Student: Ivo Fagundes David de Oliveira
Tutor: Ricardo Hiroshi Caldeira Takahashi

ivodavid@gmail.com
taka@mat.ufmg.br

Optimal Black-Box Sequential Searching

(Pesquisa Sequencial Ótima em Funções Caixa-Preta)

Master's Degree Dissertation defended at:
Federal University of Minas Gerais
Mathematics Department
June 14, 2013

Acknowledgments

To the one and true God creator of heaven and earth, who gracefully blessed me with life, knowledge, faith, hope and love along this pilgrimage, be the glory forever.

To those who are precious to me be my love and service (through this work and through my acts and words) until I cross the river.

To those who may meet these words be the blessing of He who blessed me as well.

Summary. This dissertation constructs optimal root-searching and maximum-searching algorithms in a statistical sense and compares the statistically optimal strategies to the already known mini-maximal strategies. In order to construct the so called statistical method, new results in the field of probability, capable of determining the probability of $f(x) = y$ over a pre-determined set of functions, are presented.

Sumário (Português). Esta dissertação constrói algoritmos de busca de raiz e de busca de máximos, ótimos em um sentido estatístico, e compara os métodos estatisticamente ótimos com as já conhecidas estratégias mini-maximais. A fim de construir o chamado método estatístico, novos resultados na área de probabilidade, capazes de determinar a probabilidade de $f(x) = y$ sobre um conjunto pré-determinado de funções, são apresentados.

Contents

1

Preface

"How many roads must a man walk down before you can call him a man?" Bob Dylan

I am very fond to puzzle and enigma solving ever since my young ages. I came to know a mathematical enigma at the beginning of my undergraduate studies that was presented as:

Enigma. *A boy is thinking of an integer number between 1 and 100. Given that the the boy will only answer your questions with a "yes" or "no", what is the minimum amount of questions you must ask the boy until you find out what number he is thinking?*

To this enigma me and my friends quickly found the expected answer written in the answer section of the book. The classical answer and hours of conversation arose new ideas into how to obtain better strategies to find the number the boy was thinking. One option for example, supposing that the boy isn't obliged to answer if there isn't a well defined answer, is to divide the group of possible numbers into three groups and ask questions that the answer is "yes" for one group, "no" for another and for the last the answer is undefined. This way you can subdivide and eliminate "2/3" of the possible group at each question. An example of such a question for the first 100 numbers is Answer 2 (For the classical answer read footnote [1]):

Answer 2. *Let R be the remainder of your number divided by 3. Is $(R-1)^{-1}$ greater than zero?*

R can assume 3 values: 0, 1 or 2.
for $R = 0$ (multiples of 3) the answer is "no" for:

$$\frac{1}{(R-1)} = \frac{1}{-1} = -1 < 0$$

for $R = 2$ the answer is "yes" for:

[1] **Answer 1.** *Asking questions that subdivide the group of possible numbers by two, such as „Is your number greater than 50?" with 7 questions you will find the desired number.*

$$\frac{1}{(R-1)} = \frac{1}{1} = 1 > 0$$

for $R = 1$ there is no answer, for:

$$\frac{1}{(R-1)} = \frac{1}{0} = \pm\infty$$

With this strategy we can obtain a faster convergence to the number thought by the boy. This methodology reduces by 2 the number of questions necessary to obtain the answer given by classical strategy.

Another possibility is to suppose the contrary, that is, if the boy (not omniscient) was obliged to answer "yes" or "no" to any question made to him. In this case it would be possible to trick the boy into forcing any number, say n for example, to be the only coherent number to the sequence of answers given by the boy, more precisely one answer to the following question:

Answer 3. *I am thinking in a group of numbers, is your number contained in this group?*

If the answer of the boy is "yes", the strategy is obvious, just say your group is simply the number n. Otherwise if the boy answers "no" just say the group you where thinking was a group that contained all numbers between 1 and 100 except n. Either case the chosen number n will be the only coherent number to the boy's answer, therefore forcing the result with only one question no matter how big the initial group is.

In this work, despite the simplicity in which the above mathematical enigma was presented and solved, I intend to show, in the following work, innovative methods to solve sequential root-searching problems and posteriorly maximum-searching problems and demonstrate rigorously their optimality. Along the way, new results in the field of statistics and probability shall be constructed and a rigorous theory capable of evaluating efficiency of sequential searching algorithms will emerge. At last I hope that the pleasure involved in enigma and puzzle solving, illustrated in this introductory section, may follow the reader along the way during the meditation over the various challenges involved in sequential black-box function searching.

2

Black-Box Sequential Searching

"The lot is cast into the lap, but its every decision is from the lord" Pv 16:33 NIV

Introduction

Def. A function f is called a black-box function if the access to an explicit mathematical expression of function f isn't possible, but the evaluation of function f at point x is accessible for every x in the domain of f.

In some cases when f's mathematical form is known, but it's mathematical manipulation is rather complicated, f is also treated as a black-box function for simplicity. Functions of this nature often appear as the result of a numerical simulation or of a physical experiment. The following functions are examples that can be treated as black-box functions:

1. Function $R(t)$ measures the electrical resistance(in Ohm's) of a composite exposed for t minutes to a given acid. A laboratory experiment was prepared to expose the composite to the acid and measure R of this composite.
2. A weather predicting computer algorithm gives an estimate of the temperature of Gotham City for any day of the following month. The temperature $T(d)$ is a black-box function defined over the variable day $d \in \{1, ..., 30\}$.
3. $f(x,t) = \ln(x) + \sin(t)x^{x^{\sqrt{t}}}$

Although explicit expressions for the above functions aren't known (or are up to some extent complicated), commonly some informations of the family $\mathbb{F}$ of function $f \in \mathbb{F}$ can be inferred by necessary or reasonable conditions. In 1, for example, R can be inferred as to be continuous and in 2, T may be assumed to belong to the range $5^o < T < 35^o$.

Applications often request solutions to problems defined over black-box functions. Two classical examples of problems defined over black-box functions will be studied, the root-searching problem and the maximum-searching problem. Classical procedures to solve these problems construct a sequence of points $x_1, x_2, ..., x_n, ...$ that converges to the solution and thus the reason for the title "Black-Box Sequential Searching". In fact, classical solutions to problems defined over functions $f : \Re \to \Re$ more than just converge to the solution, but often, after n steps, furnish an interval where the solution is located.

Various algorithms exist to solve both root-searching and maximum-searching problems, some of which will be explained in the respective chapters of root-searching

or maximum-searching. The terminology "strategy" will be employed here with the same meaning of "algorithm". If two strategies are allowed to evaluate function f at n points and one obtains the smallest region (interval when $f : \Re \to \Re$) with the location of the solution, then it is understood to be better than the other.

