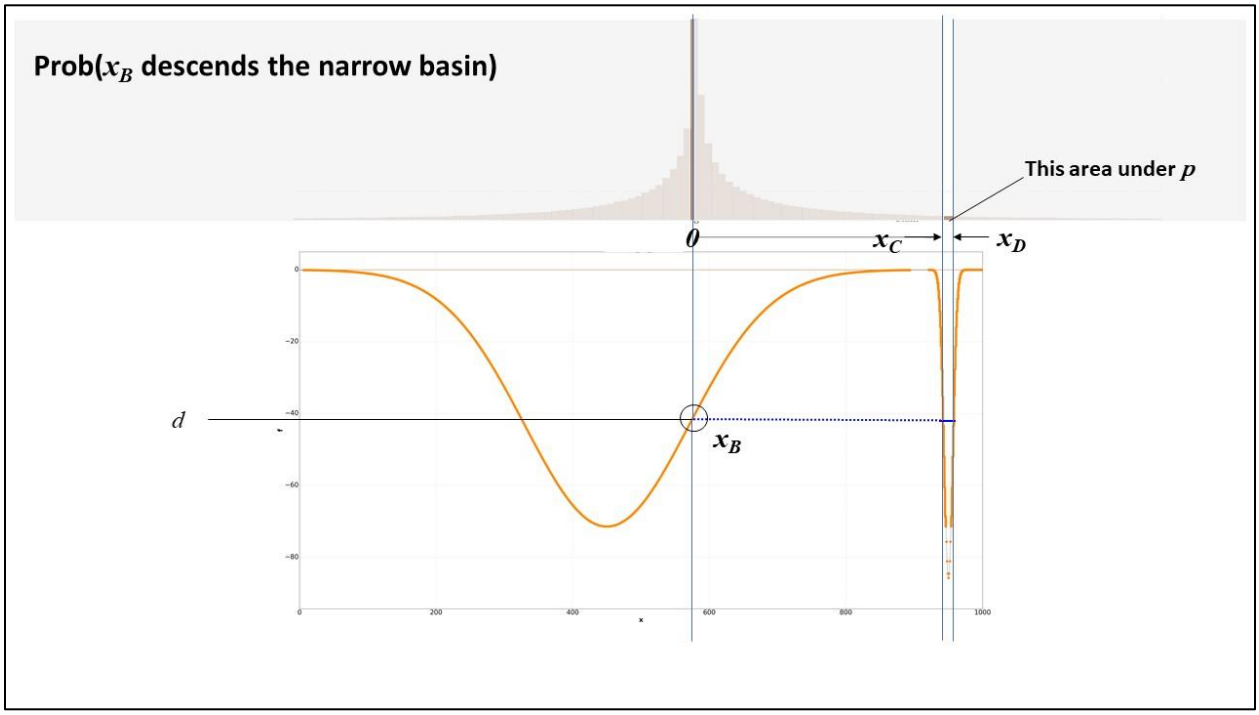


Figure Appendix C.8: Prob(x_B descends the narrow basin)

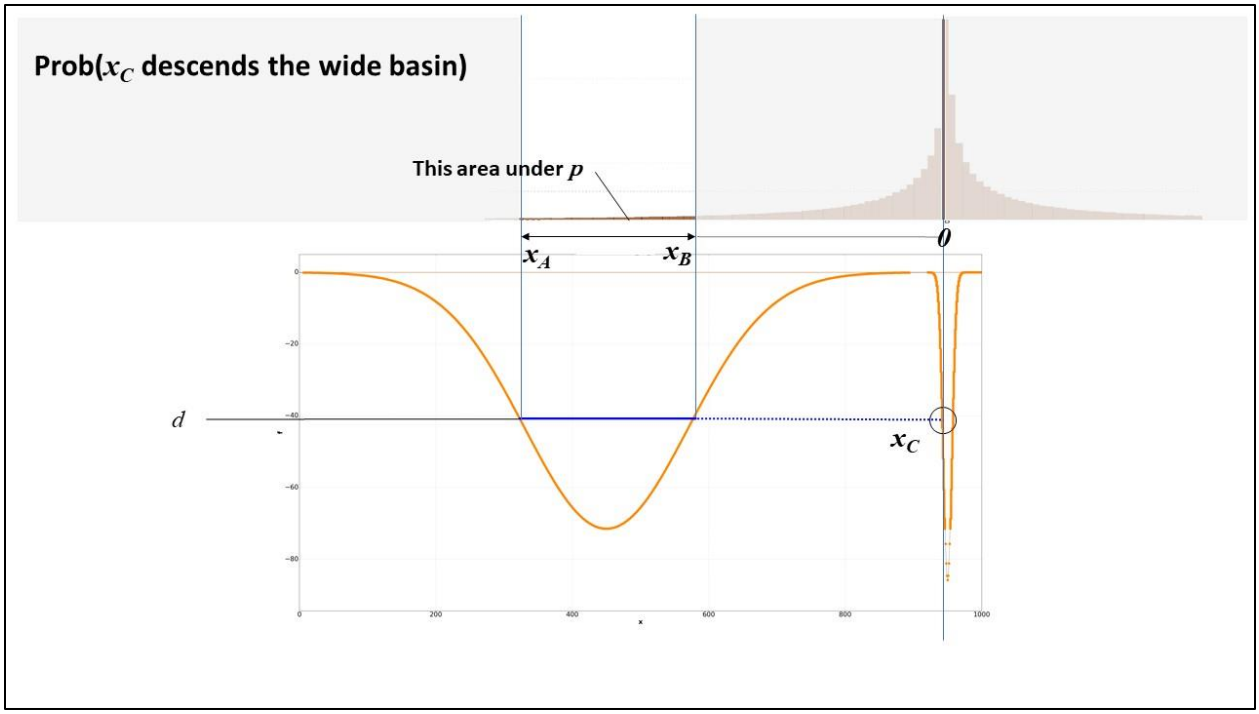


Figure Appendix C.9: Prob(x_C descends the wide basin)

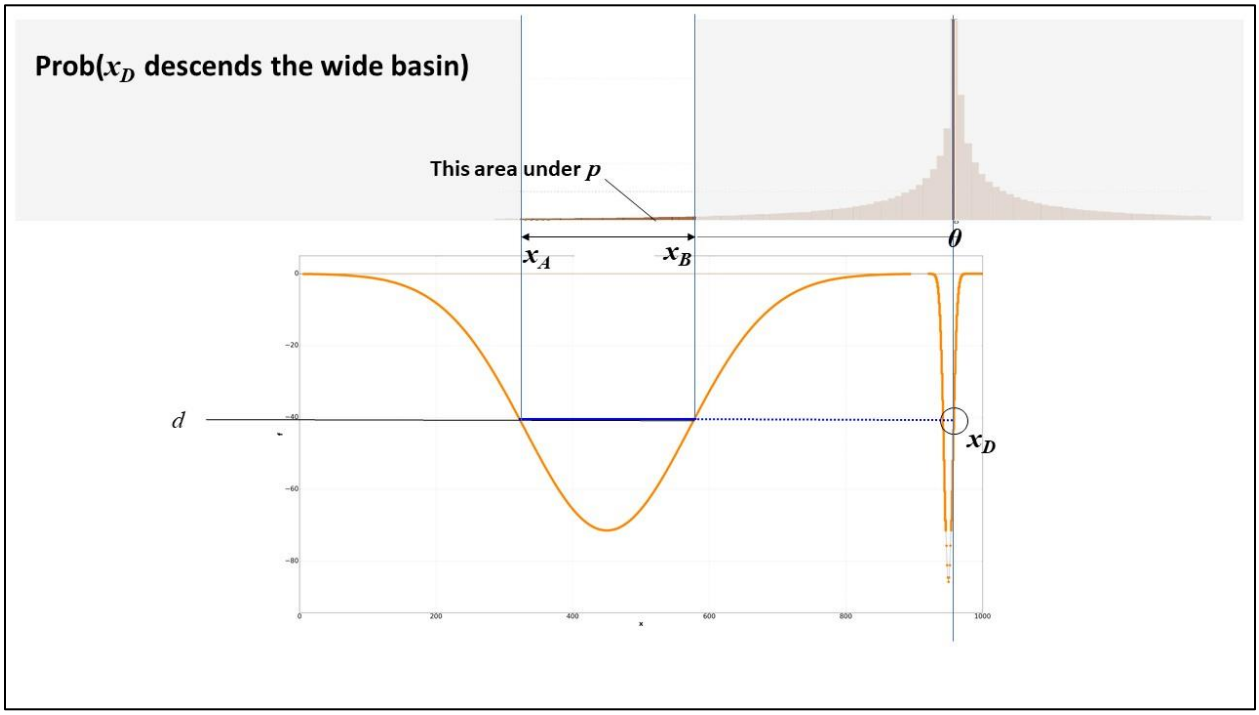


Figure Appendix C.10: Prob(x_D descends the narrow basin)

