TUNING MONOTONIC BASIN HOPPING: IMPROVING THE EFFICIENCY OF STOCHASTIC SEARCH AS APPLIED TO LOW-THRUST TRAJECTORY OPTIMIZATION

Jacob A. Englander*and Arnold C. Englander†

Abstract:

Trajectory optimization methods using monotonic basin hopping (MBH) have become well developed during the past decade [1, 2, 3, 4, 5, 6]. An essential component of MBH is a controlled random search through the multi-dimensional space of possible solutions. Historically, the randomness has been generated by drawing random variable (RV)s from a uniform probability distribution. Here, we investigate the generating the randomness by drawing the RVs from Cauchy and Pareto distributions, chosen because of their characteristic long tails. We demonstrate that using Cauchy distributions (as first suggested by J. Englander [3, 6]) significantly improves monotonic basin hopping (MBH) performance, and that Pareto distributions provide even greater improvements. Improved performance is defined in terms of efficiency and robustness. Efficiency is finding better solutions in less time. Robustness is efficiency that is undiminished by (a) the boundary conditions and internal constraints of the optimization problem being solved, and (b) by variations in the parameters of the probability distribution. Robustness is important for achieving performance improvements that are not problem specific. In this work we show that the performance improvements are the result of how these long-tailed distributions enable MBH to search the solution space faster and more thoroughly. In developing this explanation, we use the concepts of sub-diffusive, normally-diffusive, and super-diffusive random walks (RWs) originally developed in the field of statistical physics.

1. Introduction

The optimization of low-thrust interplanetary trajectories is a highly complex task. Such design problems are characterized by hundreds to thousands of decision variables and tens to hundreds of constraints. Many locally optimal solutions often exist. These problems are often solved using a nonlinear programming (NLP) problem solver but all such solvers require a good initial guess. Historically initial guesses are generated by intuition or by solving a reduced-fidelity version of the problem, such as by approximating a low-thrust orbit transfer as a Lambert arc. This can result in a great deal of hands-on work and one can still miss the globally optimal solution if it is non-intuitive.

Several researchers have sought alternative ways to generate initial guesses for low-thrust trajectory design problems using a stochastic search method called monotonic basin hopping (MBH) [1, 2, 3, 4, 5, 6]. MBH is a hybrid of the classic multi-start algorithm, an NLP solver,

*Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, 8800 Greenbelt Road, Greenbelt, MD 20770, 301-286-4710, jacob.a.englander@nasa.gov

†Vice President of Research and Development, Deployment Technologies, Inc., 250 Commercial St., Manchester, NH, 03101, 603-836-0221, aenglander@dtechinc.com

and a stochastic search step. First a random point is chosen in the decision space and the NLP solver is run from that point to attempt to find a feasible solution. This step is repeated until a feasible solution is found. MBH then attempts to improve upon the current solution by "hopping," i.e. adding small random perturbations to the decision vector and re-optimizing using the NLP solver. Over many iterations, MBH progresses towards a better solution.

In this work the concept of MBH is further developed and applied to a challenging problem of interest to the trajectory design community. Goddard's low-thrust trajectory design tool Evolutionary Mission Trajectory Generator (EMTG) [2, 3, 4, 5, 6] is used as a testbed for the MBH+NLP global search heuristic, along with a difficult benchmark problem: a low-thrust mission to Uranus.

Our goals for this paper were simple and practical: First, enable MBH to find better solutions, faster (in less computer run-time). We refer to that as improving MBH efficiency. Second, enable MBH to be less sensitive to (a) the tuning of their excursion parameters (called step-size in the classical literature on MBH wherein the RVs are drawn from uniform probability distributions), and (b) the effects of boundaries surrounding, and the internal constraints within, the space of possible solutions. We refer to that as improving MBH robustness. Our experiments show that we have achieved both goals by driving MBH with RVs drawn from specific types of long-tailed probability distributions, instead of uniform or Gaussian distributions.

Our third goal was to do everything we could towards assuring that the improvements we achieved were based on generalizable principles rather than accidental good fortune with the problem we chose to solve. For that reason, much of this paper is devoted to establishing a theory, based on the diffusion of random walks through in-homogenous media, to predict and explain the improvements in efficiency and robustness that we achieved in our experiments.

2. Low-Thrust Trajectory Modeling

2.1. Mission Architecture

Three layers of event types are defined in this work: *missions*, *journeys*, and *phases*. A *mission* is a top-level container that encompasses all of the events including departures, arrivals, thrust arcs, coast arcs, and flybys. A journey is a set of events within a mission that begin and end at user-defined targets. For example, the interplanetary cruise portion of the Cassini mission was composed of a single journey that began at Earth and ended at Saturn. JAXA's Hayabusa mission, which rendezvoused and took samples from near Earth asteroid Itokawa, had two journeys - one from Earth to Itokawa, and one from Itokawa to Earth. NASA's Dawn mission is also composed of two journeys, one from Earth to Vesta and one from Vesta to Ceres. Each journey is composed of one or more *phases*. Like a journey, a phase begins at a planet and ends at a planet, but unlike the end points of a journey, the end points of a phase may be chosen either by a human analyst or by an autonomous method. In the EMTG, an integer genetic algorithm (GA) is often used to find the optimal

sequence of flyby targets, and thus the number of phases and the endpoints of each phase [7, 2, 3]. However most commonly the GA is run only once at the beginning of a mission design process and after that the flyby sequence is specified by the user. The flyby selection process is out-of-scope for this paper and will not be discussed further. Figure 1 is a block diagram of a mission using the above nomenclature.

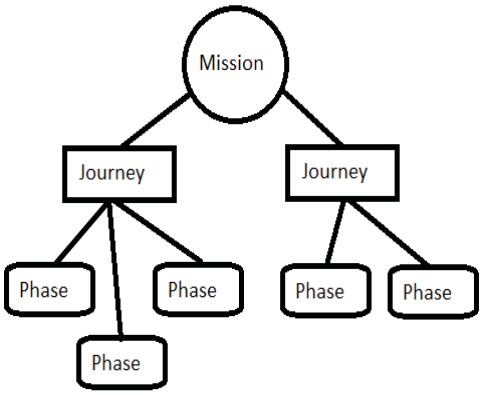

Figure 1: Anatomy of a Mission

2.2. Multiple Gravity Assist with Low-Thrust (MGALT)

Global search is performed in EMTG using the multiple gravity assist with low-thrust (MGALT) model, a simplified trajectory model that combines the well-known Sims-Flanagan transcription [8] with a simple patched-conic flyby model. The Sims-Flanagan transcription is a widely used method in which the continuous-thrust trajectory is discretized into many small time steps, and the thrust applied during each time step is approximated as a small impulse placed at the center of the time step. The trajectory is propagated between control points by solving Kepler's problem [8]. The Sims-Flanagan transcription, when used with an NLP solver such as Sparse Nonlinear Optimizer (SNOPT) and a suitable initial guess, is very fast and robust. It is considered to be a "medium-fidelity" transcription and is used in existing software packages such as Gravity Assisted Low-thrust Local Optimization Program (GALLOP) [9], Mission Analysis Low-Thrust Optimization (MALTO) [10], and Parallel Global Multiobjective Optimizer (PaGMO) [11].

In the classical Sims-Flanagan transcription, the optimizer chooses the three components of an impulsive $\Delta\mathbf{v}$ vector at the center of each time-step. In order to improve the robustness of the solver, a modified transcription known as "up-to-unit vector control" is used in EMTG, where instead of choosing the $\Delta\mathbf{v}$ vector directly the optimizer instead chooses a control 3-vector in $[-1.0, 1.0]$ that is multiplied by the maximum Δv that the spacecraft can produce in that time-step. The magnitude of the control vector is then bounded in the range $[0.0, 1.0]$, i.e.,

$$\Delta\mathbf{v}_i = \mathbf{u}_i \Delta v_{max,i}, \|\mathbf{u}_i\| \leq 1.0 \tag{1}$$

where

$$\Delta v_{max,i} = \frac{D n_{available} T_{\max} (t_f - t_0)}{mN} \tag{2}$$

where D is the thruster duty cycle, $n_{available}$ is the number of available thrusters, $T_{\max}$ is the maximum available thrust from one thruster, t_0 and t_f are the beginning and ending times of the time step, m is the mass of the spacecraft at the center of the time step, and N is the number of time steps in the phase.

In each phase of an multiple gravity assist with low-thrust (MGALT) mission, the trajectory is propagated forward from the first endpoint (i.e. planet) and backward from the second endpoint. The trajectory is propagated by solving Kepler's equation and the spacecraft mass is propagated by assuming a constant mass flow rate across the each time-step. The specific Kepler propagator algorithm used in EMTG is a Laguerre-Conway method [12, 13]. A set of nonlinear constraints are applied to ensure continuity in the center of the phase,

$$\mathbf{s}_{mf} - \mathbf{s}_{mb} = \begin{bmatrix} \Delta x & \Delta y & \Delta z & \Delta v_x & \Delta v_y & \Delta v_z & \Delta m \end{bmatrix} = \mathbf{0} \tag{3}$$

The optimizer also chooses the initial and final velocity vectors for each phase. If a phase begins with a launch, the magnitude of the initial velocity vector is used with a launch vehicle model to determine the initial mass of the spacecraft as described later in this work. If a phase begins with a planetary flyby, two nonlinear constraints are applied to ensure that the flyby is feasible. First, the incoming and outgoing velocity vectors with respect to the planet must be equal,

$$v_\infty^+ - v_\infty^- = 0 \tag{4}$$

where v_∞^- and v_∞^+ are the velocities before and after the flyby, respectively. Second, the spacecraft may not fly closer to the planet than some user-specified minimum flyby distance:

$$\frac{\mu_{planet}}{v_\infty^2} \left[\frac{1}{\sin(\frac{\delta}{2})} - 1 \right] - (r_{planet} + h_{safe}) \geq 0 \tag{5}$$

where

$$\delta = \arccos \left[\frac{\mathbf{v}_\infty^- \cdot \mathbf{v}_\infty^+}{(v_\infty^-)^2 (v_\infty^+)^2} \right] \tag{6}$$

Here μ_{planet} is the gravitational parameter of the planet, r_{planet} is the radius of the planet, and h_{safe} is the user-defined minimum altitude.

Figure 2 is a diagram of a simple low-thrust mission to Jupiter with one Earth flyby using the MGALT model. The continuity constraints are deliberately left unsatisfied in the diagram to illustrate where they must be applied.

2.3. Ephemeris and Spacecraft Hardware Modeling

EMTG interfaces with the SPICE ephemeris library to provide the position and velocity of each body visited in a mission [14]. In addition, EMTG contains accurate models of spacecraft power systems, thrusters, and launch vehicles. While the details of these models are out of scope for this paper, they may be found in a previous work by these authors [6].