This vague notion of what makes one strategy better than another one may be, and has been, better understood by two distinct definitions:

Def. Worst Case Optimality. Given a problem P defined over a black-box function $f : \Re \to \Re \in \mathbb{F} = \{\text{A given family of functions}\}$. An algorithm which is allowed to evaluate function f n times and obtains at worst(over the set $\mathbb{F}$) an interval that contains the solution of length l is said to be an optimal strategy if there is no other strategy with a worst case length $l' < l$.

Def. Statistical Optimality. Given a problem P defined over a black-box function $f : \Re \to \Re \in \mathbb{F} = \{\text{A given family of functions}\}$. An algorithm allowed to evaluate function f n times and obtains in average(over the set $\mathbb{F}$ [1]) an interval that contains the solution of length l is said to be an optimal strategy if there is no other strategy with an average length $l' < l$.

This work will construct the already known worst case optimal strategies to solve both root-searching problems and unidimensional maximum-searching problems and will also construct a statistically optimal solution to both problems. To the author's comprehension the statistically optimal solutions are original contributions to black-box sequential searching theory.

This work will also construct a result in the field of probability that will be called the Fundamental Theorem of Statistical Characterization to be described in detail in the root-searching chapter. To the author's knowledge this theorem is also original. Given a function f randomly selected in a set of functions $\mathbb{F}$, this theorem gives the means to calculate the probability of $f(x) = y$. The results in black-box sequential searching that come from this theorem can be understood as examples of applications of the constructed probability theory in the area of root-searching and optimization.

[1] It is necessary to suppose that function f is randomly selected from set $\mathbb{F}$ following a given distribution of recurrence. If the set of functions $\mathbb{F}$ is enumerable then the average can be obtained by calculating the statistical mean of $l(f)$ $\forall f \in \mathbb{F}$ (Supposing that the distribution is uniform for example). If the set of functions $\mathbb{F}$ isn't enumerable is is necessary to know a mapping from a set $U \in \Re^n \mid n < \infty$ to the set of functions with a given probability distribution over U to then calculate the average performance. When the functions being investigated are empirical, then, the term "average" may be comprehended as the average performance over an unbiased sample.

3

Optimal Root Searching

"There are those who seek knowledge for the sake of knowledge; that is Curiosity.
There are those who seek knowledge to be known by others; that is Vanity.
There are those who seek knowledge in order to serve; that is Love." Bernard of Clairvaux

3.1 Classical Root Searching

Root Search

Root search or root-finding algorithms are iterative sequential numerical methods that are constructed to solve the following problem:

Root Problem. *Given* $f : [a,b] \to \Re \mid [a,b] \subset \Re$ *a continuous function with the values* $y_a = f(a) < 0$ *and* $y_b = f(b) > 0$ *known. Find* x^* *that satisfies* $f(x^*) = 0$.

The existence of x^* is guaranteed by the intermediate value theorem.
Classical non-randomized sequential root-searching algorithms follow the following bracketing strategy:

1. Chose $x \in [a,b]$
2. Evaluate $f(x)$
3. If $f(x) > 0$ make $b \leftarrow x$
   else if $f(x) < 0$ make $a \leftarrow x$
   else if $f(x) = 0$ return $x^* = x$ and stop.
4. If stopping criteria is met, return interval $[a,b]$ and end.
   Else go to step 1.

Different strategies, or methods, differ essentially in step 1, that is, in how to chose $x \in [a,b]$. Detailed description of a collection of strategies may be found in [7, 5, 4] and in the majority of contemporary Numerical Calculus books, most procedures rely on predictions of the root location based on interpolations (polynomial, exponential etc.). Some strategies, such as the bisection method and Brent's method [5, 10], guarantee a convergent sequence with a fixed or minimum convergence rate which is commonly interpreted as "robustness" of a strategy. Brief descriptions of three methods of choosing $x \in [a,b]$ (Step 1) are given: the bisection method, the secant method and the modified Brent's method.

Bisection Method *Chose* $x = (a + b)/2$.

Secant Method *Let* r *be the line segment between points* $P_a = (a, f(a))$ *and* $P_b = (b, f(b))$. *Take* $x =$ *solution of* $(x, 0) \in r$. Therefore:

$$x = \frac{b * f(a) - a * f(b)}{f(a) - f(b)}$$

The following method alters slightly the format presented in the generic algorithm as will be seen:

Modified Brent's Method *Choose in step 1 two points to evaluate instead of only one:* $x_{1/2}$ *and* x_i. *Let* $x_{1/2} = (a+b)/2$ *and* $y_{1/2} = f(x_{1/2})$. *Let* x_i *be the inverse quadratic interpolation of* $(a, y_a), (x_{1/2}, y_{1/2})$ *and* (b, y_b). *Evaluate the function at* $y_i = f(x_i)$ *and proceed to choosing the two points among* $a, x_i, x_{1/2}, b$ *that bracket the solution with a minimum distance. If stopping criteria isn't met, restart the algorithm .*

The secant method is an example of probably one of the simplest interpolation methods that try to obtain a fast convergence rate with a prediction of the root's location.

What's interesting about Brent's method is that it is easy to demonstrate that it presents a minimal convergence rate and still uses curve fitting that propitiates a polynomial convergence[5, 10].

On the other hand, the bisection strategy not only gives a constant convergence rate at each evaluation of function f, as it is mini-maximal in the sense that given n evaluations of the function, the bisection method will present the best convergence rate for a worst case situation. The following theorem is an original demonstration, to the author's knowledge, of an already known result that will use J.Kiefer's [3] nomenclature:
$\mathbb{D}$: set of all closed intervals within [a,b].
$D \in \mathbb{D}$: Terminal decision.
$n \in \mathbb{N}$: An integer.
Let $g_k : [a, b]^{k+1} \times \Re^{k+1} \to [a, b]$ be functions $\mid k = 1, ..., n$.
And s & $t : [a, b]^{n+2} \times \Re^{n+2} \to [a, b]$: be functions $\mid s \leq t$.
A strategy S_n, will be the set $S_n = \{a, b, y_a, y_b, g_1, ..., g_n, s, t\}$ that can be computed sequentially as follows:

$$x_k = g_k(a, b, x_1, ..., x_{k-1}, y_a, y_b, f(x_1), ..., f(x_{k-1})) \mid k = 1, ..., n$$
$$D(f, S) = [s(a, b, x_1, ..., x_n, y_a, y_b, f(x_1), ..., f(x_n)), \\ t(a, b, x_1, ..., x_n, y_a, y_b, f(x_1), ..., f(x_n))]$$

And also, let $\mathbb{S}_n$ be the set of strategies $\{S_n \mid x^* \in D(f, S_n) \forall f \in F$ where F is the set of non decreasing C^0 functions.$\}$. With those definitions, given n function evaluations the bisection strategy $S_n^{1/2}$ is mini-maximal in the following sense:

Theorem 1.

$$\inf_{S \in \mathbb{S}_n} \sup_{f \in F} L(D(f, S)) = \sup_{f \in F} L(D(f, S_n^{1/2})) = \left(\frac{1}{2}\right)^n \times (b - a)$$

Where L is the length function.