Figures Appendix C.11 through Appendix C.14 illustrate probabilities of descent down each basin, at each depth *d,* for each $\boldsymbol{x}[t] = f^{-1}[d]$, respectively, assuming that p is scaled to $len(\boldsymbol{X})$. In Figure Appendix C.14 it can be seen that, using $p = len(\boldsymbol{X})\cdot$Laplace(0, ½) on this f provides a greater probability of hopping into the narrow basin, and remaining in the narrow basin, compared to using $p = len(\boldsymbol{X})\cdot$Laplace(0, 1) or $p = len(\boldsymbol{X})\cdot$Gaussian(0, ½).

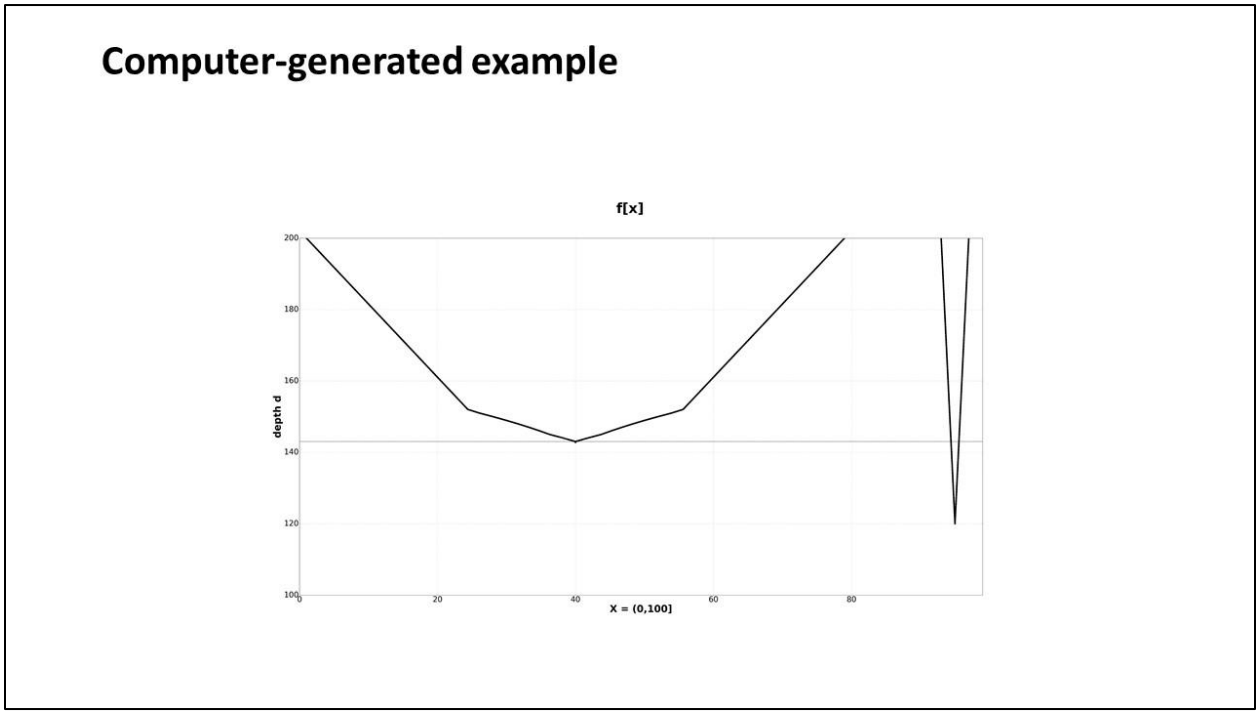


Figure Appendix C.11: f used in the computer-generated example

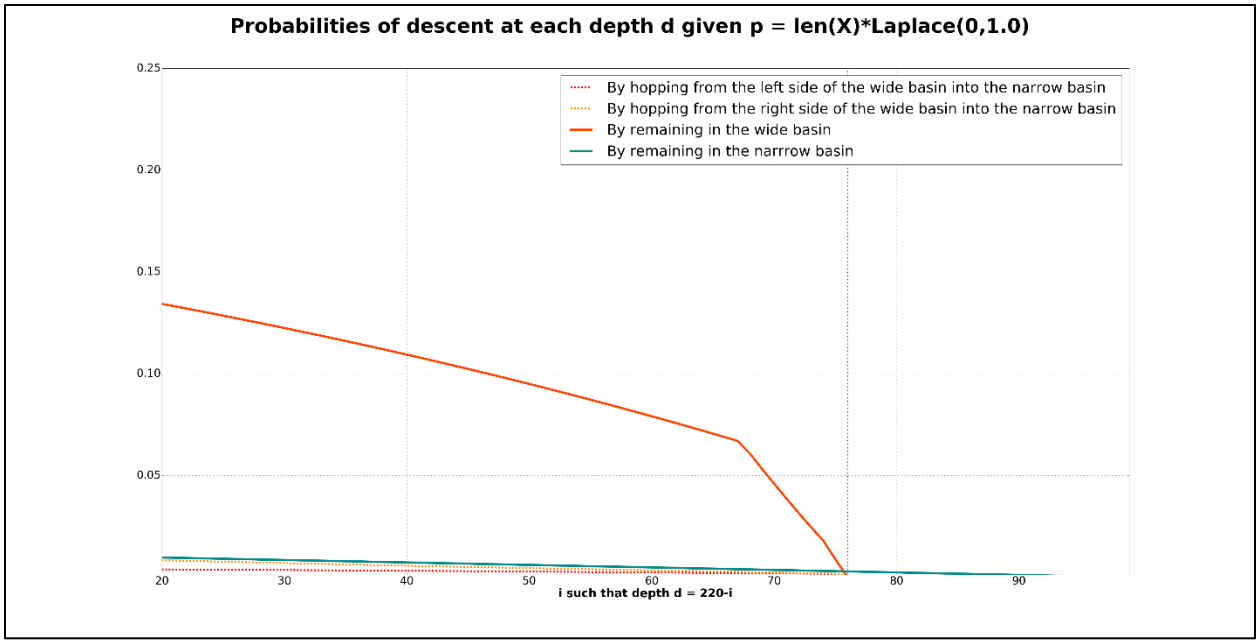


Figure Appendix C.12: In the computer-generated example, the probabilities at each depth *d* given *f* and *p* = len(***X***)·Laplace(0, 1)