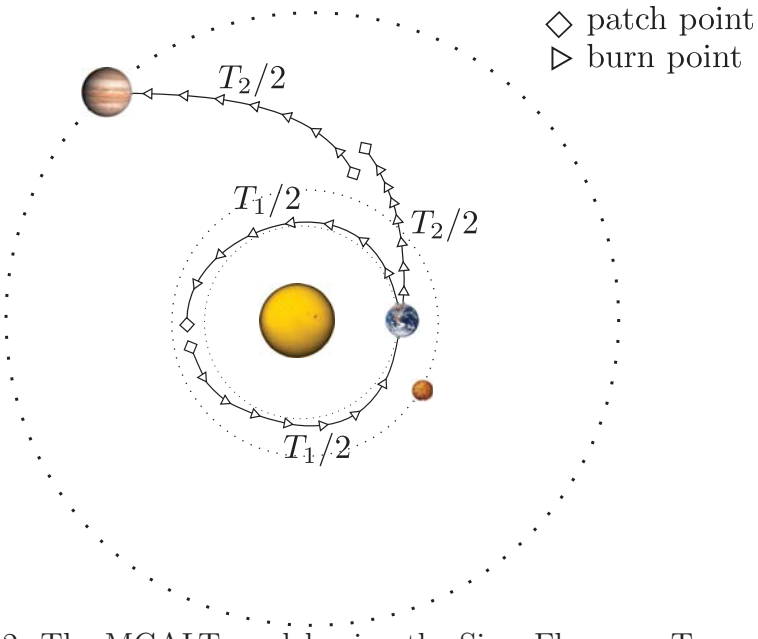

Figure 2: The MGALT model using the Sims-Flanagan Transcription

3. Optimization Method

3.1. Nonlinear Programming

The MGALT and finite-burn low-thrust (FBLT) problems may be formulated as nonlinear programming (NLP) problems. NLP problems explicitly model nonlinear constraints. The optimizer solves a problem of the form:

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\text{Subject to:} \\ &\mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub} \\ &\mathbf{c}(\mathbf{x}) \leq \mathbf{0} \\ &A\mathbf{x} \leq \mathbf{0} \end{aligned} \tag{7}$$

where $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ are the lower and upper bounds on the decision vector, $c(\mathbf{x})$ is a vector of nonlinear constraint functions, and A is a matrix describing any linear constraints (*i.e.* time constraints).

The trajectory optimization problems posed by EMTG are very large, composed of hundreds or thousands of decision variables and tens or hundreds of constraints. A large-scale NLP solver such as SNOPT [15] is therefore required to solve the problems of interest in an efficient and robust manner. However, SNOPT, like all NLP solvers, requires an initial guess of the solution and will tend to converge to a solution in the neighborhood of that initial guess. The next section will address how EMTG generates this initial guess in a fully automated manner.

3.2. Sparsity and Specification of the Problem Jacobian

Both MGALT and FBLT are sparse problem transcriptions, meaning that each nonlinear constraint is dependent only on a small subset of the decision variables. SNOPT can take advantage of the problem sparsity to speed the solution process, but only if the sparsity pattern is known. While SNOPT is capable of estimating the sparsity pattern by performing random finite-differencing operations, this estimate is often inaccurate. Instead it is preferable to specify the sparsity pattern *a priori*. However this can be a time-consuming process and not easily generalizable to different problem types. This problem is avoided in EMTG using a system of automated sparsity pattern construction. At run time EMTG automatically constructs text descriptions of each decision variable and constraint. The dependency of each constraint on each variable is determined searching the text descriptions using a set of rules hard-coded into the program. This is done instead of hard-coding the sparsity pattern itself so that the programmer can quickly and easily add new constraints without having to modify a hard-coded sparsity pattern.

3.3. Monotonic Basin Hopping

Monotonic Basin Hopping (MBH) [16] is an algorithm for finding globally optimal solutions to problems with many local optima. MBH works on the principle that many real-world problems have a structure where individual local optima, or "basins" tend to cluster together into "funnels" where one local optimum is better than the rest. A problem may have several such funnels. MBH was originally developed to solve molecular conformation problems in computational chemistry, but has been demonstrated to be effective on various types of interplanetary trajectory problems [1, 17, 18, 2, 3, 4].

First, an initial point $\mathbf{x}$ is randomly chosen. The NLP solver is run using $\mathbf{x}$ as the initial guess. If the NLP solver finds a feasible solution, then that new point $\mathbf{x}^*$ is adopted as the new current point. If the NLP solver does not find a feasible solution, then a new random point is chosen. Once a feasible solution is found, MBH will attempt to "hop" from the feasible and locally optimal $\mathbf{x}^*$ to a better point. This is a two-step process: first a small random perturbation vector is added to $\mathbf{x}^*$, producing a new $\mathbf{x}'$, and then the NLP solver is run. If the resulting solution is both feasible and superior to $\mathbf{x}^*$, then it is adopted as the new $\mathbf{x}^*$ and the hopping process begins again. Otherwise, MBH attempts a new hop from the current $\mathbf{x}^*$ and an "impatience" counter $N_{\text{not improve}}$ is incremented. If $N_{\text{not improve}}$ exceeds a user-defined threshold value $\text{Max}_{\text{not improve}}$, then MBH resets and generates a new random $\mathbf{x}$. Each feasible solution is stored in an archive.

The global reset and local hop operators allow MBH the capability both to *explore* the entire solution space and also to *exploit* the local minima in the current "funnel." That is, MBH will explore the solution space via the global reset operator until a feasible solution is found. MBH will then exploit the local funnel, i.e. search for a better local minimum in the vicinity of the current local minimum. The exploitation process continues until the algorithm fails to improve the current point $\text{Max}_{\text{not improve}}$ times, and then MBH will switch back to exploring the entire solution space.

MBH is run until either a specified number of iterations (trial points attempted) or a maximum CPU time is reached, at which point the best solution stored in the archive is returned as the solution to the outer-loop. MBH has three parameters - the stopping criterion, the parameter $N_{\text{not improve}}$, and the type of random step used to generate the perturbed points $\mathbf{x}'$. In the classical version of MBH, the random step is drawn from a uniform probability distribution in $[-\sigma, \sigma]$.

The contribution of this paper is the investigation, both experimentally and theoretically, of using RVs from distributions other than uniform for MBH, specifically Cauchy and Power Law distributions chosen because of their very long tails as originally suggested in [3]. Our experimental results show that, for our test problem, these long-tailed distributions significantly improved the MBH performance in two respects: efficiency, meaning finding better solutions in less time; and robustness, meaning efficiency that is undiminished by the boundary conditions and internal constraints of the optimization problem, and by variations in the parameters of the probability distribution. Robustness is important for achieving performance improvements that are not problem specific. The theory section explains that these performance improvements are the generalizable result of how these long-tailed distributions enable the MBH to search the solution space faster and more thoroughly. In developing this explanation, we use the concepts of sub-diffusive, normally-diffusive, and super-diffusive random walks originally developed in the field of statistical physics.

In addition, in the classical version of MBH, SNOPT occasionally freezes in the middle of a local optimization. This behavior disrupts the MBH global search, so EMTG contains a timer that ends any SNOPT run that continues for longer than some threshold time typically set to a few minutes. The pseudocode for the classical MBH is listed in Algorithm 1.

3.4. Upper and Lower Bounds on the Decision Variables

EMTG is designed to operate with as little user oversight as possible, which in turn means that the user should not be required to choose upper and lower bounds on each decision variable. The bounds however are still very important because the range of allowable values is used by both MBH and SNOPT. Some parameters, notably the launch date and initial v_∞, must still have bounds chosen by the user. However most of the parameters have bounds chosen heuristically by a set of simple laws, as listed in Table 1. Note particularly that the flight time bounds are functions of τ_i, the orbital periods of the bodies that define the endpoints of each phase.

Table 1: Automated Choosing of Inner-Loop Bounds for Continuous Thrust, Sims Flanagan Problems

Parameter	Lower Bound	Upper Bound
Launch date	user-defined	user-defined
Stay time between journeys	user-defined	user-defined
RLA	0.0	2π
DLA	user-defined	user-defined
$v_{\infty-launch}$	0.0	user-defined
For each phase:		
Flight time	$\tau/2$ (repeated flyby of same planet)	5τ (repeated flyby of same planet)
	$0.1min\left(\tau_1, \tau_2\right)$	$2.0max\left(\tau_1, \tau_2\right)$ (outermost body has $a < 2AU$)
		$max\left(\tau_1, \tau_2\right)$ (outermost body has $a \geq 2AU$)
	maximum of 600 days	minimum of 2500 days
Mass at end of phase	0.0 kg	user-defined spacecraft maximum mass
For each non-rendezvous phase:		
$v_{\infty-incoming-x,y,z}$	user-defined (if terminal intercept)	user-defined (if terminal intercept)
	-10.0 km/s (if intermediate phase)	10.0 km/s (if intermediate phase)
For each phase after the first:		
$v_{\infty-outgoing-x,y,z}$	-10.0 km/s	10.0 km/s

Algorithm 1 Monotonic Basin Hopping (MBH)

while not hit stop criterion **do**
  $N_{\text{not improve}} = 0$
  generate random point $\mathbf{x}$
  run NLP solver to find point $\mathbf{x}^*$ using initial guess $\mathbf{x}$
  if ($\mathbf{x}^*$ is a feasible point) **then**
    $\mathbf{x}_{current} = \mathbf{x}^*$
    while $N_{\text{not improve}} < \text{Max}_{\text{not improve}}$ **do**
      generate $\mathbf{x}'$ by randomly perturbing $\mathbf{x}_{current}$
      run NLP solver to find point $\mathbf{x}^*$ using initial guess $\mathbf{x}'$
      if $\mathbf{x}^*$ is feasible **and** $f(\mathbf{x}^*) < f(\mathbf{x}_{current})$ **then**
        $\mathbf{x}_{current} = \mathbf{x}^*$
        $N_{\text{not improve}} = 0$
      else
        increment $N_{\text{not improve}}$
      end if
    end while
    save $\mathbf{x}^*$ to archive
  end if
end while
return best $\mathbf{x}^*$ in archive

Parameter	Value
Mission parameters	
Launch Vehicle	Atlas V 551
Earliest allowed launch date	January 1st, 2022
Latest allowed launch date	January 1st, 2032
Maximum flight time	20 years
Arrival type	rendezvous
Propulsion system	VSI in range $[1000, 2500]$, $\eta = 0.6$
BOL power output	3.0 kW
Power system decay rate	2% per year
Spacecraft bus power	300 W
Solver parameters	
Flyby sequence	EESU
Number of time-steps	40
MBH run time	96 hours
SNOPT feasibility tolerance	1.0e-5
Objective function	maximize final mass

Table 2: Problem assumptions for the benchmark mission to Uranus

4. Investigation

4.1. Benchmark Problem - a Radioisotope-Electric Propulsion (REP) Mission to Uranus

The benchmark problem presented here is a hypothetical mission to Uranus in the 2022-2032 time frame using a variable specific impulse (I_{sp}) (VSI) radioisotope-electric propulsion (REP) system. The beginning of life (BOL) power is 3 kW, subject to a 2% per year degradation rate. 300 W is reserved to power the spacecraft bus at all times. The propulsion system is a VSI thruster with $\eta = 60\%$ that can throttle between I_{sp}s of 1000 and 2500 seconds. EMTG may choose the optimal I_{sp} independently for each time step. The mission is optimized for a maximum flight time of 20 years and an Earth-Earth-Saturn-Uranus flyby sequence as described in Table 2. This problem was chosen as a benchmark because it is particularly difficult - the optimizer must choose the launch epoch, the flight time for each phase, the magnitude and direction of the initial departure velocity, the mass at each body encounter, and the control unit vector and I_{sp} at each time-step. Forty time-steps are chosen per phase, a good balance between fidelity of the solution and tractability of the problem to the NLP solver. The benchmark problem has 503 variables and 146 constraints.

During the experiments conducted in this work, the best solution found to the benchmark problem delivered 3007 kg to Uranus for a cost function value of -0.3007. Figure 3 is a plot of the best solution found, which serves as a point of comparison for all of the experimental runs in this work. Figure 4 is a zoomed-in view of the same trajectory, focused on the path of the spacecraft in the inner solar system.