Proof. The second equality given by Theorem 1 is evident, and therefore only the first equality shall be proven.

Suppose a strategy $S_n^o \in \mathbb{S}_n$ exists such that:

$$\sup_{f \in F} L(D(f, S_n^o)) < \sup_{f \in F} L(D(f, S_n^{1/2})) = \left(\frac{1}{2}\right)^n \times (b - a)$$

Given $x_1, ..., x_n$ and $y_1 = f(x_1), ..., y_n = f(x_n)$, s and t must be given by:

$s(a, b, x_1, ..., x_n, y_a, y_b, y_1, ..., y_n) = \text{argmin}_{x=a,b,x_1,...,x_n} \|f(x)\|$ with $f(x) \leq 0$
$t(a, b, x_1, ..., x_n, y_a, y_b, y_1, ..., y_n) = \text{argmin}_{x=a,b,x_1,...,x_n} \|f(x)\|$ with $f(x) \geq 0$

The above equations are true for if not it is possible to construct $S_n^{o+} \in \mathbb{S}_n$ by copying S_n^o and substituting s and t for the above functions and obtaining a $\sup_{f \in F} L(D(f, S_n^{o+})) < \sup_{f \in F} L(D(f, S_n^o))$.

The proof is obtained by observing that given strategy $S_n^o \in \mathbb{S}_n$ it is possible to construct a function $f^* \in F$ that $\sup_{f \in F} L(D(f, S_n^o)) = L(D(f^*, S_n^o))$ with the following procedure:

Let $k = 1$

While $k \neq n$:

1. Evaluate $x_k = g_k(a, b, x_1, ..., x_{k-1}, y_a, y_b, f(x_1), ..., f(x_{k-1}))$
2. Make $f^*(x_k) = f(\text{argmin}_{x=x_a,x_b} L(x_k, x))$
3. Make $k \leftarrow k + 1$
4. Return to step 1.

This way at best, the length of the root location given by $L(s, t)$ will reduce in size by half at each iteration yielding a $L(D(f^*, S_n^o))$ with same value of $S_n^{1/2}$. □

Without loss of the validity of the above demonstration the hypothesis of a non decreasing function may be relaxed and in fact removed. A slight modification to the definition of a root of a function allows the removal of the hypothesis of the function being continuous instead of the hypothesis of the function being non decreasing. This new definition, slightly different to the definition given by $\{x \mid f(x) = 0\}$, will be useful along this work to search for roots in a discrete function environment; it is given by:

Def. A root of a non decreasing function f is $\{x \mid f(x) = 0\} \cup \{x \mid f(x) < 0$ *and* $f(x + \delta) > 0\} \cup \{x \mid f(x) > 0$ *and* $f(x - \delta) > 0\}$ $\forall (x + \delta), (x - \delta) \in D_f\}$, where D_f is the domain of function f and $\delta > 0$.

In this work this extended definition of a root will be adopted, and any reference to the word "root" will be interpreted by this definition.

3.2 Statistical Performance

A natural question arouses with an investigation of the known root searching methods: "Which method converges the fastest?". Although for a worst case scenario this question has already been answered, and for that the bisection method should be elected, applications can commonly prefer methods that converge faster in average

instead of in a worst case scenario. Most commonly applications are computational and therefore functions are discrete instead of continuous as well.

This subsection will show what will be called the Statistical Method and will rigorously prove that it has the fastest convergence rate in average over the set of non decreasing discrete functions. The theory elaborated to construct the Statistical Method can be used to obtain fastest convergence rate in average over different sets of functions as long as a statistical characterization, as will be shown in this work, is done for the desired set. The following theory is an original contribution of the author:

Statistical Characterization of Sets of Functions

Let P and Q be sets and let $\mathbb{F}$ be a set of functions defined from $P \to Q$

Def. A statistical characterization of $\mathbb{F}$ will be the density of probability:

$$\rho : P \times Q \to [0, \infty)$$

The statistical characterization $\rho(p, q)$ will measure the density of probability of randomly selecting a function f from set $\mathbb{F}$ and obtaining $f(p) = q$. If ρ is a statistical characterization of $\mathbb{F}$ then we say that ρ characterizes $\mathbb{F}$ or $\mathbb{F}$ is characterized by ρ.

The following original theorem will show how to obtain the statistical characterization of any set of functions that is described by a set of parameters. This theorem is central and may be used to describe finite polynomial sets, Fourier series and combinations of functions with a constitutive relation given by parameters. The comprehension of the following Theorem is therefore essential to further proceed to the construction of the so called Statistical Method. The Fundamental Theorem of Statistical Characterization may be understood as the main result of this work and in fact the other results may be considered as consequences of the following contribution.

The following definitions are given:

Def. The derivative of the inverse of function $h : [0, 1]^p \to \Re^m$ evaluated at $y \in \Re^m$, given by $\left\| \frac{\mathrm{d}}{\mathrm{d}y}\left(h^{-1}(y)\right) \right\|$ is defined by :

$$\left\| \frac{\mathrm{d}}{\mathrm{d}y}\left(h^{-1}(y)\right) \right\| \equiv \lim_{\delta \to 0} \frac{\int_{[h^{-1}(B(y,\delta))]} \mathrm{d}v}{\int_{B(y,\delta)} \mathrm{d}v}$$

This definition, with the proper assumptions over function h (such as $m = p$ and be locally diffeomorphism) $\left\| \frac{\mathrm{d}}{\mathrm{d}y}\left(h^{-1}(y)\right) \right\|$ is in fact given by $\left\| \det\left(Dh^{-1}(y)\right) \right\|$ for by the change of variables theorem $\lim_{\delta \to 0} \frac{\int_{[h^{-1}(B(y,\delta))]} \mathrm{d}v}{\int_{B(y,\delta)} \mathrm{d}v} = \lim_{\delta \to 0} \frac{\int_{B(y,\delta)} \left\| \det\left(Dh^{-1}(y)\right) \right\| \mathrm{d}v}{\int_{B(y,\delta)} \mathrm{d}v} = \left\| \det\left(Dh^{-1}(y)\right) \right\|$.

Now let $g : [0, 1]^p \times \Re^n \to \Re^m$ be a function called a constitutive function.