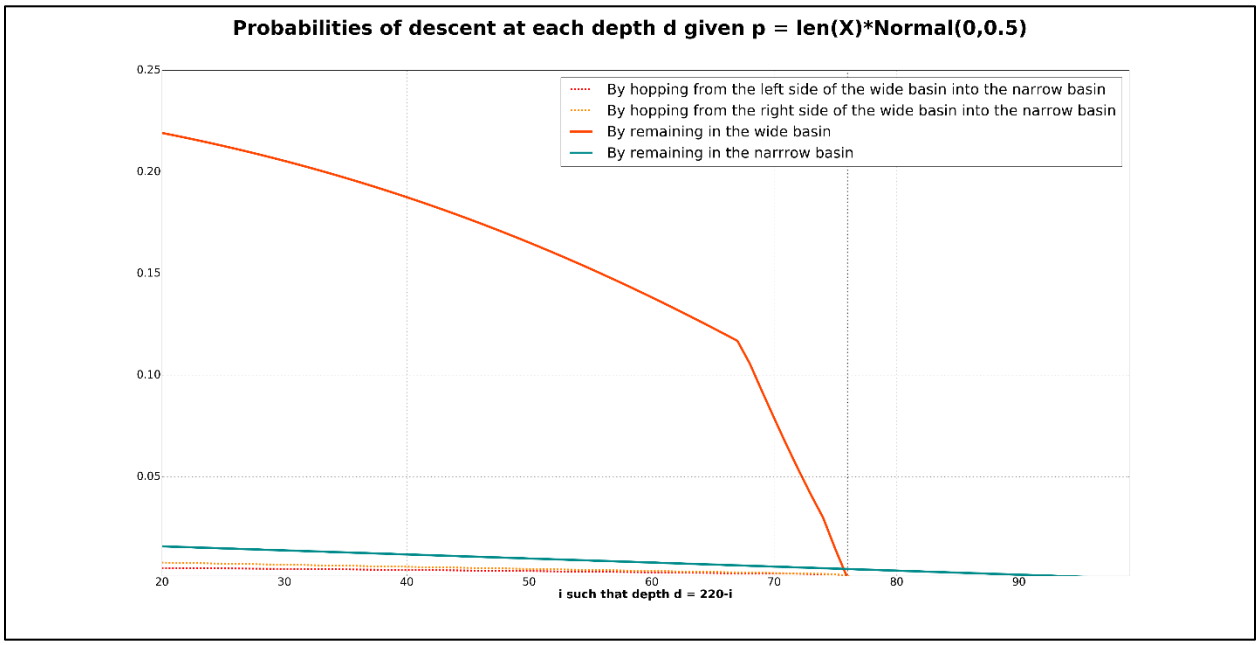


Figure Appendix C.13: In the computer-generated example, the probabilities at each depth *d* given *f* and *p* = len (***X***)·Gaussian(0, ½)

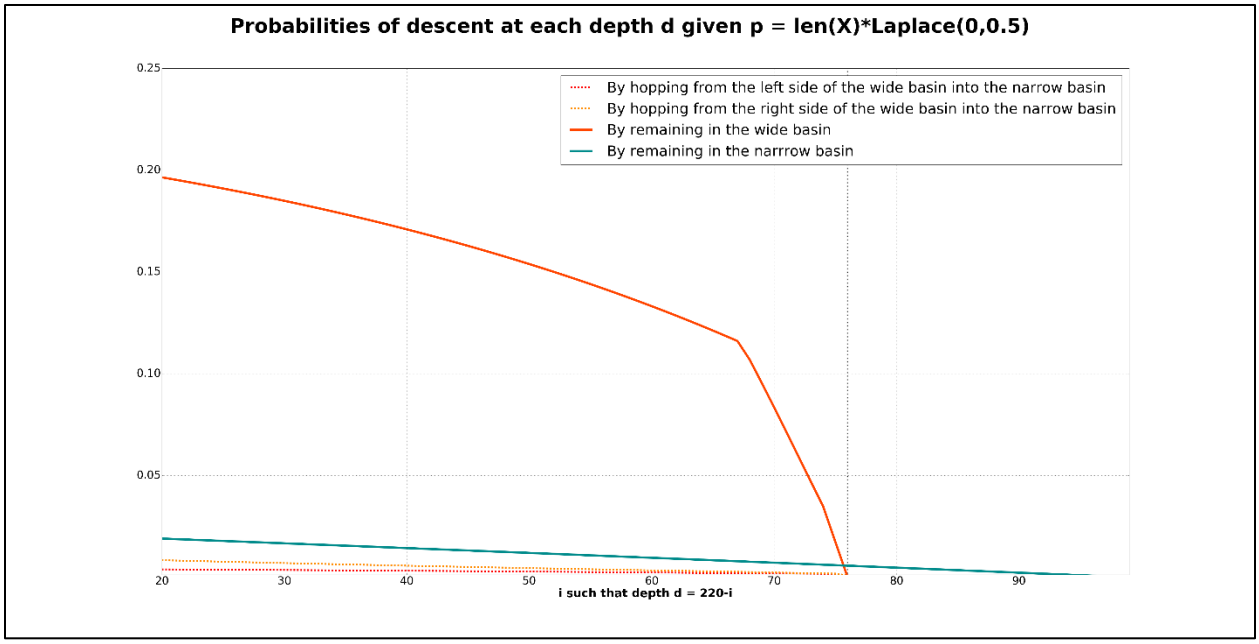
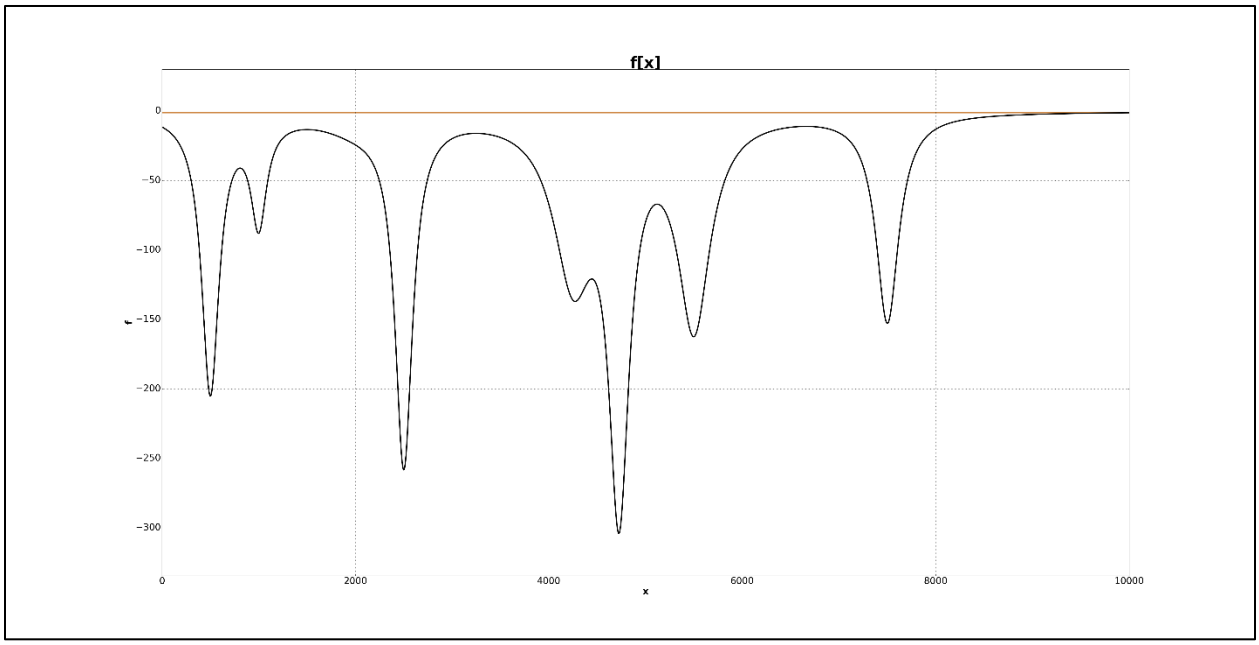


Figure Appendix C.14: In the computer-generated example, the probabilities at each depth *d* given *f* and *p* = len (***X***)·Laplace(0, ½)