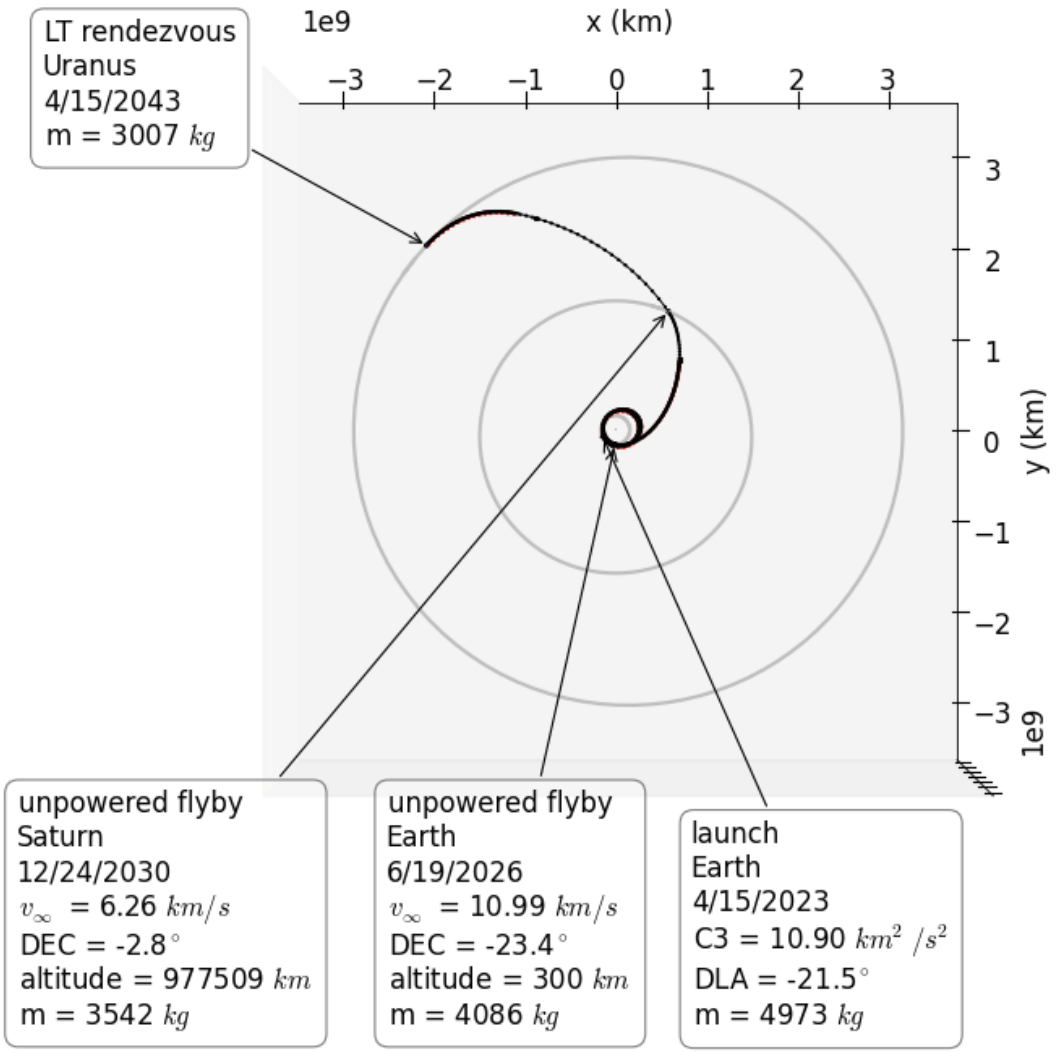

Figure 3: Best solution found to the VSI Earth-Earth-Saturn-Uranus problem

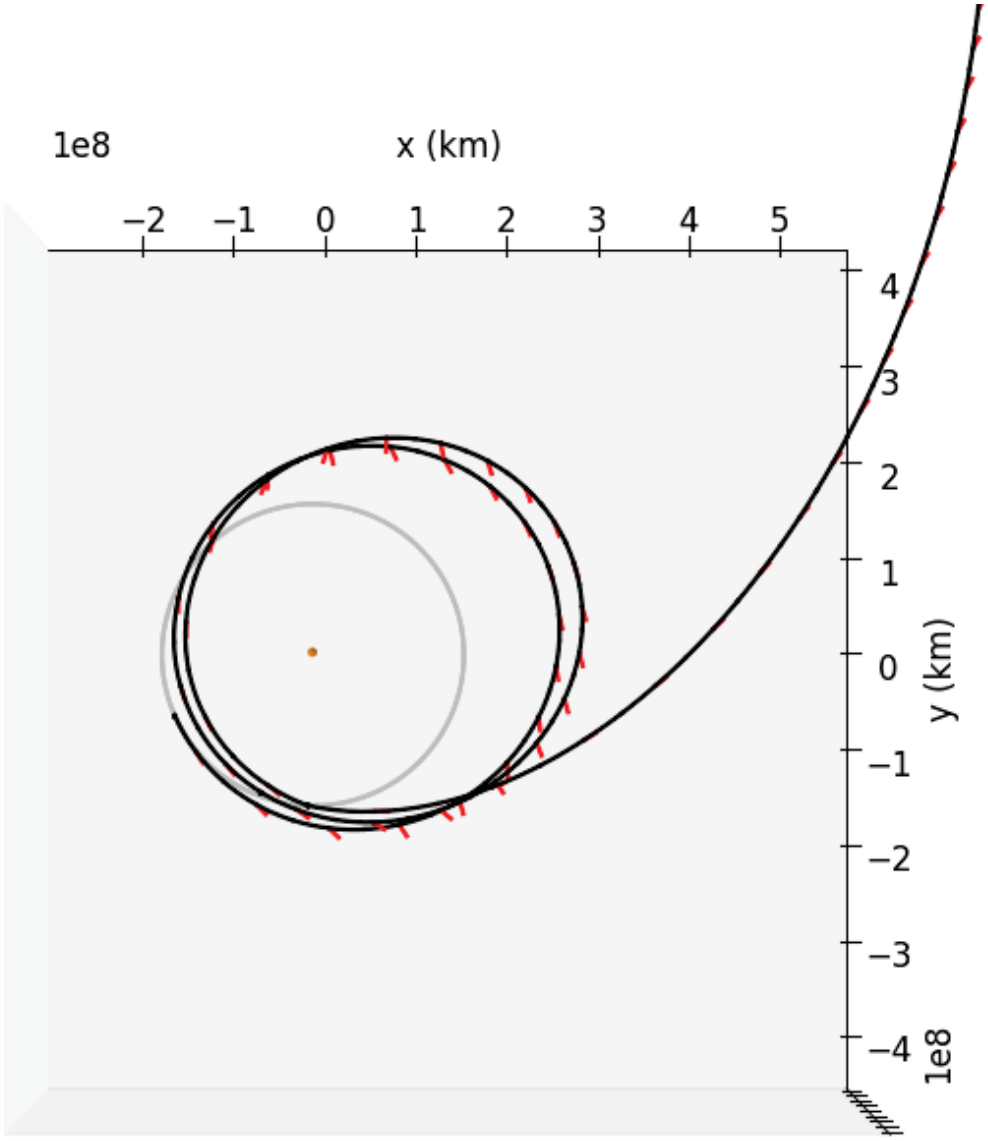

Figure 4: Best solution found to the VSI Earth-Earth-Saturn-Uranus problem, zoomed in to view the inner solar system trajectory

4.2. Experiment

The goal of the experiment was to test and evaluate our hypothesis that RWs driven by RVs drawn from long-tailed probability distributions, particularly Cauchy and bi-polar Pareto distributions, provide significantly more efficient and robust MBH performance than RWs driven by RVs drawn from either the "no-tail" uniform distributions classically used with MBH or "normal-tailed" Gaussian distributions.

The experiment consisted of running the MBH+NLP optimizer on the problem described in Section 4.1.. The optimizer was run using 64 different cases of RWs. The 64 cases were divided into 4 case types-one case type per probability distribution being investigated: uniform, Gaussian, Cauchy, and bi-polar Pareto.

Each probability distribution was provided with an array of 16 different excursion parameters. In the case of uniform distributions, the excursion parameter is the absolute value of the lower and upper bounds on the values of the RVs that can be drawn. For example, a uniform distribution with an excursion parameter of 0.1 provides a uniform probability of drawing a random value between -0.1 and 0.1 (inclusive), and zero probability of drawing RVs with values less than -0.1 or greater than 0.1. In the case of a Gaussian distribution, the excursion parameter is the standard deviation of the distribution. In the case of a Cauchy distribution, the excursion parameter is what probability theory specialists refer to as the scale factor, and in the case of a bi-polar Pareto distribution, the excursion parameter is what probability theory specialists call the Power Law exponent alpha.

For each of the four probability distributions, the "average" value of the RVs was set to zero, meaning that range of possible values is symmetric around zero. That means that, for each of the four probability distributions, at every step, the RW is equally likely to move in any direction. The reason we put "average" in quotes will become clear in the next paragraph.

In the case of a uniform distribution, the "average" is automatically set to zero by setting the lower bound to be negative one times the upper bound. In the case of a Gaussian distribution the "average" is set to zero when sets the mean to zero. In the case of Cauchy and bi-polar Pareto distributions, the "average" is undefined because the infinitely log tails of these distributions prevents their being integrated, which is a requirement for defining an average. However probability theory allows for the range of possible values drawn from Cauchy and bi-polar Pareto distributions to be centered on zero by using setting parameter called location to zero.

Table 3 lists the sixteen excursion parameters used for each of the four distributions. For each of the four distributions, the range of excursion parameters was designed to be wide enough such that the efficiency of MBH could be evaluated as a function of the excursion parameter.

Distribution	Excursion Parameters
Uniform($-stepsize, stepsize$)	$stepsize = [0.01, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18, 0.20, 0.22, 0.24, 0.26, 0.28, 0.30]$
Gaussian($mean, \sigma$)	$mean = 0.0$ $sigma = [0.01, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18, 0.20, 0.22, 0.24, 0.26, 0.28, 0.30]$
Cauchy($location, scale$)	$location = 0.0$ $scale = [0.000125, 0.00025, 0.0005, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.008, 0.010, 0.012, 0.014, 0.016, 0.018, 0.020]$
Bi-polar Pareto($location, alpha$)	$location = 0.0$ $alpha = [1.0025, 1.0050, 1.0075, 1.010, 1.015, 1.020, 1.030, 1.040, 1.050, 1.060, 1.070, 1.080, 1.090, 1.10, 1.11, 1.12]$

Table 3: Excursion parameters for each probability distribution

In each case we ran MBH for a total of 10000 steps, down-sampled as 1000 serial bins of 10 steps to simplify analysis of the data set. For every bin we recorded the best fitness found by that case thus far. We refer to each case's run of 1000 bins of 10 serial steps as a path.

The RVs were generated using the functions generated in Table 4. The Uniform, Gaussian, and Pareto RVs were generated from their probability density functions (PDFs) while the Cauchy RVs were generated from the cumulative density function (CDF). In Table 4, $r = uniform(0.0, 1.0)$ and s is a fair coin flip, *i.e.* has equal probability of -1.0 and 1.0.