Theorem 2. ***Fundamental Theorem of Statistical Characterization*** *Given a set of functions* $\mathbb{F} = \{f_a : \Re^n \to \Re^m \mid f_a(x) = g(a,x), a \in [0,1]^p \subset \Re^p\}$ *and* $\mathbb{H} = \{h_x : [0,1]^p \subset \Re^p \to \Re^m \mid h_x(a) = g(a,x)\}$. *Given that function* $f_a \in \mathbb{F}$ *will be randomly selected by choosing* $a \in [0,1]^p$ *with a uniform probability, then* $\rho : \Re^n \times \Re^m$ *for* $\mathbb{F}$ *is given by:*

$$\rho(x,y) = \left\| \frac{\mathrm{d}}{\mathrm{d}y}\left(h_x^{-1}(y)\right) \right\|$$

Proof.

$$\rho(x,y) = \lim_{\delta\to 0} \frac{P[f(x) \in B(y,\delta)]}{\int_{B(y,\delta)} \mathrm{d}v} = \lim_{\delta\to 0} \frac{P[g(a,x) \in B(y,\delta)]}{\int_{B(y,\delta)} \mathrm{d}v} =$$

$$= \lim_{\delta\to 0} \frac{P[a \in h_x^{-1}\Big(B(y,\delta)\Big)]}{\int_{B(y,\delta)} \mathrm{d}v} = \lim_{\delta\to 0} \frac{Vol\left[h_x^{-1}\Big(B(y,\delta)\Big)\right]}{\int_{B(y,\delta)} \mathrm{d}v \times Vol\,[0,1]^n} =$$

$$= \lim_{\delta\to 0} \frac{\int_{[h_x^{-1}(B(y,\delta))]} \mathrm{d}v}{\int_{B(y,\delta)} \mathrm{d}v \times \int_{[0,1]^n} \mathrm{d}v} = \lim_{\delta\to 0} \frac{\int_{[h_x^{-1}(B(y,\delta))]} \mathrm{d}v}{\int_{B(y,\delta)} \mathrm{d}v} \equiv \left\| \frac{\mathrm{d}}{\mathrm{d}y}\left(h_x^{-1}(y)\right) \right\|$$

□

Theorem 2 provides a powerful tool capable of calculating the statistical characterization of many sets of functions, particularly sets that are defined by a constitutive function $g(a,x)$. The following example and proposed exercises illustrate sets of parametrized functions that may be characterized by using the Fundamental Theorem of Statistical Characterization.

Example 1. *Given a set of functions* $\mathbb{F} = \{f_a(x) = a^x \mid a \in [1,2]\}$. *Calculate* $\rho : \Re \times \Re$ *for* $\mathbb{F}$, *given that function* $f_a \in \mathbb{F}$ *will be randomly selected by choosing* $a \in [1,2]$ *with a uniform probability:*

Solution. *For* $y \in [1, 2^x]$*:*

$$\rho(x,y) = \lim_{\delta\to 0} \frac{P[f(x) \in (y-\delta/2, y+\delta/2)]}{\delta} = \lim_{\delta\to 0} \frac{P[a^x \in (y-\delta/2, y+\delta/2)]}{\delta} =$$

$$= \lim_{\delta\to 0} \frac{P[a \in ((y-\delta/2)^{\frac{1}{x}}, (y+\delta/2)^{\frac{1}{x}})]}{\delta} = \lim_{\delta\to 0} \frac{(y+\delta/2)^{\frac{1}{x}} - (y-\delta/2)^{\frac{1}{x}}}{\delta[2-1]} =$$

$$= \lim_{\delta\to 0} \frac{(y+\delta/2)^{\frac{1}{x}} - (y-\delta/2)^{\frac{1}{x}}}{\delta} = \frac{\mathrm{d}}{\mathrm{d}y}\left(y^{\frac{1}{x}}\right) = \frac{1}{x} y^{\left(\frac{1}{x}-1\right)}$$

For $y \notin [1, 2^x]$ *then* $\rho(x,y) = 0$*, so:*

$$\rho(x,y) = \begin{cases} \frac{1}{x} y^{\left(\frac{1}{x}-1\right)} \mid \textit{for } y \in [1, 2^x] \\ 0 \mid \textit{for } y \notin [1, 2^x] \end{cases}$$

It may be observed that with the use of Theorem 2 this result would be immediate once $h_x(a) = a^x \therefore h_x^{-1}(y) = y^{\frac{1}{x}}$.

Exercise 1. *Calculate $\rho(x, y)$ for the set of n'th degree polynomials $\mathbb{P} = \{P(x) = \sum_{i=0\ldots n} a_i x^i\}$ with polynomial coefficients chosen at random in a uniform distribution from $[0, 1]$.*

Exercise 2. *Calculate $\rho(x, y)$ for the set of n'th degree polynomials with polynomial coefficients defined by $\mathbb{P} = \{P(x) = \sum_{i=0\ldots n}(a_i x)^i\}$ chosen at random in a uniform distribution from $[0, 1]$. Why is the result different from the Exercise 1?*

Exercise 3. *Calculate the distribution at which $a \in [1, 2]$ must be chosen to obtain a characterization of the set of functions defined by Example 1 dependent only on variable x.*

If sets P & Q are finite the number of elements of each set will be represented by: $\#\{P\} = n_p$ & $\#\{Q\} = n_q$. Assuming that for a root searching problem it is necessary to construct the statistical characterization of the set of non decreasing functions $f : [a, b] \to [y_a, y_b]$ computationally represented with a resolution of n_p points uniformly distributed over $[a, b]$ and n_q points distributed over $[y_a, y_b]$ the problem becomes a discrete version of the initial problem.

Adopting the generalization of the definition of a root given at the end of section 3.1 it is still possible to guarantee the existence of a root over the computational representation of the problem. Now, assuming a uniform probability of occurrence of each non decreasing function over:

$$\{a, a + \delta_p, ..., (n_p - 1) \times \delta_p = b\} \to \{y_a, y_a + \delta_q, ..., (n_q - 1) \times \delta_q = y_b\}$$

Where $\delta_p = \frac{b-a}{n_p}$ and $\delta_q = \frac{y_b - y_a}{n_q}$.

It is possible to calculate ρ for this set as follows:

Lemma 1. *The number of non decreasing functions from $A = \{1, ..., n_p\}$ to $B = \{1, ..., n_q\}$ is given by :*

$$\#f\Big\uparrow_{\{1,\ldots,n_p\}}^{\{1,\ldots,n_q\}} = \#solutions\{c_1 + c_2 + ... + c_{n_q} = n_p \mid c_i \in \mathbb{N}^*\} = \frac{(n_p + n_q - 1)!}{n_p!(n_q - 1)!}$$

Proof. Given a solution function f let $c_i = \#\{x \in \{1, ..., n_p\} \mid f(x) = i\}$ be the number of elements in A that are on the one dimensional curve level defined by the hight i. This way the sum of all elements from each possible curve level will be the exact total of elements in the domain of the function A. Therefore for each non decreasing solution function there is a unique solution to $\{c_1 + c_2 + ... + c_{n_q} = n_p \mid c_i \in \mathbb{N}^*\}$. Once the the solution function must be non decreasing, for each solution $\{c_1, ..., c_{n_q}\}$ to $\{c_1 + c_2 + ... + c_{n_q} = n_p \mid c_i \in \mathbb{N}^*\}$ there is also one unique function to which c_i is the number of elements in the domain A with function value i or simply $c_i = \#\{x \in [1, n_p] \mid f(x) = i\}$.