D. Python code used to generate simulated *f*

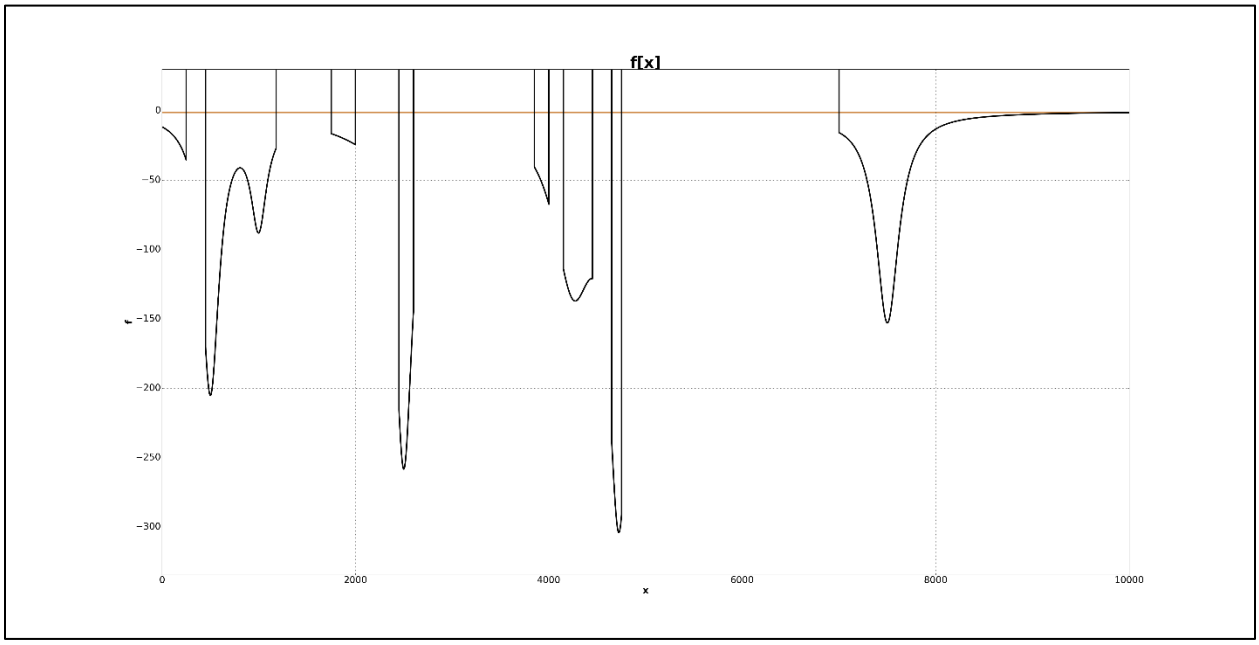
To generate:




```
import numpy as np
extent=10000
f=np.zeros(extent)
x_c1=int(0.20*extent)
x_c6=int((float(500)/float(extent))*extent)
kf6=4000.0
b6=12000.0
x_c7=int((float(1000)/float(extent))*extent)
kf7=1200.0
b7=9600.0
x_c8=int((float(2500)/float(extent))*extent)
kf8=5000.0
b8=12000.0
x_c8a=int((float(4250)/float(extent))*extent)
kf8a=11000.0
b8a=58500.0
x_c9=int((float(4725)/float(extent))*extent)
kf9=7200.0
b9=16000.0
x_c9a=int((float(5500)/float(extent))*extent)
kf9a=10000.0
```

```
b9a=40000.0
x_c10=int((float(7500)/float(extent))*extent)
kf10=5000.0
b10=20000.0

for x_index in range(0,extent):
        f[x_index]=-2480.0/(((x_index-x_c1)*(x_index-x_c1))+170000.0)
        f[x_index]=f[x_index]-kf6/(((x_index-x_c6)*(x_index-x_c6))+b6)
        f[x_index]=f[x_index]-kf7/(((x_index-x_c7)*(x_index-x_c7))+b7)
        f[x_index]=f[x_index]-kf8/(((x_index-x_c8)*(x_index-x_c8))+b8)
        f[x_index]=f[x_index]-kf8a/(((x_index-x_c8a)*(x_index-x_c8a))+b8a)
        f[x_index]=f[x_index]-kf9/(((x_index-x_c9)*(x_index-x_c9))+b9)
        f[x_index]=f[x_index]-kf9a/(((x_index-x_c9a)*(x_index-x_c9a))+b9a)
        f[x_index]=f[x_index]-kf10/(((x_index-x_c10)*(x_index-x_c10))+b10)

        f[x_index]=f[x_index]*0.1*0.299996
```

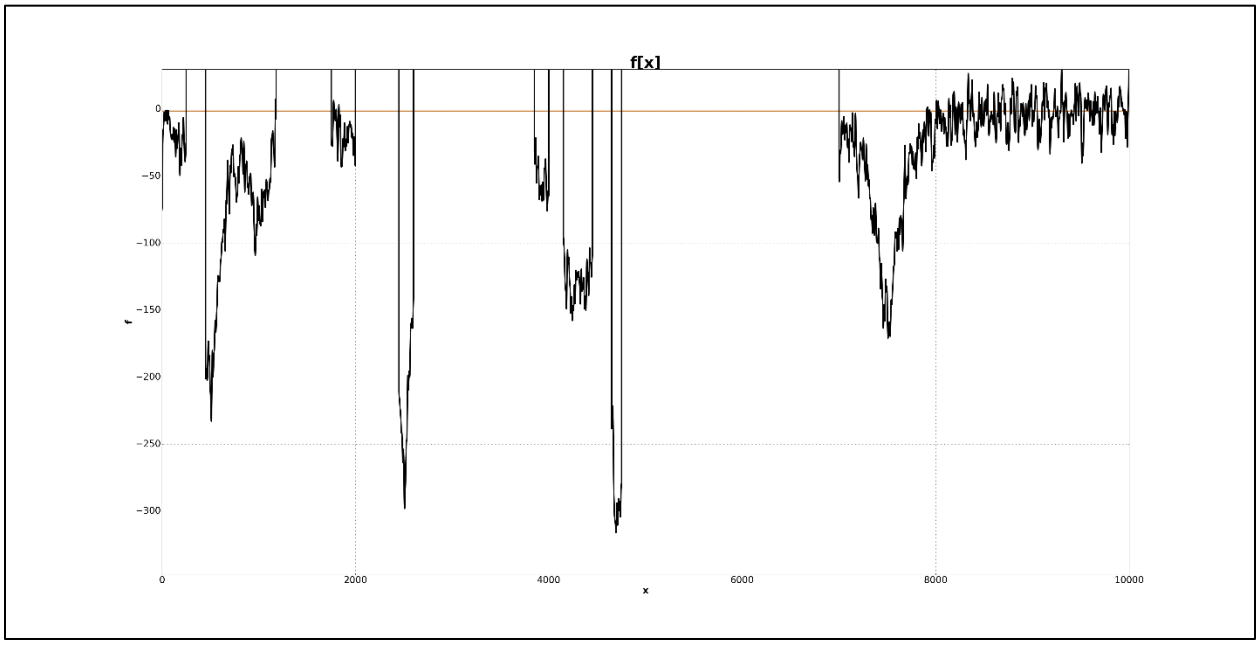
To generate:



After the code provided above, add:

```
disconnected_X=True
if disconnected_X==True:
    for x_index in range(0,extent):
        if ((x_index>250 and x_index<450) or (x_index>1180 and x_index<1750) or (x_index>2000 and x_index<2450) or
        (x_index>2600 and x_index<3850) or (x_index>4000 and x_index<4150) or(x_index>4450 and x_index<4650) or
        (x_index>4752 and x_index<7000))
        f[x_index]=10.0
```

To generate:



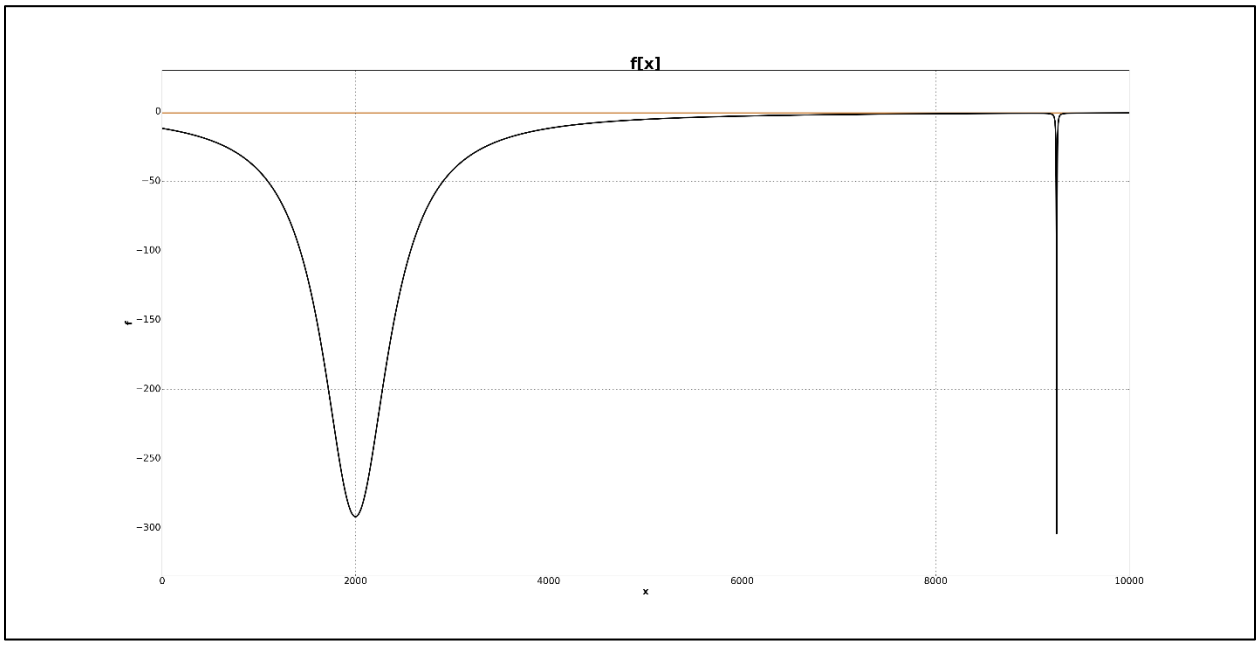
After the code provided above, add:

```
for x_index in range(0,extent):
    f[x_index]=f[x_index]-(0.000350*( np.cos(128.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000325*( np.cos(183.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000300*( np.cos(231.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000275*( np.cos(311.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000250*( np.cos(397.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000225*( np.cos(431.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000200*( np.cos(517.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000175*( np.cos(603.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000150*( np.cos(671.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000125*( np.cos(749.0*pi*float(x_index)/float(extent)) ))
    f[x_index]=f[x_index]-(0.000100*( np.cos(861.0*pi*float(x_index)/float(extent)) ))
```

Finally, for generating any of the three versions of prototypical 1-dimensional f, after the code provided above, add:

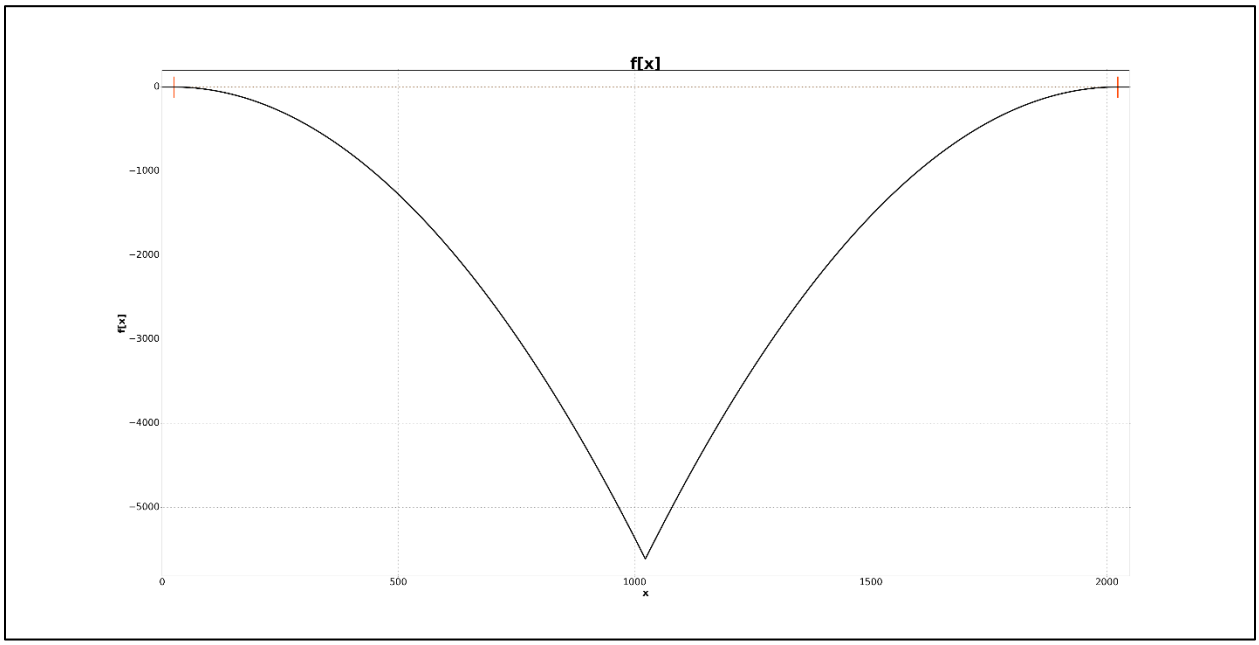
```
f=20000.0*f
```

To generate:



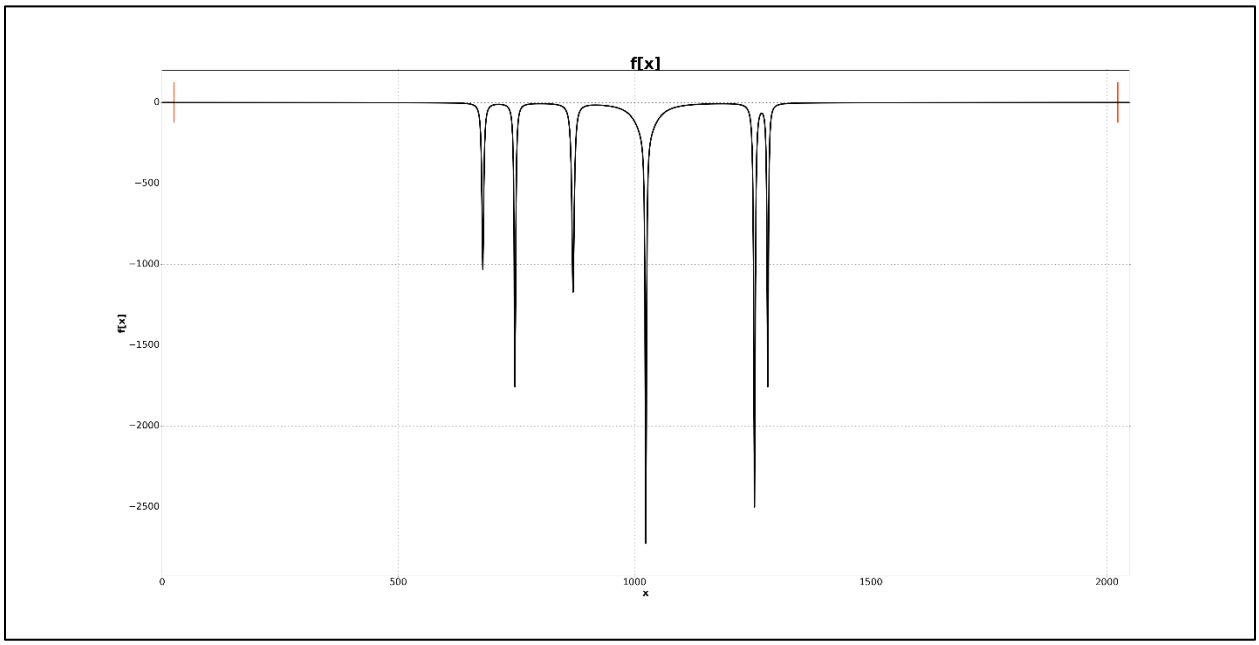

```
import numpy as np
extent=10000
f=np.zeros(extent)
x_c1=int(0.20*extent)
x_c2=int(0.925*extent)

kf=1.0

for x_index in range(0,extent):
        f[x_index]=-2480.0/(((x_index-x_c1)*(x_index-x_c1))+170000.0)# 2480.0, 170000.0
        f[x_index]=f[x_index]-(kf*(0.0985)/(((x_index-x_c2)*(x_index-x_c2))+6.5))
f=20000.0*f
```