Distribution	RV Generator
Uniform	$2\,(r - 0.5)$
Gaussian	$\frac{s}{\left(\sigma\sqrt{2\pi}\right)} \exp \frac{-r^2}{2\sigma^2}$
Cauchy	$\tan\left(\pi\,(r - 0.5)\right)$
Pareto	$\frac{s}{\epsilon} \frac{(\alpha - 1.0)}{\left(\frac{\epsilon}{\epsilon + r}\right)^{-\alpha}}$

Table 4: RV Generators

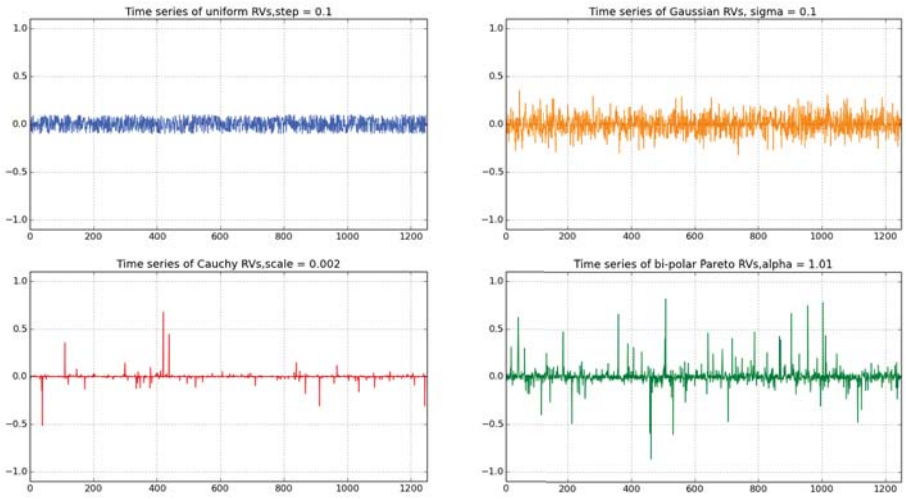

Figure 5: Times series of uniform, Gaussian, Cauchy, and bi-polar Pareto distributed RVs

Figure 5 illustrates the one-dimensional behavior of these RVs, Figure 6 illustrates their histograms (approximations to their probability density functions), Figure 7 shows a detail of the positive-side tail of each histogram, and Figure 8 shows two-dimension walks driven by each of them. (The space shown is bounded but does not have any internal constraints.) Together, Figures 5 through 8 give a sense of what Benoit Mandelbrot [19] called the "mildness" of uniform and Gaussian distributed RVs, compared to the "wildness" of Cauchy and bi-polar Pareto distributed RVs.

4.3. Results

Figure 9 shows the best fitness found thus far at each bin of ten serial steps by MBH driven by each of the four probability distributions using their respective best excursion parameters. The best solution was found by MBH driven by the Cauchy distribution, but a very nearly equally as good solution was found in much less search time by MBH driven by the bi-polar Pareto distribution. The bi-polar Pareto distribution is therefore more efficient in the sense of finding a better solution. Both the Cauchy and the bi-polar Pareto MBH found a better solution and were more efficient than the Gaussian and Uniform MBH. In the legend of Figure 9, "best path" refers to the path using that distribution's best excursion parameter, in terms of best fitness found in the experiment.

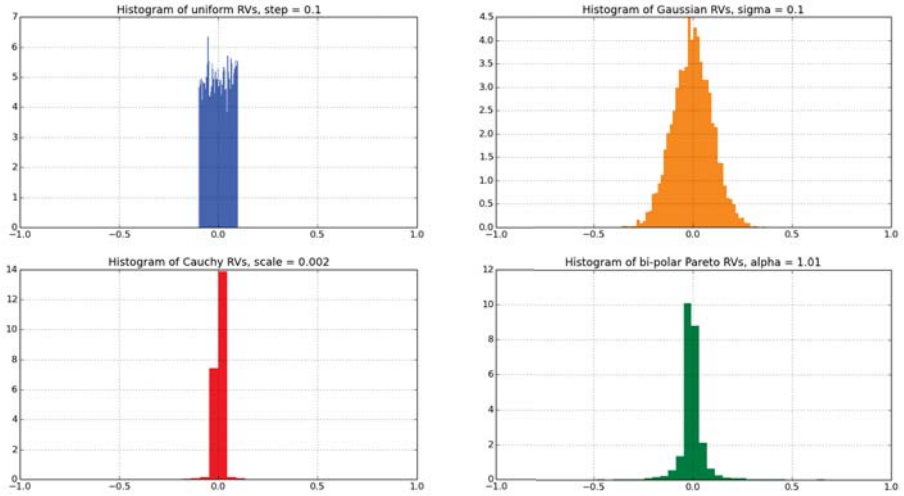

Figure 6: Histograms of times series of uniform, Gaussian, Cauchy, and bi-polar Pareto distributed RVs

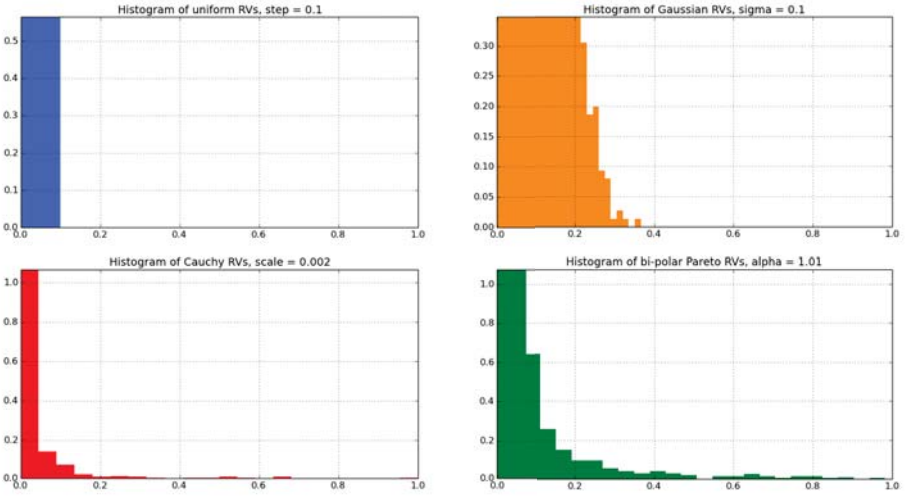

Figure 7: Magnifications of the tails of the histograms shown in Figure 6

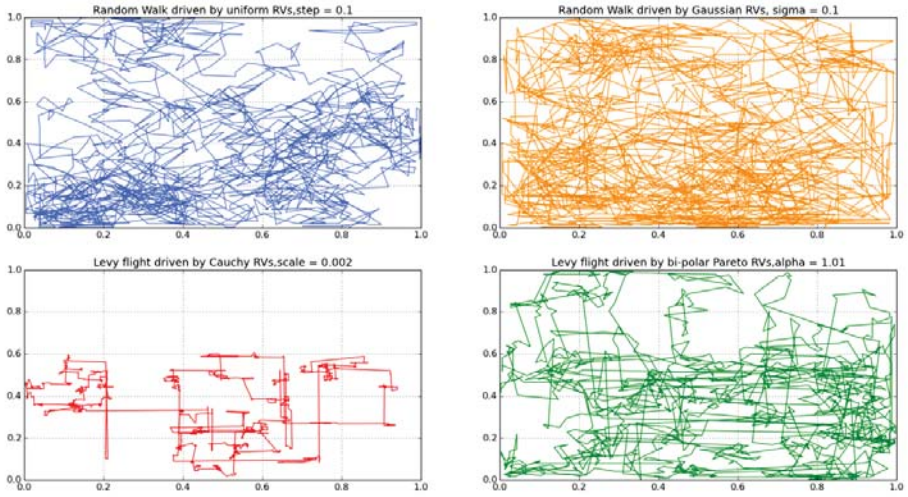

Figure 8: Two-dimensional random walks driven by time series of RV shown in Figure 6

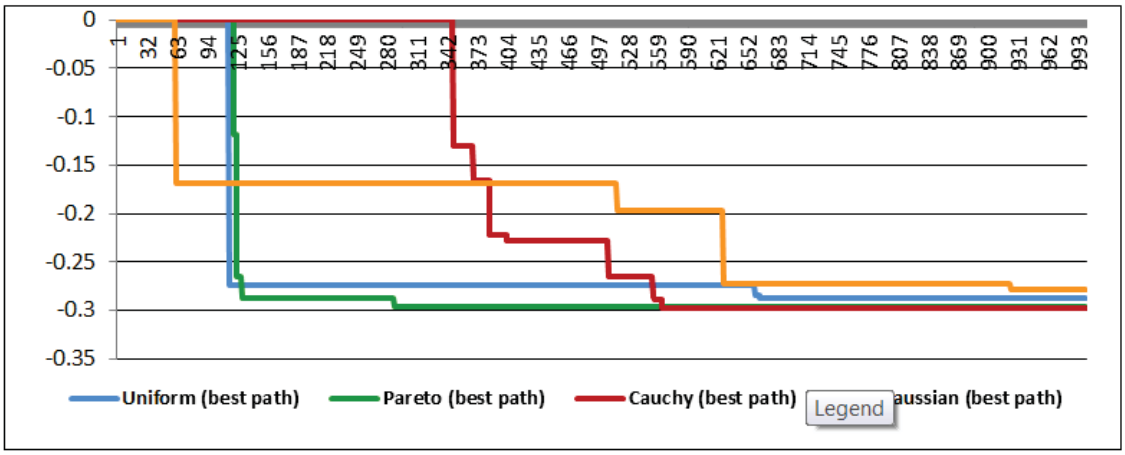

Figure 9: MBH performance along the best paths of each type of driving probability distribution

Figure 10 shows the best fitness found thus far at each step by the MBH driven by RVs drawn from each of the four probability distributions using their respective worst excursion parameters. We see that the bi-polar Pareto MBH finds a better solution than the other variants in much less time.

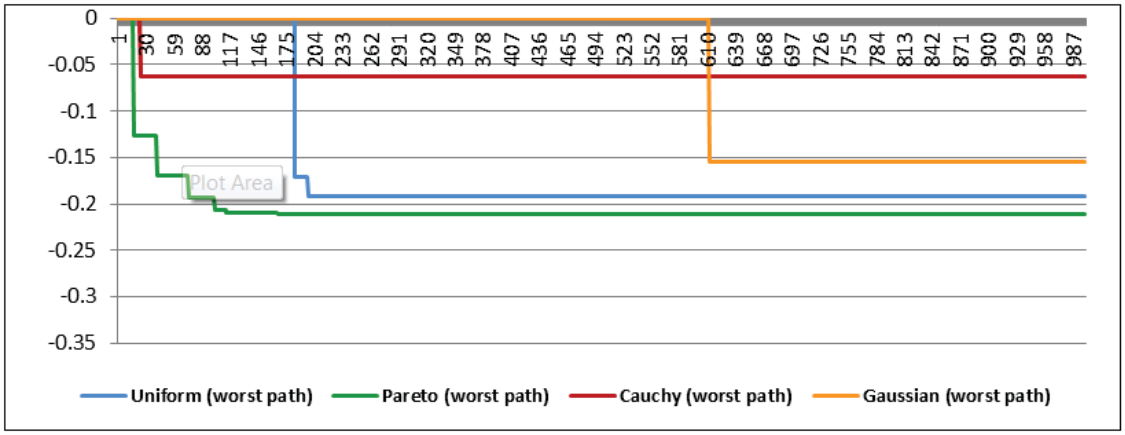

Figure 10: MBH performance along the worst paths of each type of driving probability distribution

Figure 11 shows the best fitness found thus far at each bin of ten serial steps by MBH driven by RVs drawn from each of the four probability distributions, based on their respective average performance (averaged at each bin across the distribution's 16 excursion parameters). Again we see that MBH driven by RVs drawn from bi-polar Pareto distribution finds a better solution in less time than the other variants.