Therefore first part of the equation of Lemma 1 yields.

The second part of the equation is a classic combinatorial analysis result whose demonstration may be found in [6] or in most combinatoric books. □

Theorem 3. *The Statistical Characterization ρ of the set of non decreasing functions defined from $A = \{1, ..., n_p\}$ to $B = \{1, ..., n_q\}$ is given by :*

$$\rho(x,y) = \frac{\frac{(x-1+y-1)!}{(x-1)!(y-1)!} \times \frac{(n_p-x+n_q-y)!}{(n_p-x)!(n_q-y)!}}{\frac{(n_p+n_q-1)!}{n_p!(n_q-1)!} \times \delta_q} \mid (x,y) \in A \times B$$

Proof. Once a uniform distribution of probability for each non decreasing function is assumed, the probability $P[f(x) = y]$ is obtained by calculating the number of favourable cases divided by the total number of solutions. The number of favourable cases is given by the total of non decreasing functions with the constraint $f(x) = y$, this is calculated by evaluating the number of non decreasing functions from $\{1, ..., x-1\}$ to $\{1, ..., y\}$ times the number of non decreasing functions from $\{x+1, ..., n_p\}$ to $\{y, ..., n_q\}$ and the total number of non decreasing functions is given by lemma 1. This way :

$$P[f(x) = y] = \frac{\#f\Big|_{\{1,...,x-1\}}^{\{1,...,y\}} \times \#f\Big|_{\{x+1,...,n_p\}}^{\{y,...,n_q\}}}{\#f\Big|_{\{1,...,n_p\}}^{\{1,...,n_q\}}}$$

Substituting the equation by the result given by lemma 1 and dividing by δ_q the result yields . □

Theorem 2 provides the means to obtain $\rho(x, y)$ for sets of parametrized functions f defined from $\Re^n$ to $\Re^m$. On the other hand Theorem 3 exemplifies the calculation of $\rho(x, y)$ for a known set of discrete functions. The following sections will build on this information; that is, the following sections will suppose that given a set of functions $\mathbb{F}$ (be it continuous or discrete) ρ can be calculated.

3.2.1 Statistical Root Searching Method

With a statistical characterization ρ such as given by Theorem 2 it is now possible to display strategy $\bar{S}_n \in \mathbb{S}_n$, the statistical method. Initially $\bar{S}_1 \in \mathbb{S}_1$ shall be calculated, but first, the computational problem to be solved will be displayed for convenience:

Discrete Root Problem *Given $f : \{a, ..., b\} \to \{y_a, ..., y_b\}$ a discrete non decreasing function with n_p elements from a to b and n_q elements from $y_a < 0$ to $y_b > 0$, with a, b, y_a and y_b known. Find $x^* \in [a, b]$ where x^* is a root of f in the extended definition.*

$\bar{S}_1$ can be obtained by calculating the point $x_1 \in [a, b]$ such that in average the greatest convergence rate is achieved, or in other words, the greatest interval is discarded.

Lemma 2. *The average length of the discarded region in one step of evaluation of function f in the generic root searching algorithm in point x is:*

$$s^1(x) = \int_{y_a}^{y_b} \rho(x,y) l(x,y) \mathrm{d}y$$

Where:

$$l(x,y) = \begin{cases} (b-x) \mid for\ y > 0 \\ (b-a) \mid for\ y = 0 \\ (x-a) \mid for\ y < 0 \end{cases}$$

Proof. Step 3 of the bracketing strategy will perform at each iteration of the algorithm the following discard operations:

1. If $f(x) > 0$ discard interval $(x, b]$
2. If $f(x) = 0$ discard interval $[a, x) \cup (x, b]$
3. If $f(x) < 0$ discard interval $[a, x)$

Therefore the average length of the discarded region is given by:

$$s^1(x) = P[f(x) > 0] \times (b-x) + P[f(x) = 0] \times (b-a) + P[f(x) < 0] \times (x-a)$$

Using the definition of a probability distribution and integrals in discrete spaces the result is obtained. □

Therefore if $\bar{S}_1$ is the strategy for which in one step the greatest discard operation is done in average, then $\bar{S}_1$ shall evaluate function f in x_1:

$$x_1 = argmax_{x \in [a,b]} s^1(x)$$

Theorem 4. *$\bar{S}_n \in \mathbb{S}_n$ is given by a sequence of evaluations of function f on points $x_n, ..., x_1$ given by:*

$$x_i = argmax_{x \in [a,b]} s^i(x)$$

a and b are updated at each iteration given by step 3 of the generic bracketing strategy and $s^i(x)$ for $i = 2, ..., n$ is given by:

$$s^i(x) = \int_{y_a}^{y_b} \rho(x,y) \left[l(x,y) + \max_{\alpha \in [a^*, b^*]} \left(s^{i-1}(\alpha) \right) \right] \mathrm{d}y$$

s^{i-1} in the above equation is evaluated in $[a^*, b^*]$, that is the updated interval of function f when $f(x) = y$, and $l(x)$ is the same as given by Lemma 2.

Proof. When there are i steps left for evaluation, x_i must be chosen to maximize the average sum of the discard operation given by that step $l(x,y)$ plus the average discarded quantity for the next $i-1$ steps. Once the strategy of choice of x_k is always to evaluate at maximum point of s^k, the average discarded quantity for the next $i-1$ iterations is given by $\max_{\alpha \in [a^*, b^*]} \left(s^{i-1}(\alpha) \right)$ where $[a^*, b^*]$ is the updated interval by step 3 of the bracketing strategy given by the current function interval $[a, b]$ when evaluated at x and obtaining $f(x) = y$. □

One may notice that for a variety of sets of functions the value of ρ, s^i and even in some cases x_n may be pre-calculated. Discrete uni-modal functions or n^{th} degree polynomial sets are examples of sets that may have ρ, s^i and x_n pre-evaluated in order to establish efficient root-searching algorithms in a statistical sense. Hopefully the numerical experiments at the end of this chapter and meditation over the proposed exercises will convince the reader of these statements.

The following lemma may prove itself a useful tool to evaluate s^1 in case the Statistical Characterization $\rho(x, y)$ isn't known over a set of strictly increasing functions $\mathbb{F}$ but the density of probability $\rho[f(x) = 0] = \rho(x, 0)$ is over domain P.