To generate:




```
import numpy as np
extent=2048
q=np.zeros(extent)
q_x_offset=int(0.25*Num_x)
for x in range(0,Num_x):
        if (x>=xa and x<int(0.5*Num_x)):
                q[x]=q[x-1]-0.75
         if (x>=int(0.5*Num_x) and x<=Num_x-xa):
                q[x]=q[x-1]+0.75
q=-0.01*(q*q)
```

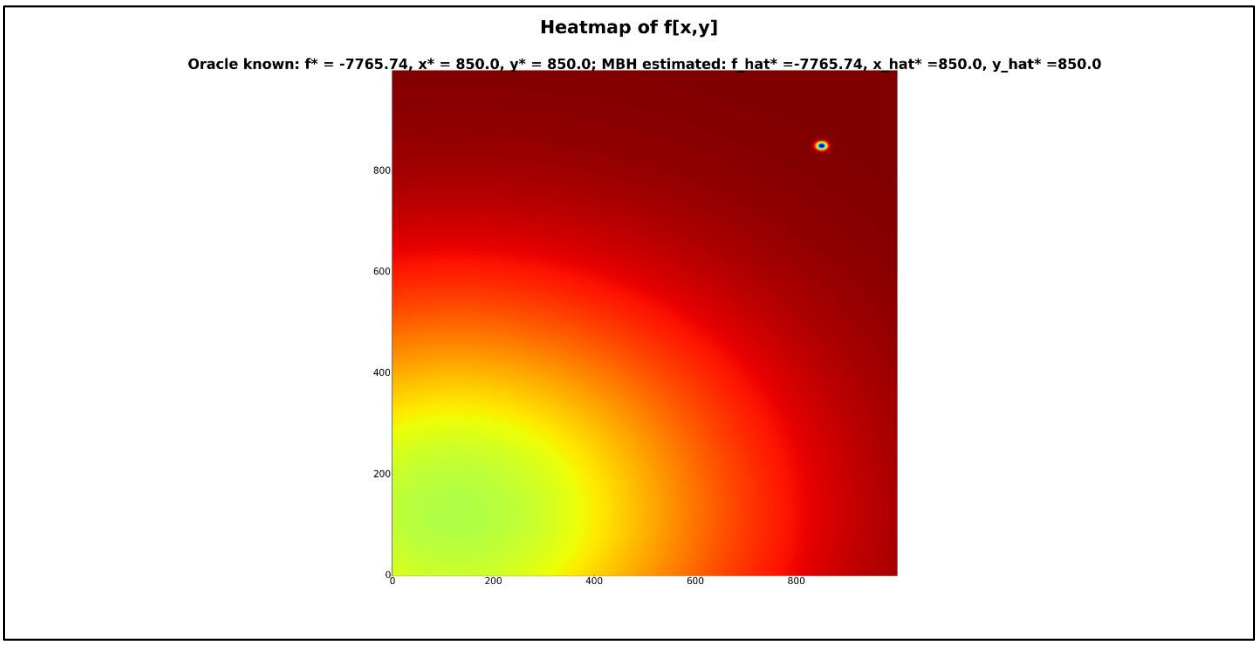
To generate:




```
import numpy as np
extent=2048
z_simp=np.zeros(extent)
Num_x=float(extent)
scale_f=1.5
arg_z_star=int(0.5*Num_x)
arg_z_star2=int(0.332*Num_x)
arg_z_star3=int(0.6125*Num_x)
arg_z_star4=int(0.365*Num_x)
arg_z_star5=int(0.626*Num_x)
arg_z_star6=int(0.425*Num_x)
arg_z_star7=int(0.675*Num_x)
arg_z_star8=int(0.295*Num_x)
arg_z_star9=int(0.545*Num_x)
arg_z_star10=int(0.29*Num_x)
arg_z_star11=int(0.72*Num_x)
arg_z_star12=int(0.24*Num_x)
arg_z_star13=int(0.75*Num_x)
arg_z_star14=int(0.48*Num_x)
arg_z_star15=int(0.51*Num_x)
arg_z_star16=int(0.43*Num_x)
 Kx=float(Num_x)/512.0
 scale_f_per_span_of_x=float(Num_x)/512.00
```

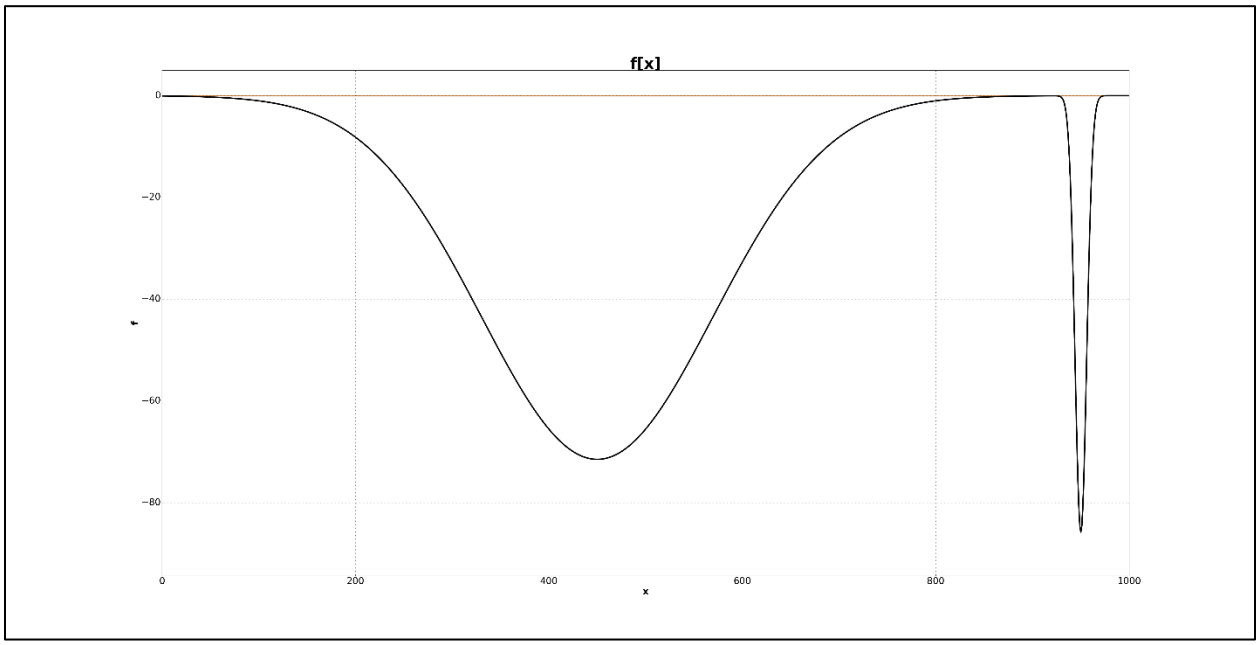
```
for x in range(0,Num_x):

    if (x>=x_a and x<=Num_x-x_a):
      z_simp[x]=-(scale_f*scale_f_per_span_of_x*1050.0/( (float(x-arg_z_star)*float(x-arg_z_star) )+2.5))
      z_simp[x]=z_simp[x]-(scale_f*scale_f_per_span_of_x*900.0/( (float(x-arg_z_star2)*float(x-arg_z_star2) )+5.25))
      z_simp[x]=z_simp[x]-(scale_f*scale_f_per_span_of_x*1350.0/( (float(x-arg_z_star3)*float(x-arg_z_star3) )+3.25))
      z_simp[x]=z_simp[x]-(scale_f*scale_f_per_span_of_x*950.0/( (float(x-arg_z_star4)*float(x-arg_z_star4) )+3.25))
      z_simp[x]=z_simp[x]-(scale_f*scale_f_per_span_of_x*800.0/( (float(x-arg_z_star5)*float(x-arg_z_star5) )+2.75))
      z_simp[x]=z_simp[x]-(scale_f*scale_f_per_span_of_x*1700.0/( (float(x-arg_z_star6)*float(x-arg_z_star6) )+8.75))
      z_simp[x]=z_simp[x]-(scale_f*scale_f_per_span_of_x*20500.0/( (float(x-arg_z_star)*float(x-arg_z_star) )+600.15))


 f=z_simp
```