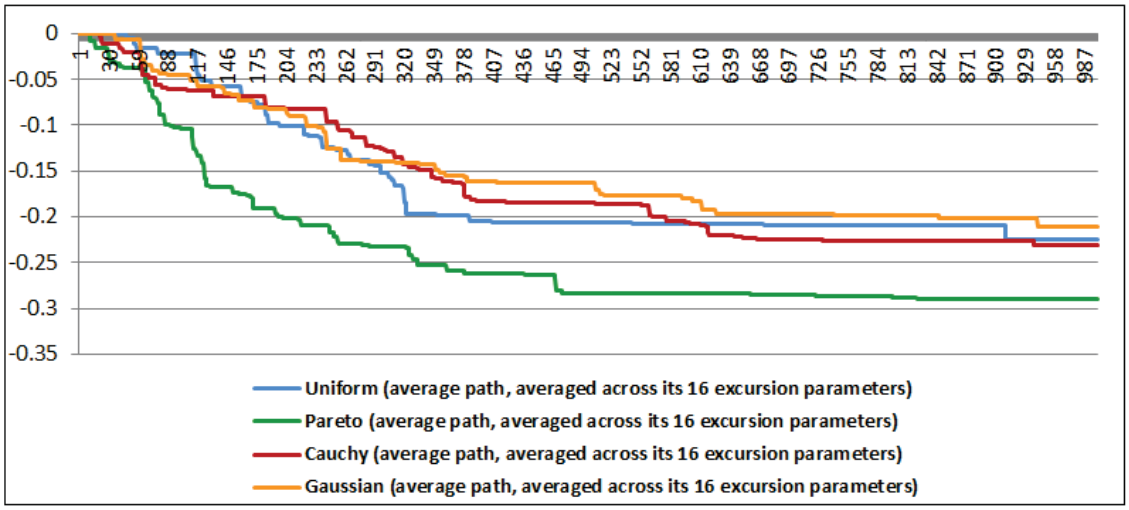

Figure 11: MBH performance averaged across the 16 paths belonging to each driving probability distribution

Figure 11 is also an indirect way of comparing the robustness of the efficiency each distribution provides the MBH given variations in that distribution's excursion parameters. We see clearly that the bi-polar Pareto distribution is the most robust in this sense because, averaged across its 16 excursion parameters, the bi-polar Pareto distribution drives the MBH to find the best solution in the least time.

A more direct way of way of comparing the robustness of the distributions with respect to variations in excursion parameters is to examine the bin-by-bin standard deviation in the best fitness found thus far, for each distribution, where the standard deviation is taken across each distribution's respective 16 excursion parameters. That time-series of standard deviations indicates how much dispersion or difference there is across the distribution's 16 excursion parameters. Less dispersion (smaller standard deviation) means greater robustness with respect to variations in excursion parameters as shown in Figure 12. The bi-polar Pareto distribution is the most robust in the sense that it's time-series of standard deviations across its excursion parameters decays the soonest and steepest.

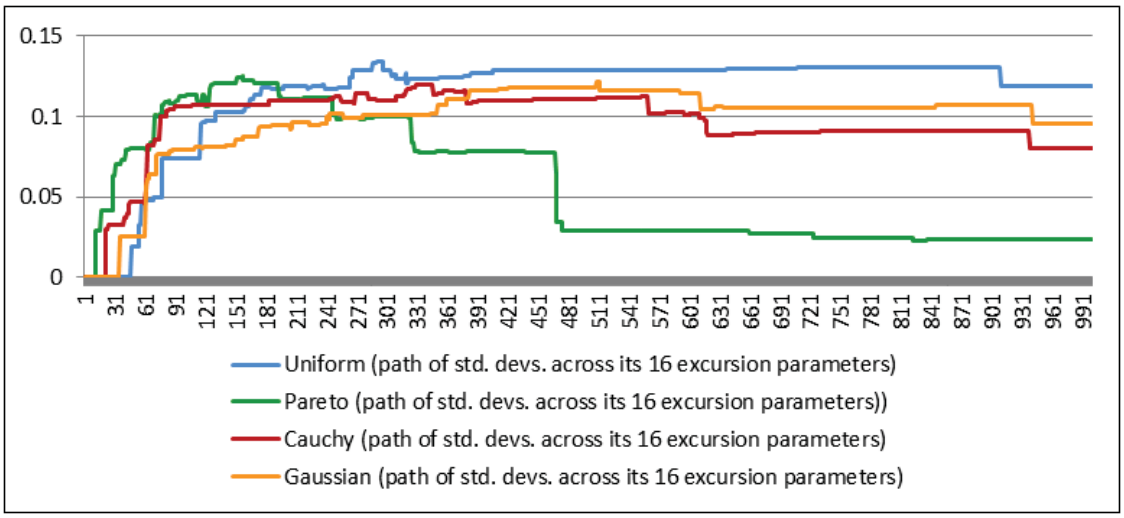

Figure 12: The standard deviation, at each 10th step, across the 16 paths belonging to each driving probability distribution

5. Theory

5.1. Introduction to Theory

The results of the experiments demonstrate that, for the optimization problem being solved, the MBHs comprised of RWs (Levy flights) driven by RVs drawn from the long-tailed Cauchy and bi-polar Pareto distributions are more efficient and robust than the MBHs comprised of RWs driven by RVs drawn from the "no-tail" uniform distribution and the "normal tail" Gaussian distribution. Further, we have demonstrated that is especially true in the case of the bi-polar Pareto distribution.

In this section we investigate the source of this superior efficiency and robustness. We were motivated to develop the theory because the strong agreement between the experimental results and the theory gives reason to believe that hypothesis is not only true for the specific problem being optimized, but is true in general, and therefore can be used to develop a sound and widely applicable engineering methodology.

Intuitively, one may expect that the source of this superior efficiency and robustness lies in how the RVs drawn from long-tailed distributions enable the MBH to search the solution space faster and more thoroughly. But we want to go beyond the intuition in order to assure that we are building on solid and generalizable principles.

We begin by identifying a metric for evaluating how fast a RW, in a given number of steps, tends to travel through a space and - in the sense of it not repeatedly "getting stuck" or "covering the same ground".

5.2. Mean Squared Displacement as a Measure of Efficiency

In this section we compare RWs via their mean squared displacement (MSD). If one RW has a greater MSD than another RW, after the same number of steps, the first RW has traveled the domain faster and more thoroughly than the second RW. In terms of our hypothesis, MSD is a way to measure efficiency of the RWs upon which MBH is built.

In statistical physics, MSD is used to describe RWs as diffusions through media. One of the benefits of adopting MSD as our metric is that it enables us to address the boundaries and internal constraints of the solution space as in-homogeneities in the medium through which the RW diffuses. The theory of diffusions in in-homogenous media, and how the in-homogeneities affect the MSD, is well developed and can be used to provide a theoretical foundation for our experimental results.

The diffusivity of RWs driven by RVs from different distributions with different excursion parameters can be evaluated, categorized, and ranked using MSD. Thus, the MSD metric is the link between this theory section and our experimental results.

For a RW driven by RVs that are independent and identically distributed (i.i.d.), analogous to assuming that the RW is traveling through a homogenous medium, drawn from a distribution that has a finite variance, $\alpha = 1$ and D is proportional to the variance of the distribution. This is defined as normal diffusivity.

It can be shown that $\langle r^2(t)\rangle \propto t$ in the case of a RW driven by RVs drawn from a Gaussian distribution, where the $\langle x\rangle$ is defined as "mean of x." The method for showing this involves solving the stochastic differential equation (SDE) that describes the RW. In the interest of focus and brevity, we refer to [20] rather than doing that here.

Likewise, it can be shown that for a RW driven by RVs that are i.i.d. but drawn from a distribution having an infinite variance, the MSD is $\langle r^2(t)\rangle \propto t^\alpha$ where $\alpha > 1$. RVs drawn from Cauchy and bi-polar Pareto distributions are i.i.d., and Cauchy and bi-polar Pareto distributions have infinite variances. Therefore, for RWs driven by RVs drawn from Cauchy and bi-polar Pareto distributions, the MSD grows non-linearly faster than the number of steps taken. This is defined as super-diffusion.

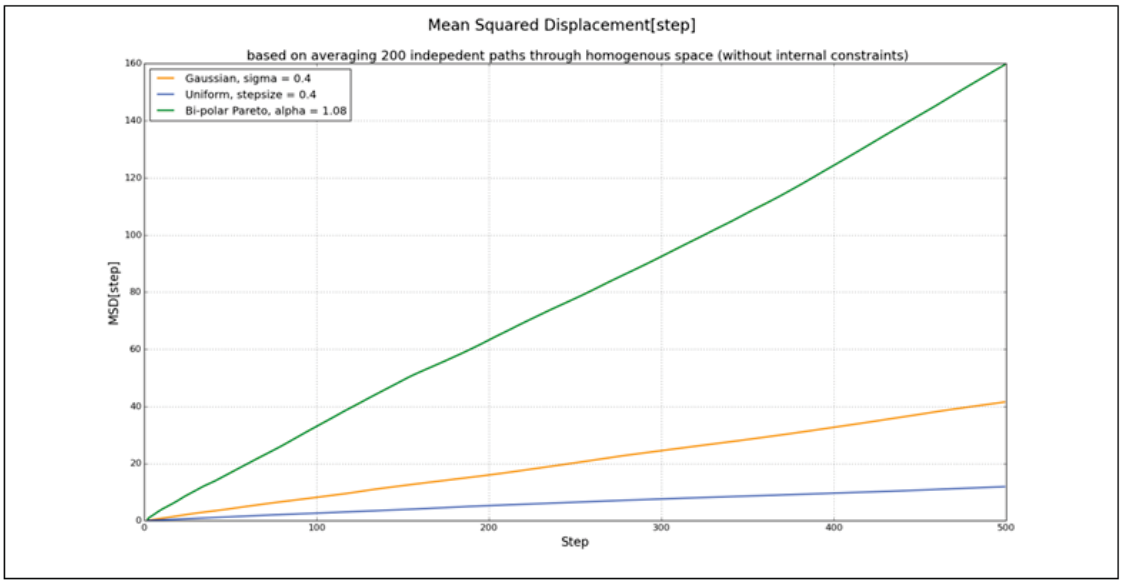

Figure 13: The Mean Squared Displacement, as a function of steps taken, for random walks driven by bi-polar Pareto, Gaussian, and uniform RVs diffusing across a homogenous (unbounded, unconstrained) space

That $\langle r^2(t)\rangle \propto t^\alpha$ for $\alpha > 1$ can be shown by solving the stochastic differential equation (SDE) that describes the RW, but the infinite variance of the distributions make the mathematics very difficult. So, again in the interest of focus and brevity, we refer to [21, 22] rather than doing that here. We do, however, illustrate this here in Figure 13, which shows the results of Monte Carlo simulations.

We see that the RWs driven by RVs drawn from the bi-polar Pareto distribution has a larger MSD than the RW driven by RVs drawn from the Gaussian distribution after each number of steps taken. We can also see that the MSD grows non-linearly as a function of the number of steps taken, although that curvature is subtle given our choice of $\alpha = 1.08$, which is within the range of α actually used in the experiment.

We can also see that the RW driven by uniformly distributed RVs has, after each number of steps taken, a smaller MSD than the RW driven by Gaussian distributed RVs after the same of steps taken. Furthermore, we can see that the MSD of the RWs driven by Gaussian and uniform RVs, respectively, are linear as a function of steps taken.