Lemma 3. *s^1 is given by:*

$$s^1(x) = \int_{y_a}^{y_b} \rho(x, y) l(x, y) \mathrm{d}y = \int_a^b \rho[f(\alpha) = 0] l(x, x - \alpha) \mathrm{d}\alpha$$

Proof.

$$\int_a^b \rho[f(\alpha) = 0] l(x, x - \alpha) \mathrm{d}\alpha =$$

$$= P[f(\alpha < x) = 0] l(x, (x - \alpha) > 0) + P[f(\alpha = x) = 0] l(x, 0) + P[f(\alpha > x) = 0] l(x, (x - \alpha) < 0) =$$

$$= P[f(x) > 0](b - x) + P[f(x) = 0](b - a) + P[f(x) < 0](x - a) =$$

$$= \int_{y=y_a}^{y_b} \rho(x, y) l(x, y) \mathrm{d}y =$$

□

Combinatorial Paradox

A natural question arises about the consequences of adopting greater values of n_p and n_q for the grid of the computational root-searching problem. What happens to $\rho(x, y)$ when $n_p, n_q \to \infty$ for the set of non decreasing discrete functions maintaining a proportion α_x, α_y to the observed point?

$$\rho(x, y) = \frac{\frac{(x-1+y-1)!}{(x-1)!(y-1)!} \times \frac{(n_p - x + n_q - y)!}{(n_p - x)!(n_q - y)!}}{\frac{(n_p + n_q - 1)!}{n_p!(n_q - 1)!} \times \delta_q} \mid (x, y) \in A \times B$$

Considering that $n_p = n_q$, $\delta_q = 1/n_q$; $x = \alpha_x n_p$ and $y = \alpha_y n_q$

$$\lim_{n_p \to \infty} \rho(x, y) = \lim_{n_p \to \infty} \frac{\frac{(\alpha_x n_p - 1 + \alpha_y n_q - 1)!}{(\alpha_x n_p - 1)!(\alpha_y n_q - 1)!} \times \frac{(n_p - \alpha_x n_p + n_q - \alpha_y n_q)!}{(n_p - \alpha_x n_p)!(n_q - \alpha_y n_q)!}}{\frac{(n_p + n_q - 1)!}{n_p!(n_q - 1)!} \times \frac{1}{n_q}} =$$

$$= \lim_{n_p \to \infty} \frac{\frac{(n_p(\alpha_x + \alpha_y) - 2)!}{(n_p \alpha_x - 1)!(n_p \alpha_y - 1)!} \times \frac{(n_p(2 - \alpha_x - \alpha_y))!}{(n_p(1 - \alpha_x))!(n_p(1 - \alpha_y))!}}{\frac{(n_p 2 - 1)!}{n_p!(n_p - 1)!} \times \frac{1}{n_p}} =$$

$$= \lim_{n_p \to \infty} \frac{\left(\frac{(n_p \alpha_x)(n_p \alpha_y)}{(n_p(\alpha_x + \alpha_y))(n_p(\alpha_x + \alpha_y) - 1)}\right) \times \frac{(n_p(\alpha_x + \alpha_y))!}{(n_p \alpha_x)!(n_p \alpha_y)!} \times \frac{(n_p(2 - \alpha_x - \alpha_y))!}{(n_p(1 - \alpha_x))!(n_p(1 - \alpha_y))!}}{\frac{n_p}{2n_p} \frac{(2n_p)!}{n_p! n_p!} \times \frac{1}{n_p}} =$$

$$= 2\frac{\alpha_x\alpha_y}{(\alpha_x+\alpha_y)^2}\lim_{n_p\to\infty}\frac{\frac{(n_p(\alpha_x+\alpha_y))!}{(n_p\alpha_x)!(n_p\alpha_y)!}\times\frac{(n_p(2-\alpha_x-\alpha_y))!}{(n_p(1-\alpha_x))!(n_p(1-\alpha_y))!}}{\frac{(2n_p)!}{n_p!n_p!}\times\frac{1}{n_p}} =$$

Using Stirling's approximation: $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, where $f(n) \sim g(n)$ means that $\lim_{n\to\infty}\frac{f(n)}{g(n)} = 1$

$$=_{\ldots}\lim_{n_p\to\infty}\frac{\frac{\sqrt{2\pi n_p(\alpha_x+\alpha_y)}\left(\frac{n_p(\alpha_x+\alpha_y)}{e}\right)^{n_p(\alpha_x+\alpha_y)}}{\sqrt{2\pi n_p\alpha_x}\left(\frac{n_p\alpha_x}{e}\right)^{n_p\alpha_x}\sqrt{2\pi n_p\alpha_y}\left(\frac{n_p\alpha_y}{e}\right)^{n_p\alpha_y}}\times\frac{\sqrt{2\pi n_p(2-\alpha_x-\alpha_y)}\left(\frac{n_p(2-\alpha_x-\alpha_y)}{e}\right)^{n_p(2-\alpha_x-\alpha_y)}}{\sqrt{2\pi n_p(1-\alpha_x)}\left(\frac{n_p(1-\alpha_x)}{e}\right)^{n_p(1-\alpha_x)}\sqrt{2\pi n_p(1-\alpha_y)}\left(\frac{n_p(1-\alpha_y)}{e}\right)^{n_p(1-\alpha_y)}}}{\frac{\sqrt{2\pi 2n_p}\left(\frac{2n_p}{e}\right)^{2n_p}}{\sqrt{2\pi n_p}\left(\frac{n_p}{e}\right)^{n_p}\sqrt{2\pi n_p}\left(\frac{n_p}{e}\right)^{n_p}}\times\frac{1}{n_p}} =$$

$$= \ldots\lim_{n_p\to\infty}\frac{\frac{\sqrt{n_p}\left(n_p(\alpha_x+\alpha_y)\right)^{n_p(\alpha_x+\alpha_y)}}{\sqrt{n_p}\left(n_p\alpha_x\right)^{n_p\alpha_x}\sqrt{n_p}\left(n_p\alpha_y\right)^{n_p\alpha_y}}\times\frac{\sqrt{n_p}\left(n_p(2-\alpha_x-\alpha_y)\right)^{n_p(2-\alpha_x-\alpha_y)}}{\sqrt{n_p}\left(n_p(1-\alpha_x)\right)^{n_p(1-\alpha_x)}\sqrt{n_p}\left(n_p(1-\alpha_y)\right)^{n_p(1-\alpha_y)}}}{\frac{\sqrt{n_p}\left(2n_p\right)^{2n_p}}{\sqrt{n_p}\left(n_p\right)^{n_p}\sqrt{n_p}\left(n_p\right)^{n_p}}\times\frac{1}{n_p}} =$$

$$= \ldots\lim_{n_p\to\infty}\frac{\frac{\sqrt{n_p}\left(\alpha_x+\alpha_y\right)^{n_p(\alpha_x+\alpha_y)}}{\sqrt{n_p}\left(\alpha_x\right)^{n_p\alpha_x}\sqrt{n_p}\left(\alpha_y\right)^{n_p\alpha_y}}\times\frac{\sqrt{n_p}\left(2-\alpha_x-\alpha_y\right)^{n_p(2-\alpha_x-\alpha_y)}}{\sqrt{n_p}\left(1-\alpha_x\right)^{n_p(1-\alpha_x)}\sqrt{n_p}\left(1-\alpha_y\right)^{n_p(1-\alpha_y)}}}{\frac{\sqrt{n_p}\left(2\right)^{2n_p}}{\sqrt{n_p}\sqrt{n_p}}\times\frac{1}{n_p}} =$$

$$= \ldots\lim_{n_p\to\infty}\sqrt{n_p}\times(\psi(\alpha_x,\alpha_y))^{n_p}$$

The reader may observe that the above limit assumes only two possible values:

$$\lim_{n_p\to\infty}\sqrt{n_p}\times(\psi(\alpha_x,\alpha_y))^{n_p} = 0, \text{ if } \psi(\alpha_x,\alpha_y) < 1$$
$$\lim_{n_p\to\infty}\sqrt{n_p}\times(\psi(\alpha_x,\alpha_y))^{n_p} = \infty, \text{ if } \psi(\alpha_x,\alpha_y) \geq 1$$