To generate:



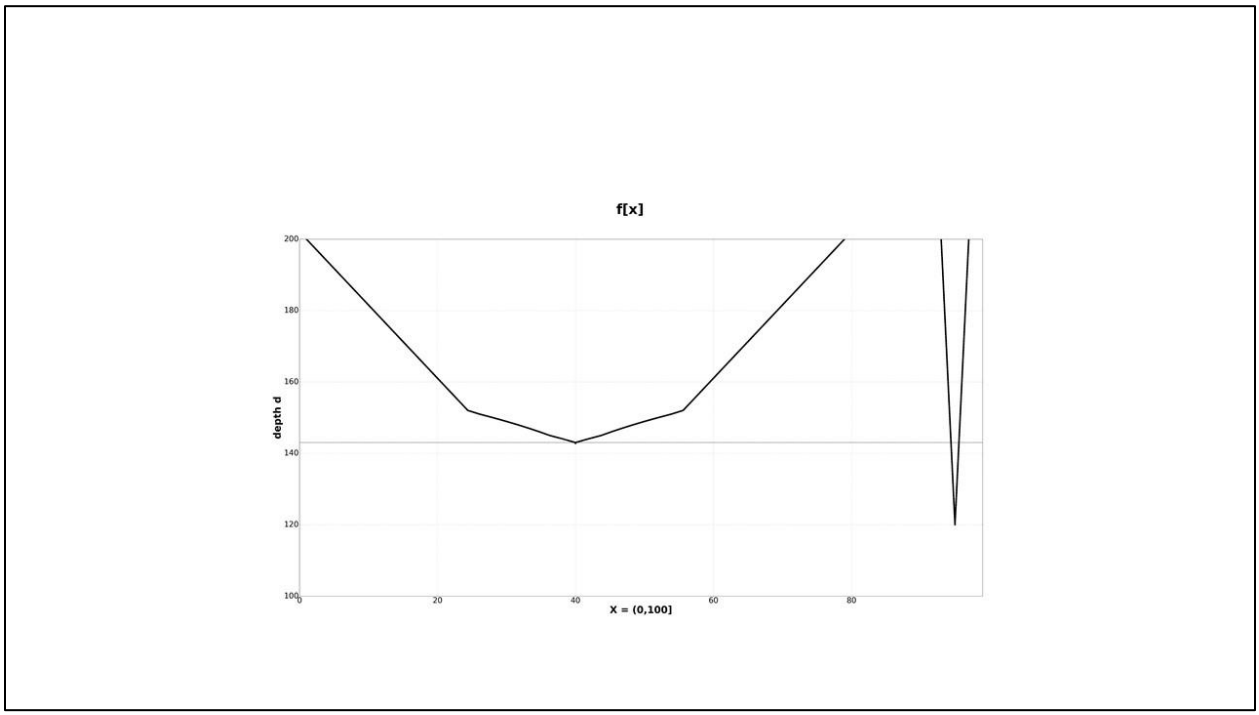

```
import numpy as np
extent=1000 # A lattice of 1 million squares
f=np.zeros((extent,extent))
u=np.zeros(extent)
v=np.zeros(extent)

x = np.arange(0,extent,1)
y = np.arange(0,extent,1)
X, Y = np.meshgrid(x, y)

g = -17000000.0*mlab.bivariate_normal(X-int(0.85*extent), Y-int(0.85*extent), 5, 7, 0, 0)
                        # Variance, variance, mean, mean
 h = -25550000000.0*mlab.bivariate_normal(X-int(0.125*extent), Y-int(0.125*extent), 300, 400, 0, 0)
                        # Variance, variance, mean, mean
f = 0.1*(g + h)
```

To generate:




```
import numpy as np
np.zeros(extent)
f=np.zeros(extent)
f1=np.zeros(extent)
f2=np.zeros(extent)
x_c1=int(0.45*extent)
x_c2=int(0.95*extent)
x_array=arange(0,extent,1)
f1=(-100.0)*mlab.normpdf(x_array,x_c1,120)
f2=(-6.0)*mlab.normpdf(x_array,x_c2,6)

f=(f1+f2)*215.0
```

To generate:




```
import numpy as np
loc_a=np.zeros(100)
loc_b=np.zeros(100)
loc_c=np.zeros(100)
loc_d=np.zeros(100)
depth=np.zeros(100)
local_min_at_depth=np.zeros(100)
for i in range(0,100):
        depth[i]=float(-i)-120
        loc_a[i]=(local_min-float(i))
        loc_b[i]=(local_min+float(i))
        loc_c[i]=(global_min-float(i))
        loc_d[i]=(global_min+float(i))
        if i<>23:
                local_min_at_depth[i]=(local_min)
                loc_a[i]=(local_min-(float(i)*0.4875))
                loc_b[i]=(local_min+(float(i)*0.4875))
        if i==29:
                loc_a[i]=(local_min-(float(i)*0.475))
                loc_b[i]=(local_min+(float(i)*0.475))
```

```
if i==30:
            loc_a[i]=(local_min-(float(i)*0.465))
             loc_b[i]=(local_min+(float(i)*0.465))
if i==31:
            loc_a[i]=(local_min-(float(i)*0.45))
            loc_b[i]=(local_min+(float(i)*0.45))
 if i==30:
            loc_a[i]=(local_min-(float(i)*0.40))
            loc_b[i]=(local_min+(float(i)*0.40))
if i==29:
            loc_a[i]=(local_min-(float(i)*0.35))
            loc_b[i]=(local_min+(float(i)*0.35))
if i==28:
            loc_a[i]=(local_min-(float(i)*0.30))
            loc_b[i]=(local_min+(float(i)*0.30))
if i==27:
            loc_a[i]=(local_min-(float(i)*0.25))
            loc_b[i]=(local_min+(float(i)*0.25))
if i==26:
            loc_a[i]=(local_min-(float(i)*0.20))
            loc_b[i]=(local_min+(float(i)*0.20))
if i==25:
            loc_a[i]=(local_min-(float(i)*0.15))
            loc_b[i]=(local_min+(float(i)*0.15))
 if i==24:
            loc_a[i]=(local_min-(float(i)*0.075))
            loc_b[i]=(local_min+(float(i)*0.075))
 if i<=23:
            loc_a[i]=(local_min)
            loc_b[i]=(local_min)

loc_c[i]=(global_min - (float(i)*0.025))
loc_d[i]=(global_min + (float(i)*0.025))
```

E. Details regarding the PyKep model

f in the use-case as returned by PyKep queries

In the use-case, f is the total change in velocity ($|\Delta v|$), in units of meters per second, attributable to the consumption of propellant on-board the spacecraft only. f does not include $|\Delta v|$ attributable to the consumption of propellant on-board the launch vehicle. The exclusion from f of $|\Delta v|$ attributable to the consumption of propellant on-board the launch vehicle is assured by the manner in which PyKep is set up and PyKep queries are coded. That will be specified below.