Unfortunately, results of Monte Carlo simulations of Cauchy RV-driven RWs are difficult to draw and interpret. This because RWs are cumulative sums, and cumulative sums of Cauchy distributed RVs are "wild" in the sense that they tend towards the extreme values contained the time-series, rather than averages (indeed, averages are not defined for sums of Cauchy distributed RVs - it is the "mean" in MSD that is causing the problem). As a result, it is difficult to draw the MSD of a RW driven by Cauchy distributed RVs on the same graph as the MSDs of RWs driven by uniform-, Gaussian- and bi-polar Pareto distributed RVs, and it is very difficult to say whether the Cauchy RV-driven MSD simulated is in any

way typical. Therefore we show the Monte Carlo simulated MSDs of the RWs driven by uniform-, Gaussian- and bi-polar Pareto distributed RVs alongside one another, but without the simulated MSD of Cauchy RV-driven RWs.

The MSDs of RWs driven by uniform distributed RVs are usually not discussed in the literature of diffusions, but because they are the RW upon which MBH has historically been built, we address them here. RVs drawn from uniform distributions are i.i.d., and uniform probability distributions have finite variances. Therefore the MSDs of RWs driven by uniform distributed RVs are linear as a function of steps taken, and therefore RWs driven by uniform distributed RVs are normal diffusions rather than sub- or super-diffusions. But, as can be shown analytically by noting that the variance of any uniform distribution over the internal $[-d, d]$ is much smaller than d, the MSD of a uniform RV-driven RW diffuses much more slowly than a Gaussian RV-driven RW when the uniform distribution has the same excursion parameter as the Gaussian. In other words, the diffusivity of RWs driven by uniform distributed RVs is normal, but very slow. As we will see below, it has a normal diffusivity that can easily be turned into sub-diffusivity by the effects of boundaries around, and constraints within, the space through which the RW is traveling. We will see that this contributes to why Cauchy and bi-polar Pareto RV-driven MBHs are not only more efficient than uniform RV-driven MBHs, they are also more robust with respect to the effects of boundaries and constraints.

Note that we measure and compare normal diffusivity, sub-diffusivity, and super-diffusivity using motions of different RWs over the same space. This is important when comparing diffusions through spaces that are bounded and have internal constraints (analogously, diffusions through in-homogenous media), to spaces free of boundaries and internal constraints (analogously, diffusions through homogenous media). The ability to make such comparisons is required for our evaluation of robustness of different probability distributions as generators of RVs that drive the RWs upon which MBH is based. With that observation, we are ready to use MSD and the notion of diffusions through in-homogenous and homogenous media to address the effects of the boundaries of, and internal constraints within, the solution space.

Before proceeding to discuss diffusions through in-homogenous spaces in order to model the effects of boundaries and internal constraints, we want to tie the current paper to prior work in which RVs drawn from long-tailed distributions were used to accelerate Simulated Annealing (SA) optimizations on widely bounded multi-dimensional spaces that were free from internal constraints (analogously, diffusions through homogenous media).

Investigations into improving stochastic optimization algorithms using RVs drawn from long-tailed probability distributions go back at least as far the 1987 contributions of Szu and Hartley [23] and their use of Cauchy distributions to accelerate SA optimizations. Their work addressed the optimization of objective functions having multiple local minima on multi-dimensional spaces that were widely bounded and free of internal constraints (for all practical purposes, free space). Using our terminology, Szu and Hartley assumed that the solution spaces were homogenous media. In 1996, Tsallis and Stariolo [24] extended the work of Szu and Hartley by developing a Generalized Simulated Annealing (GSA) algorithm based

on non-extensive statistical mechanics (a field pioneered by Tsallis). Tsallis and Stariolo reported that their GSA algorithm demonstrated itself to be even faster than the so-called "Fast Simulated Annealing" (a.k.a. "Cauchy Machine") developed by Szu and Hartley. Like the work of Szu and Hartley, the work of Tsallis and Stariolo assumed, and was numerically tested on, objective functions having multiple local minima on multi-dimensional spaces that were, for all practical purposes, free space (homogenous media). In 2004, Junior, et. al. [25] investigated the performance and parameterizations of the GSA algorithm in the context of objective functions having multiple local minima on multi-dimensional spaces that were, again, essentially free space (homogenous media). In addition, in all three investigations, while the theory was developed for multi-dimensional spaces having any number of dimensions, the numerical experiments involved objective functions on spaces that were at most two-dimensional.

In the current paper, we focus on multi-dimensional spaces that are both tightly bounded and populated with plentiful internal constraints of all sizes - simply because these are essential properties of solution spaces in which MBH is used to optimize low-thrust trajectory designs for space missions. In addition, the numerical experiments in the current paper are optimizations of actual design problems, and therefore involve objective functions on 500+ dimensional spaces. In each dimension, internal constraints have a large effect on the behavior of MBH. Therefore, we now turn to the topic of diffusions through in-homogenous media, which is our model for the effects of the tight boundaries and plentiful internal constraints on the speed and robustness of MBH.

5.3. Diffusion through In-homogenous Media Explains Robustness

Before we develop the theory of robustness and how it relates to the effects of the boundaries of, and the internal constraints within, the solution space, we start with what intuition suggests. It is intuitive that the boundaries and internal constraints of the solution space make it - as the medium through which the RWs are traveling - in-homogenous. Intuition suggests that the in-homogeneities would "slow" the RW. What they actually do is more complicated and interesting, as explained below. In order to go beyond the intuition and explain this more deeply, we need a few more concepts from probability theory, statistical physics, and recent financial math: serially correlated, serially non-correlated (statistically independent), and serially anti-correlated (negatively correlated) RVs.

Most people think about randomness in terms of serially non-correlated RVs. With serially non-correlated RVs, one random outcome in a time series is unaffected by the prior random outcomes, with the most common example being flips of fair coins. Computer random number generators are carefully designed to generate RVs that are serially non-correlated (or if they are serially correlated, their serial correlations are inconsequentially - and hopefully, immeasurably - small).

With serially correlated RVs, one random outcome tends to condition the probabilities of the next (or the next set of) random outcome(s). Saying that RVs are serially correlated RVs, in contrasted to serially anti-correlated, implies that one random outcome tends to condition

the probabilities of a second random outcome in favor of the same direction as that of the first outcome. Such serial correlations are often referred to as positive serial correlations, trends, or drifts. Common examples are trends caused by the herding behaviors of cattle, lemmings, drivers on the Capital Beltway, or stock market traders, or by avalanches of sand, snow or mud.

With serially anti-correlated RVs (a.k.a. negatively correlated RVs), one random outcome tends to condition the probabilities of the next (or the next set of) random outcome(s) in favor of the direction opposite to that of the first outcome. These sometimes referred to as reversion behaviors. Examples of reversion behaviors can often be found in the dynamics of supply chains, traffic jams, nearly oscillatory group behaviors of insects, and in short-term cyclic patterns that can be found in highly traded broad stock market indices.

Serial correlation refers to statistical dependencies that unfold across time, which are sometimes referred to a as a "long term memory" effects. Correlations across members of populations, such as cattle, lemmings or stock traders, are referred to as spatial correlations. This comes from statistical physics where the members of populations are molecules or particles moving through space. Now that it is clear that we are referring to serial correlations, we will simply use the shorter expression, "correlation." It is very difficult to show, analytically, that RWs driven by anti-correlated RVs have smaller MSDs than RWs driven by non-correlated RVs. Showing this analytically involves specialized topics in stochastic processes [26]. Here we illustrate it using Monte Carlo simulations. It can also be shown that RWs driven by serially correlated RVs have larger mean path lengths than RWs driven by serially non-correlated RVs, but we do not make use of that in this paper, so we do not illustrate it here.

Before proceeding, some nomenclature needs to be explained. We want to say that RWs driven by anti-correlated RVs have smaller MSDs than RWs driven by non-correlated RVs even when the RVs come from the same distribution, but saying this is mathematical nonsense. In order for this statement to make sense, we are required to point out that the RVs were i.i.d. when they were drawn from their probability distribution of origin. We need to think of the correlation process as taking place after the originating distribution generates non-correlated RVs. Therefore, we are really discussing how the MSDs are made smaller by the anti-correlations induced by the effects of the boundaries surrounding, and internal constraints within, the solution space.

In fact, the process that then correlates them alters their distribution, if only slightly. Nonetheless, it is useful to note that the RVs that were from that distribution (for example, Gaussian) before becoming correlated. That is because those correlated RVs often share properties among themselves, and with non-correlated RVs from the same distribution, and those shared properties may be so strong that keeping track of their distribution of origin is important. This is so useful that such RVs are sometimes referred to by the technical misnomer "correlated Gaussian," etc.

We now use the property that RWs driven by anti-correlated RVs have smaller MSD than RWs driven by non-correlated RVs, and the fact that correlated (in this case anti-correlated) RVs retain many of the properties of their distributions of origin, to explain robustness.

The boundaries and internal constraints induce anti-correlations in the RVs driving the RWs. An easy way to think about this is to imagine the RW "stuck" at some boundary in some dimension. The only way it can get "unstuck" in that dimension is to draw a sufficiently large RV for that dimension that sends it in the opposite direction. Otherwise it must ignore the RV it has drawn and remain where it is.

The RVs in that dimension have, by "pinning" the RW against the boundary in that dimension, have conditioned the probable acceptance of the next RVs in that dimension - they have assured that only sufficiently large RVs in the opposite direction will be acceptable. This induces anti-correlations. The anti-correlations, in turn, decrease the mean path length of the RW, meaning that the RW moves through the space more slowly and less thoroughly. In some literature, RWs that are driven by RVs that are anti-correlated are said to be "anti-persistent."

Depending upon the extent of the anti-correlations, by reducing the MSD, they can turn a non-correlated normally-diffusive RW into a correlated sub-diffusive RW, they can make a non-correlated sub-diffusive RW much more sub-diffusive, and they can even turn a non-correlated super-diffusive RW (Levy flight) into a normally- or sub-diffusive RW.

However, the RWs (Levy flights) driven by Cauchy and bi-polar Pareto RVs (especially bi-polar Pareto RVs) are so super-diffusive that they remain super-diffusive despite the boundaries and internal constraints associated with the optimization problem being solved (and, we believe, problems like it). This explains how the theory of diffusion through in-homogenous media explains the relative robustness of MBHs built on RWs driven by RVs drawn from different probability distributions.

Again, for reasons explained above, the results of Monte Carlo simulations of Cauchy RV-driven RWs are difficult to draw and interpret. Therefore we show the results of Monte Carlo simulations of RWs driven by uniform-, Gaussian- and bi-polar Pareto distributed RVs alongside one another, as they are affected by the boundaries and internal constraints of the solution space, but without showing a simulated MSD of a RW driven by Cauchy distributed RVs.

Figure 14 illustrates the effects of the boundaries and internal constraints on MSDs using Monte Carlo simulation. We see that the MSD of the RW driven by Gaussian RVs is now significantly diminished, no longer much larger than the MSD of the RW driven by uniform RVs, and that it is curved. The curvature indicates that the MSD of the RW driven by Gaussian RVs is now longer linear (proportional) with respect to the number of steps taken. This non-linearity indicates that it has been changed by the effects of the boundaries and internal constraints (the in-homogeneities) in ways more complicated and interesting than having simply been "slowed."