For $\alpha_y = \alpha_x$ $\psi(\alpha_x,\alpha_y) = 1$. [1]

This result is highly counter-intuitive for one may say that the characterization of the set of discrete increasing functions with $n_p \to \infty$ will yield a result that converges to the characterization of the set of increasing functions in $[a,b] \to [a,b]$ or at least $[a,b] \cap \mathbb{Q}$. If this interpretation in fact is true then the above limit "says" that increasing functions defined from $[a,b] \to [a,b]$ will have $\forall\delta$ a null probability of $f(\alpha_x) \neq \alpha_y \mid \alpha_y \in [\alpha_x-\delta, \alpha_x+\delta]$ which is to say at least unexpected but still possible. Such assumptions can be misguiding and an example of why such interpretations should be avoided may be observed by analysing the following reflection operation defined over the set of increasing functions $\mathbb{C} = \{c : [0,1] \to [0,2] \mid \text{c is increasing function}\}$ to $\mathbb{U} = \{u : [0,1] \to [0,1] \mid \text{u is uni-modal}\}$:

[1] If instead the assumption that $n_q = kn_p$ was adopted over $n_p = n_q$ the result $\psi(\alpha_x,\alpha_y) = 1$ for $\alpha_y = k\alpha_x$ would also be valid.

$$\lambda : \mathbb{C} \to \mathbb{U}$$

$$u(x) = \lambda(c(x)) = \begin{cases} c(x) & if\ c(x) \leq 1 \\ 2 - c(x) & if\ c(x) > 1 \end{cases}$$

It is easy to demonstrate that λ is a one-to-one and therefore for every increasing function in $[0,1] \times [0,2]$ there is a uni-modal function in $[0,1] \times [0,1]$. Lemma 8 (next chapter) proves that the same reflection operation defined over discrete functions isn't one-to-one and would lead to very different conclusions if uni-modal functions were studied by the reflection operation when the number of elements in the domain and range of the uni-modal functions where evaluated at taking the limit of the grid going to infinite ($n_p, n_q \to \infty$).

Any way the hypothesis that the limit in fact represents a characterization of the set of increasing functions can still be true. In fact it is important to remember that a statistical characterization of a set of functions isn't necessarily unique. This isn't surprising for in any set of parametrized functions the characterization depends on the parametrization adopted, therefore if the above result in fact represents the characterization of this set it isn't necessarily the best representation of the recurrence of $f(x) = y$ in human applications.

3.2.2 Statistical Method x Mini-maximal Method, a Pareto Set

Now that the statistical method was presented and its optimality was demonstrated, this section will show that between the Statistical method $\bar{S}_n$ and the bisection method $S_n^{1/2}$ there is a set of strategies that are Pareto optimal. For this the meaning of Pareto optimality and the corresponding multiple objectives for the set of strategies are displayed for convenience.

Pareto Optimality

Def. Given a set A and a function $f : A \to R^n$ a solution $x^* \in A$ is said to be an optimal solution of f (or Pareto optimal) if $\nexists y \in f(A) \mid y \neq f(x^*)$ & $y(i) \geq f_i(x^*)$ $\forall i = 1...n$ where f_i is the i'th coordinate of f. For simplicity of notation $y^* = f(x^*)$ is also called an optimal solution of f. The set of Pareto optimal solutions is called the Pareto optimal set or simply Pareto set.

As can be observed, a Pareto optimal set is generally not constituted by a single point but by a set of points. Problems that use this concept of optimality are usually referred to as multi-objective optimization problems or vector optimization problems; for more information on vector optimization refer to [1].

Given a set of functions $\mathbb{F}$ described by a set of parameters $q \in [0,1]^p$ (as in the fundamental theorem of statistical characterization) defined over $[a,b] \times \Re$ with $f(a) = y_a < 0$ and $f(b) = y_b > 0$. Let $\mathbb{S}_n$ be the set of root searching strategies with n steps that evaluates functions $f \in \mathbb{F}$ randomly selected by selecting $q \in [0,1]^p$ with uniform probability. Then if $S \in \mathbb{S}_n$ two objectives are defined:

Def. The worst case objective $\underline{O}_n$ (or the conservative objective) is a function defined for all $S \in \mathbb{S}_n$. It is given by the length of the terminal decision for the worst case function $f \in \mathbb{F}$ to S to evaluate. Or in more mathematical terms:

$$\underline{O}_n : \mathbb{S}_n \to \Re \mid \underline{O}_n(S) = \sup_{f \in \mathbb{F}} L(D(f, S))$$

Def. The statistical objective $\overline{O}_n$ is also a function defined for all $S \in \mathbb{S}_n$. It is given by the average length, over the set of functions $\mathbb{F}$ with a given distribution of recurrence, of the terminal decision of $S \in \mathbb{S}_n$ for every function, more precisely:

$$\overline{O}_n : \mathbb{S}_n \to \Re \mid \overline{O}_n(S) = \overline{L(D(f, S))}_{f \in \mathbb{F}}$$

The sub-index n may be omitted an thus be refered as only $\overline{O}$ or $\underline{O}$.

As discussed in the previous sections, given the set of functions $\mathbb{F}$ and its statistical characterization, it is possible to construct the statistically optimal root bracketing strategy $\bar{S}_n$. In a great number of cases, such as over the set of discrete increasing functions, the mini-maximal strategy described by the bisection method $S_n^{1/2}$ is also known. Theorem 1 guarantees that $S_n^{1/2}$ minimizes $\underline{O}_n$ and the construction of Theorem 4 guarantees that $\bar{S}_n$ minimizes $\overline{O}_n$, both over the set of discrete increasing functions.

Def. Let $\bar{S}_n$ be the strategy that minimizes $\overline{O}_n$ and $\underline{S}_n$ the strategy that minimizes $\underline{O}_n$. Thus $S_n^{1/2} = \underline{S}_n$ for the set of continuous functions (which is not true for other sets of functions), the statistical method $\bar{S}_n$ by construction leads always to the minimizer of $\overline{O}$.

Before the next result is presented a question to stimulate the reader is displayed: is it possible to construct a set of functions $\mathbb{F}$ defined by parameters such that $\bar{S}_n = \underline{S}_n$?