The manner in which the control parameters [x_1, x_2, x_3], where x_1 is the epoch of launch from Earth, x_2 is the duration of the voyage from Earth to the nearest approach to Jupiter, and x_3 is the duration of the voyage from the nearest approach to Jupiter to the nearest approach to Saturn, and x_1, x_2, and x_3 are real numbers in IEEE Floating Point representation, are mapped to f is performed by what trajectory optimization practitioners refer to as a transcription. The transcription used by PyKep to map a given vector of control parameters [x_1, x_2, x_3] to f[x_1, x_2, x_3] is called Multiple Gravity Assist (MGA). Epoch of launch from Earth refers to the exact time of launch including date, hour, minutes, seconds, and fractions of seconds (no more coarse than milliseconds).

Transcriptions typically make simplifying assumptions in both their mathematics and physics models. Sometimes these simplifying assumptions limit the precision and/or the fidelity (realism) with which the transcription maps [x_1, x_2, x_3] to f[x_1, x_2, x_3]. MGA makes the following simplifying assumptions:

- The physics model assumes 2-body gravitational dynamics, i.e., the spacecraft and one natural body. When very close to a planet, the model takes into account the gravity of only that planet. When not close to a planet, the model takes into account the gravity of only the Sun. This is a

simplification in the sense that at various times in the set of candidate solutions evaluated by the MBH process, the spacecraft is simultaneously affected by the gravity of the Sun and all other bodies in the solar system. However, the perturbations caused by additional bodies are small, and so the two-body approximation is reasonable and historically applied in preliminary design.

- Because the voyage is long compared to the propellant-consuming control maneuvers, the impact of the control maneuvers is modeled as a sum of small number of infinitesimal-duration impulse functions rather than as the integration of finite-duration forcing functions within a second order differential equation.
- Because, by Kepler's equations, the trajectories from Earth to Jupiter and Jupiter to Saturn approximately lie on respective conic sections, a numerical root-finding solution to Lambert's problem is used to map the travel time from x_1 to x_2, and x_2 to x_3, to conic sections and the changes in velocity that are required to transfer the spacecraft from one conic section to another. The use of Lambert's problem, and particularly a numerical solution to Lambert's problem, is an approximation. The use of Lambert's problem has various implications, including:
    - Lambert's problem assumes 2-body gravitational dynamics.
    - Numerical (e.g., root-finding) solutions to Lambert's problem are unstable when the angle of transfer from one conic section to another the other is 180 degrees. If the optimal solution that globally minimizes $f[x_1, x_2, x_3]$ requires a transfer angle equal to or nearly 180 degrees, use of Lambert's problem may return an erroneous f or an error message instead of a value for f. To prevent this, PyKep's MGA transcription includes a penalty function near every point $f[x_1, x_2, x_3]$ such that $[x_1, x_2, x_3]$ corresponds to a

transfer angle equal to or nearly 180 degrees. The problem with that is that if the global minimum $f[x_1, x_2, x_3]$ requires a transfer angle equal to or nearly 180 degrees, that f may never be found because it is obscured by the penalty function.

- The MGA transcription allows for a single maneuver at the point of closest approach (periapse) to each flyby planet. In the case of the example problem in this work, there is only one flyby planet (Jupiter). The maneuver $|\Delta v|$ is equal to the difference in periapse velocity between the incoming and outgoing hyperbolas at the flyby planet. These hyperbolas are approximated from the incoming and outgoing velocity vectors that are the result of solving Lambert's problem. This approximation is called the method of patched conics.

Wikipedia provides an article on Lambert's problem at Lambert's problem - Wikipedia. For interested readers, that article points to many papers on the subject including an article on MGA. In addition, MGA and its use of Lambert's problem are explained in [21, 68 - 71].

$\boldsymbol{X^F}$ in the use-case

Because the propellant tanks on-board the Pioneer 11 spacecraft were very small (in order to maximize the payload of scientific instruments), according to one expert at NASA Goddard's Space Flight Center, any $f > 10$ would have been totally un-flyable (non-feasible). In the use-case, that makes $\boldsymbol{X^F}$ appear to be, as best as can be determined, disconnected and sparse. In the opinion of a different expert at NASA Goddard's Space Flight Center, any $f > 3$ is totally un-flyable (non-feasible). That implies that $\boldsymbol{X^F}$ may be even more severely disconnected and sparse. That is simply a debate about how disconnected and sparse $\boldsymbol{X^F}$ may be. The use-case results provided $f << 3$. As shown in Table III in Chapter VI, across 100 MBH trials made, the present author's methodology produced a mean($\widehat{f^*}$) = *0.0067*, a max($\widehat{f^*}$) = *0.0250*, and a min($\widehat{f^*}$) = *0.0003*, where $\widehat{f^*}$ is the global minimum of f found by the MBH process in each of the *100* trials. Therefore, the debate

over whether $f > 3$ or $f > 10$ is totally un-flyable (non-feasible) is, in the context of the use-case, moot.

Assuring, by the set-up of PyKep, and how PyKep queries are coded, that changes in velocity due to the consumption of propellant on-board the launch vehicle are excluded from *f*

A small Python Class named MGA_intercept was provided to the present author by Jacob A. Englander as an aid in the development of the use-case software. Essential portions of that code and an example of its use are provided below. Comments are provided in non-bold Times New Roman font and Python code is provided in Bold Calabri font.

```
import MGA_intercept
```

Set up the MGA_intercept code and thereby PyKep.

```
Pioneer11 = MGA_intercept.MGA_intercept(seq=seq, t0=[-9863.0, -9948.0], tof=[800.0, 2000.0], vinf=12.0, tof_encoding='direct')
```

Launch from Earth may occur any time in 1974. Flight time from Earth to Jupiter is between 0 and 800 days. Flight time from Jupiter to Saturn is between 0 and 2000 days. The hyperbolic excess velocity attributable to the consumption of propellant on-board the launch vehicle at Earth departure may be up to, but no more than, 10 km/s. If it is more than 10 km/s then the spacecraft must maneuver, and *f* will increase because of a change in velocity attributable to the consumption of propellant on-board the launch vehicle at Earth. As long as vinf<=10.0 km/s, then the *f* returned by PyKep will not include any $|\Delta v|$ attributable to the consumption of propellant on-board the launch vehicle. Jacob A. Englander assured the present author that the change in velocity attributable to the consumption of propellant on-board the launch vehicle could not have exceeded 10.0 km/s because 10.0 km/s is larger than the launch vehicle injection velocity necessary to reach Jupiter. tof_encoding='direct' selects one of three

transcriptions provided by PyKep, one in which the entries of **x** directly represent time in units of days, e.g., x_2 = 986.598496203486… days.

The following code provides a crude candidate **x** for a query to PyKep and shows how the query is structured. Convert calendar dates to Modified Julian Dates relative to January 1, 2000 (MJD2000 dates) and flight times: April 6, 1973 = MJD 41778 which translates into MJD2000 = -9767 (this should really be an IEEE Floating Point value); Dec 3, 1974 = MJD 42384 which makes Time-of-Flight (TOF) from Earth to Jupiter 606 days (this should really be an IEEE Floating Point value); September 1, 1979 = MJD 44117 which makes Time-of-Flight (TOF) from Jupiter to Saturn 1733 days (this should really be an IEEE Floating Point value).

```
x = [-9766.0, 606.0, 1733.0]
```

The precision of this example of **x** is inadequate for the use-case. The following line of code evaluates F(**x**):

```
F = Pioneer11.fitness(x)
```

Because each query of PyKep returns F as a Python list, that list needs to be parsed by adding the line of code:

```
f_candidate=F[0]
```