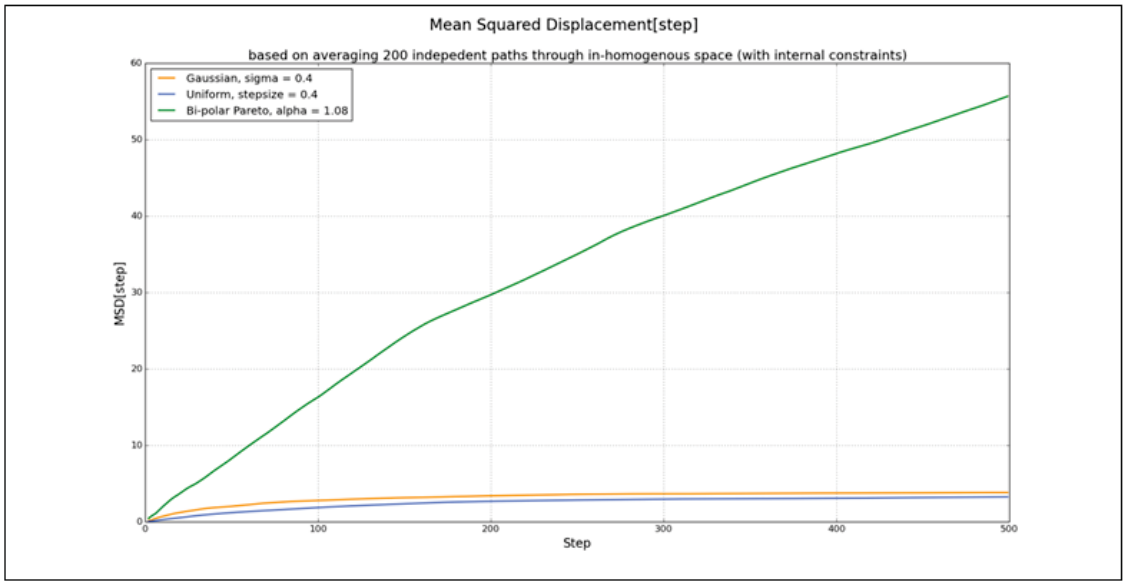

Figure 14: The Mean Squared Displacement, as a function of steps taken, for random walks driven by bi-polar Pareto, Gaussian, and uniform RVs diffusing across an in-homogenous (bounded and constrained) space

When we look at the MSD of the RW driven by bi-polar Pareto RVs, we see that it is diminished, but not as much as the mean square displacements of the uniform RV-driven and the Gaussian RV-driven RWs. The mean square displacement of the Pareto RV-driven RW is larger than the others by more than it was in the case of the unbounded space free of internal constraints (i.e. the diffusions through the homogenous medium).

Figure 15 shows the respective uniform RV-driven and the bi-polar Pareto RV-driven random walks in the unbounded space free of internal constraints (corresponding the MSDs shown in Figure 13), and in the bounded space having internal constraints(corresponding the MSDs shown in Figure 14). We see that the "mildness" of the uniform RV-driven random walk does not enable it to "jump over" the internal constraints, so it remains "stuck" in a sub-space, but the "wildness" of the long-tailed bi-polar Pareto RV-driven random walk often "jumps over" the internal constraints and is thus able to explore the entire bounded space.

Because all the mean square displacements are decreased in Figure 14, we infer that the boundary limits and internal constraints induce anti-correlations (serial negative correlations).

One can also verify the presence of serial negative correlations by taking auto-correlations functions of the step-by-step differences in the positions in RWs. An auto-correlation function is a set of correlation functions between a time series and lagged versions of itself. The auto-correlation function of time series of RVs that are serially uncorrelated, because their values at any point in time are unaffected by their values at any other point in time, is one at lag zero, and zero at every other lag. The auto-correlation of any time series of RVs that is serially anti-correlated, necessarily and uniquely has negative values at some lags. When we

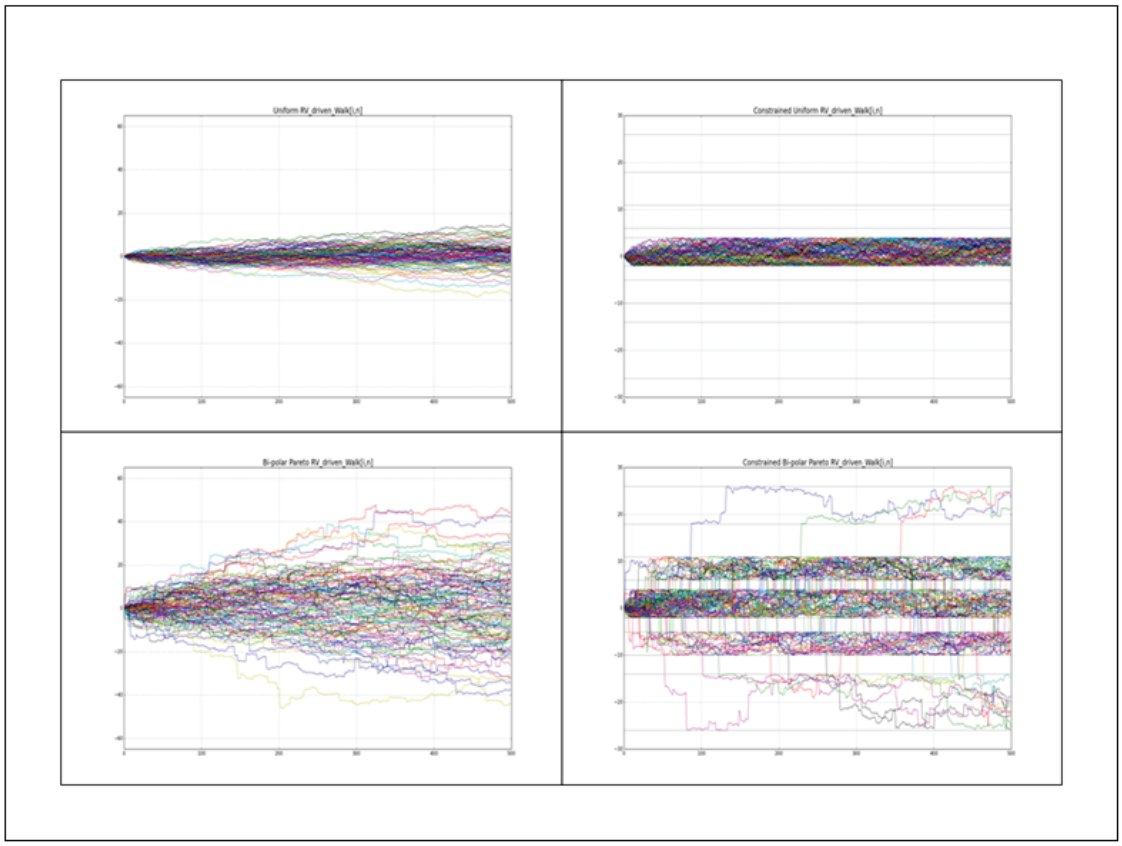

Figure 15: The random walks driven by uniform (top) and bi-polar Pareto (bottom) RVs diffusing across a homogenous space (left) and an in-homogenous space (right)

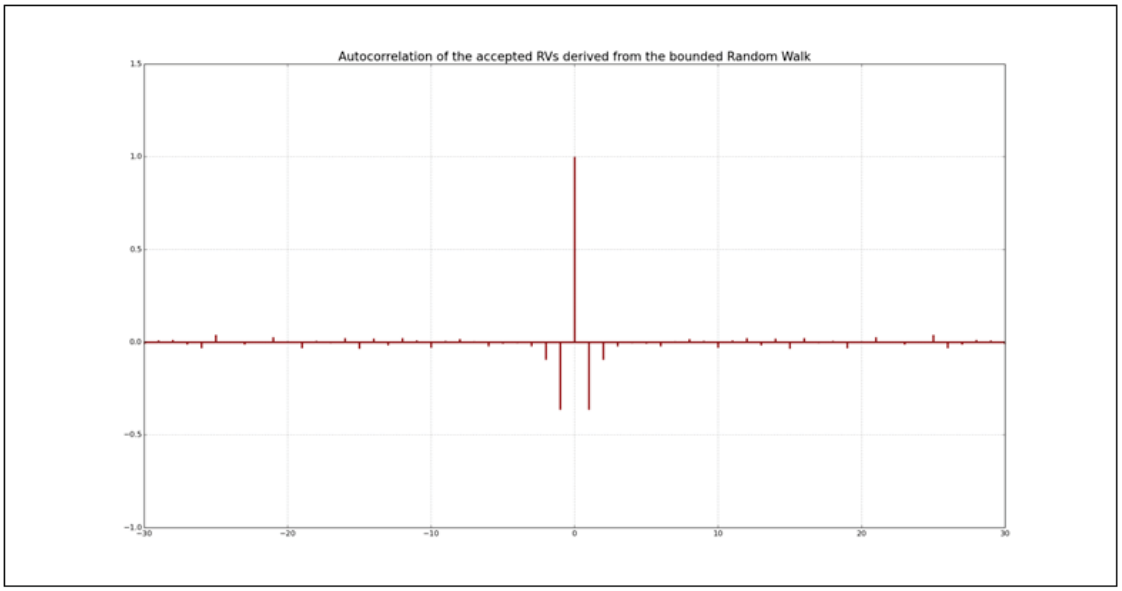

Figure 16: Autocorrelation function of the uniform RVs that were "accepted" and permitted to drive the random walk, despite the boundaries and constraints as shown in the right side of Figure 15

take the auto-correlations of the step-by-step differences in the positions in RWs through a space that is free of boundaries and has no internal constraints, we see one at lag zero, and zero at every other lag. When we take the auto-correlations of the step-by-step differences in the position in RWs through a space that has boundaries and internal constraints, however, we see negative values at some lags.

In the autocorrelation function shown in Figure 16, we see clear signs of serial anti-correlation in the Gaussian RV-driven RW, caused by the boundaries and internal constraints of the solution space. We do see the expected value of one at the zeroth lag (of course, every RV is completely and positively correlated with itself), but we also see strong negative values at lags 1 and -1. These negative values at lag 1 and -1 mean that often the next RVs moves the walk in the opposite direction that the prior RV had moved it (because it must have to be in order for the RW to get "unstuck" and start moving again).

In the autocorrelation function shown in Figure 17, we see the expected value of one at the zeroth lag, and almost zero at every other lag. (The only reason that the values are not exactly zero at every lag is that the random number generator is imperfect. Indeed, taking auto-correlations is an important test of imperfections in random number generators.) The same can be shown for the other distributions used in this paper.

This confirms that the boundaries and internal constraints are inducing negative serial correlations. Thus it is the negative (a.k.a. "anti-") serial correlations that slow the diffusion. The reason that the MSD of the Pareto RV-driven RW is slowed the least of all, is because the RWs driven by bi-polar Pareto RVs are so super-diffusive in unconstrained space (homogenous media) that their MSD is not significantly affected by the anti-correlations induced by

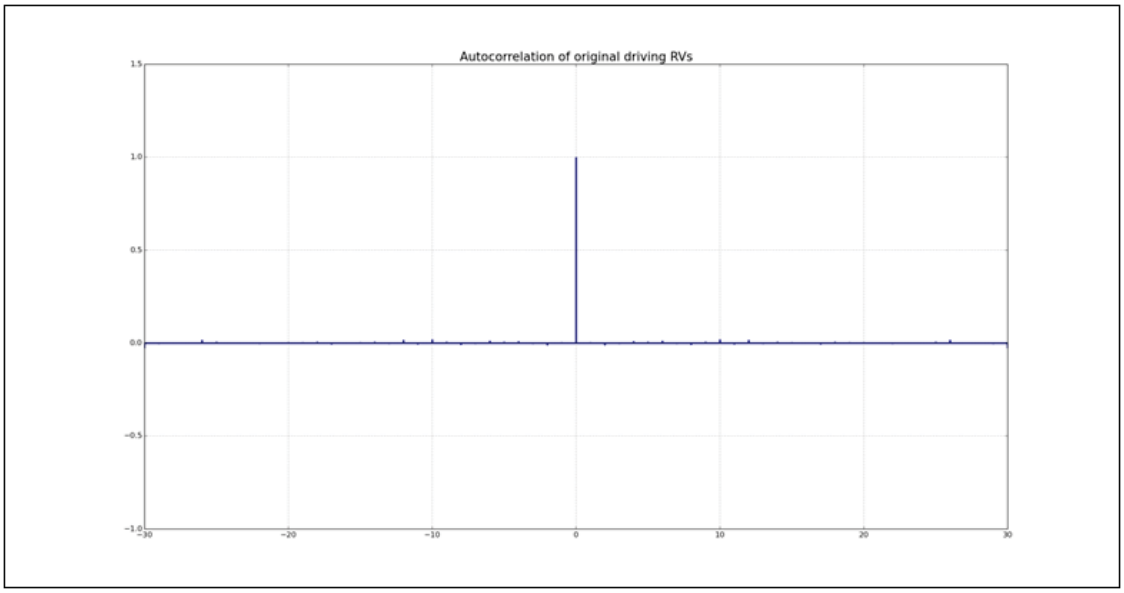

Figure 17: Autocorrelation function of the uniform RVs, all of which were "accepted" when the space was unbounded and unconstrained as shown in the left side of Figure 15

boundaries and internal constraints (in-homogeneities in the media). This, in turn, explains the superior robustness of building MBHs based on bi-polar Pareto RV-driven RWs, which is evident in the experimental results.

5.4. Summary of the Theory

We have now supported the strong agreement between our hypothesis - that MBH based on RWs (Lévy flights) driven by RVs drawn from Cauchy and bi-polar Pareto distributions are more efficient and robust than MBH based on RWs driven by RVs drawn uniform distribution or Gaussian distributions - on the theory of diffusions of through homogenous and in-homogenous media. While, in the interest of brevity and focus, we have not proved all of the mathematical ideas we have used, we have referred to sources where these proofs can be found. Instead, we have provided illustrations from Monte Carlo simulations.

Overall, this theory section explains our belief that our experimental results are not problem specific and should be applicable to other problems of the same class (*i.e.* other trajectory optimization problems, and perhaps also problems in other fields).

6. Future Work

Future work on this project will be focused on two areas: more thorough benchmarking and improved acquisition of the initial feasible solution. The authors recognize that the numerical results presented in this work are based on a limited set of test cases. This was due to the high computational cost of testing and the limited time available. A more thorough investigation of this topic should include, at the very least, more runs of the benchmark problems presented in this work. In addition, it is possible that the best tunings for the various distributions are

dependent on the problem chosen. Additional benchmark problems, other than the Earth to Uranus low-thrust example, could be constructed in EMTG and tested using the methods described in this work. Alternatively, or in addition to those tests, the variants of MBH described in this work could be applied to other problems outside of the EMTG framework and even problems outside of the Astrodynamics field.

Second, the tests are all dependent on finding an initial feasible point. The classical MBH does this by choosing uniform random points from the solution space and running the NLP solver. This process is repeated until a feasible point is found. While the classical MBH is effective at finding initial feasible points for most problems, it struggles on extremely complex problems (more complex than the example presented in this work). Further work will examine alternate strategies to find the initial feasible point.

7. Conclusion

We have investigated the extent to which basing MBH on RWs driven by RVs drawn from long-tailed probability distributions - rather than RVs drawn from the uniform distribution that has been used classically - improves the performance of the optimizer. The improvements we sought are of two kinds: MBH efficiency, which we define as finding better solutions in less time; and MBH robustness, which we define as MBH efficiency that is not diminished by reasonable variations in the excursion parameters of the probability distributions from which MBH's driving RVs are drawn. We consider robustness to be a very important indicator that the improved efficiency is neither highly problem-dependent nor highly dependent upon the tuning of the MBH.

Our hypothesis was that long-tailed probability distributions would provide both kinds of improvements, based on our theoretical model that MBH involves diffusions through multi-dimension spaces that are extremely in-homogenous in nature because of their tight boundaries and plentiful internal constraints of all shapes and sizes. Our model suggested that using RVs drawn from long-tailed probability distributions would result in super-diffusive search behaviors that would enable the MBH to cover the space faster and more thoroughly, without being significantly impeded by the in-homogeneities, compared to using RVs drawn from uniform (or other "relatively tame") probability distributions.

We tested our hypothesis with numerical experiments that were, in fact, an actual low-thrust trajectory optimization problem with which one of us had extensive experience solving with classical MBH. This actual problem has solution space that is 500+ dimensions; tight boundaries; and has internal constraints that are very challenging because of their sizes, shapes, clustering, and frequency.

The results of the experiment confirm our hypothesis that RWs driven by RVs drawn from long-tailed probability distributions (particularly Cauchy and bi-polar Pareto distributions) provide significantly more efficient and robust MBH performance than RWs driven by RVs from drawn uniform distributions or Gaussian distributions.

Because this experimental confirmation of our hypothesis is highly consistent with our theoretical model, we believe that our results are very likely to be general rather than being problem specific.

We find that of the two long-tailed distributions, bi-polar Pareto (in contrast to Cauchy), is the more robust to variations in excursion parameters. Therefore, we are inclined to recommend the bipolar Pareto as the distribution that best improves the performance of MBH over the performance achieved by the classical use of the uniform distribution.

The results found in this paper are directly applicable to improving the performance of EMTG on low-thrust trajectory optimization problems and have been adopted as the default settings for EMTG's MBH+SNOPT optimizer. We expect that this work will have broader applicability beyond low-thrust trajectory optimization and even beyond Astrodynamics in general.

Acknowledgments

The authors would like to acknowledge earlier contributions to EMTG by Donald Ellison, Bruce Conway, Alexander Ghosh, Matt Vavrina, Steve Cooley, Steve Hughes, Frank Vaughn, and Max Schadegg. The National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) independent research and development (IRAD) program funded the development of EMTG.

8. References

[1] Yam, C., di Lorenzo, D., and Izzo, D. "Low-Thrust Trajectory Design as a Constrained Global Optimization Problem." "Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering," Vol. 225, pp. 1243–1251. 2011.

[2] Englander, J. A., Conway, B. A., and Williams, T. "Automated Interplanetary Mission Planning." "AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, MN," August 2012.

[3] Englander, J. A. Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions. Ph.D. thesis, University of Illinois at Urbana-Champaign, April 2013.

[4] Ellison, D. H., Englander, J. A., and Conway, B. A. "Robust Global Optimzation of Low-Thrust, Multiple-Flyby Trajectories." "AAS/AIAA Astrodynamics Specialist Conference, Hilton Head, SC," August 2013.

[5] Ellison, D. H., Englander, J. A., Ozimek, M. T., and Conway, B. A. "Analytical Partial Derivative Calculation of the Sims-Flanagan Transcription Match Point Constraints." "AAS/AIAA Space-Flight Mechanics Meeting, Santa Fe, NM," January 2014.

[6] Englander, J. A., Ellison, D. H., and Conway, B. A. "Global Optimization of Low-Thrust, Multiple-Flyby Trajectories at Medium and Medium-High Fidelity." "AAS/AIAA Space-Flight Mechanics Meeting, Santa Fe, NM," January 2014.

[7] Englander, J., Conway, B., and Williams, T. "Automated Mission Planning via Evolutionary Algorithms." Journal of Guidance, Control, and Dynamics, Vol. 35, No. 6, pp. 1878–1887, 2012.

[8] Sims, J. A. and Flanagan, S. N. "Preliminary Design of Low-Thrust Interplanetary Missions." "AAS/AIAA Astrodynamics Specialist Conference," Girdwood, Alaska, August 1999.

[9] Vavrina, M. and Howell, K. "Global Low Thrust Trajectory Optimization through Hybridization of a Genetic Algorithm and a Direct Method." "AIAA/AAS Astrodynamics Specialist Conference and Exhibit," Honolulu, Hawaii, August 18-21 2008.

[10] Sims, J., Finlayson, P., Rinderle, E., Vavrina, M., and Kowalkowski, T. "Implementation of a low-thrust trajectory optimization algorithm for preliminary design." "AIAA/AAS Astrodynamics Specialist Conference," August 2006.

[11] "PaGMO (Parallel Global Multiobjective Optimizer).", March 2012. `http://pagmo.sourceforge.net/pagmo/index.html`.

[12] Prussing, J. and Conway, B. Orbital Mechanics. Oxford University Press, New York, 1993.

[13] Conway, B. A. "An Improved Algorithm due to Laguerre for the Solution of Kepler's Equation." Celestial Mechanics, Vol. 39, No. 2, pp. 199–211, 1986.

[14] "SPICE Ephemeris.", August 2013. `http://naif.jpl.nasa.gov/naif/`.

[15] Gill, P. E., Murray, W., and Saunders, M. A. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization." SIAM Rev., Vol. 47, No. 1, pp. 99–131, 2005. ISSN 0036-1445. doi:http://dx.doi.org/10.1137/S0036144504446096.

[16] Leary, R. "Global optimization on funneling landscapes." Journal of Global Optimization, Vol. 18, No. 4, pp. 367–383, December 2000. ISSN 0925-5001. doi:{10.1023/A:1026500301312}.

[17] Vasile, M., Minisci, E., and Locatelli, M. "Analysis of Some Global Optimization Algorithms for Space Trajectory Design." Journal of Spacecraft and Rockets, Vol. 47, No. 2, pp. 334–344, March-April 2010. ISSN 0022-4650. doi:{10.2514/1.45742}.

[18] Addis, B., Cassioli, A., Locatelli, M., and Schoen, F. "A global optimization method for the design of space trajectories." Computational Optimization and Applications, Vol. 48, No. 3, pp. 635–652, April 2011. ISSN 0926-6003. doi:{10.1007/s10589-009-9261-6}.

[19] Mandelbrot, B. and Hudson, R. The Misbehavior of Markets: A Fractal View of Financial Turbulence. Basic Books, 2004.

[20] Chechkin, A. "A Short Introduction to the Theory of Levy Flights.", 2009. `http://www.maths.qmul.ac.uk/~klages/bee_wshop/bbees_chechkin.pdf`.

[21] Baschnagel, J. and Paul, W. Stochastic Processes from Physics to Finance. Springer, 1999.

[22] Klafter, J., Schlesinger, M., and Zumofen, G. "Beyond Brownian Motion." Physics Today, pp. 33–39, February 1996.

[23] Szu, H. and Hartley, R. "Fast Simulated Annealing." Physics Letters A, Vol. 122, No. 3, p. 157, June 1987.

[24] Tsallis, C. and Stariolo, D. "Generalized Simulated Annealing." Physica A, Vol. 233, pp. 395–406, 1996.

[25] Junior, A., Silva, R., Mundi, K., and Dardenne, L. "Performance and parameterization of the algorithm Simplified Generalized Simulated Annealing." Genetics and Molecular Biology, Vol. 27, No. 4, pp. 616–622, 2004.

[26] ben Avraham, D. and Havlin, S. Diffusion and Reactions in Fractals and Disordered Systems. Cambridge University Press, 2004.