Now that the objective functions and their minimizers were defined, it is possible to follow up to the next result that is the purpose of this section. The following lemma answers partially(for a case with $n = 1$) the question: "What is the Pareto optimal[2] set of function $O_n : \mathbb{S}_n \to \Re^2$?" Where O_n is given by:

$$O_n(S) = [\overline{O}_n(S), \underline{O}_n(S)]$$

Lemma 4. *If $\underline{S}_1 = S_1^{1/2}$ for the set of functions $\mathbb{F}$ then the Pareto optimal set of O_1 is a subset of strategies S_1^{λ} that shall evaluate the selected function at point x_λ, where x_λ is the maximizing argument of $s_1(x) \mid \max(x - a, b - x) \leq \lambda$ and $\lambda \in [(b - a)/2, b - a]$, and:*

1. *For $\lambda = b - a$ the Pareto solution $S_1^{\lambda=b-a} = \bar{S}_1$.*
2. *For $\lambda = (b - a)/2$ the Pareto solution $S_1^{\lambda=(b-a)/2} = \underline{S}_1$.*
3. *If s_1 is uni-modal(Is this the case for discrete increasing functions?), then the Pareto optimal set is the set of convex combinations of $\bar{x}$ and $\underline{x}$, where $\bar{x}$ and $\underline{x}$ are given by $\bar{S}$ and $\underline{S}$ respectively.*

Proof. Observe that the constraint $\max(x - a, b - x)$ is the value of the worst case discard.

[2] Actually Pareto optimality was here defined with $\nexists y \in f(A) \mid y \neq f(x)$ & $y(i) \geq f_i(x^*)$ and for this case the inequality is needed to be the opposite, that is $\nexists y \in f(A) \mid y \neq f(x)$ & $y(i) \leq f_i(x^*)$ or simply taking the negative value of O_n

1. For $\lambda = b - a$ the constraint $\max(x - a, b - x) \leq \lambda$ may be abandoned and therefore the minimizing argument of $s_1(x)$ by definition is $\bar{x}$.
2. For $\lambda = (b - a)/2$ the constraint $\max(x - a, b - x) \leq \lambda$ allows only the value of $x = \frac{a+b}{2} = x_{1/2}$ which is by hypothesis $\underline{S}_1$.
3. For λ increasing from $(b - a)/2$ to $b - a$ the constraint $\max(x - a, b - x) \leq \lambda$ is equivalent to $x \in [\frac{a+b}{2} - \delta, \frac{a+b}{2} + \delta]$ for δ increasing from 0 to $(b - a)/2$. If $s_1(x)$ reaches its maximum $\bar{x}$ at $x_{1/2}$ there is nothing to prove, so it will be assumed that $\bar{x} \neq x_{1/2}$. Therefore for δ increasing from 0 to $||\bar{x} - x_{1/2}||$ once s_1 is assumed uni-modal $\max s_1(x) \mid x \in [\frac{a+b}{2} - \delta, \frac{a+b}{2} + \delta]$ has a increasing objective function value and an argument value equal to the extreme that is closest to $\bar{x}$. For $\delta \in [||\bar{x} - x_{1/2}||, (b - a)/2]$ $\max s_1(x)$ with the constraint $x \in [\frac{a+b}{2} - \delta, \frac{a+b}{2} + \delta]$ will have a constant solution given by $\bar{x}$.

□

Lemma 4 gives a good idea of what intermediate solutions look like. Lemma 4 (3) demonstrates that any intermediate value between $\bar{x}$ and $\underline{x}$ are optimal as a "last" step. In fact every Pareto optimal root searching method is a solution of the problem $\min \overline{O}_n(S)\Big|_{\underline{O}_n \leq \lambda}$ for some $\lambda \in \Re$, which reinforces the idea that intermediate root searching strategies are given by the convex combination of $\bar{x}$ and $\underline{x}$. A rigorous construction of the Pareto set of strategies in $\mathbb{S}_n$ is desired but not yet known.

Example 2. (Finding the best version of Brent's Method) *$\underline{O}\left(S_2^{Brent's\ Method}\right)$ at every two steps of Brent's Method has the same value $\underline{O}\left(S_1^{1/2}\right)$ of one step of the bisection method. Find the best method S_2 such that $\underline{O}(S_2) = \underline{O}\left(S_1^{1/2}\right)$.*

Solution. Once the worst case convergence rate of the strategy is fixed as half the convergence rate of the bisection method to find the "best" method S_2 it is necessary to maximize the statistical performance with the constraint of a minimum fixed worst case discard. Therefore it is necessary to solve:

$$\min \overline{O}(S_2) \mid \underline{O}(S_2) = \underline{O}\left(S_1^{1/2}\right) = \frac{b - a}{2}$$

The solution to this optimization problem is in fact an element of the Pareto set of strategies discussed in this subsection. Lemma 4 does not apply to this example once its results extends only over $\mathbb{S}_1$ so it will be necessary to apply the theory of statistical performance to find strategy S_2 solution to the problem.

For this the re-evaluation of the expected discard interval in two steps considering the constraint over mini-maximal performance will be needed. This way the new expected discard interval with the constraint is:

$$s^2(x) \mid_{\underline{O} = \frac{b-a}{2}} = \int_{y_a}^{y_b} \rho(x, y) \left[l(x, y) + \max_{\alpha \in [b^* - \frac{b-a}{2}, a^* + \frac{b-a}{2}]} \left(s^1(\alpha) \right) \right] \mathrm{d}y$$

The argument that maximizes then $s^2(x) \mid_{\underline{O} = \frac{b-a}{2}}$ must be chosen and the final step then at the maximizing argument of $s^1(x)$ over the resulting feasible interval $x \in [b^* - \frac{b-a}{2}, a^* + \frac{b-a}{2}]$. The reason why the new interval of "alowed" points is now $[b^* - \frac{b-a}{2}, a^* + \frac{b-a}{2}]$ and not $[a^*, b^*]$ is to guarantee that the value of $\underline{O}(S_2) = \frac{b-a}{2}$.

□

Other Exercises

Exercise 4. *Let $f(0) = 0, f'(0) = v$ and $f'(i+1) = f(i) + d$ where d is a random variable with a given distribution r. Calculate $\rho(i, j)$.*

Exercise 5. *Suppose that if the value of $f(x_0) = y_0$ is known for a given x_0, and that an acceptable estimate of the value of $f(x)$ is given by a normal distribution with $N(y_0, \sigma(x)^2) = N(y_0, \|x - x_0\|)$. What happens to the $\rho(x, y)$ if the value of function f is evaluated at $x_1 \neq x_0$? (Hint: Consider that two different estimates of the value of the function are acceptable, one from x_0 and one from x_1. Then construct an estimate with a coherent variance for $f(x)$ that considers both estimates with weights that ponder the distance from x_0 and x_1.)*

3.3 Numerical Experiments on Root-Searching

This section will exemplify the constructed theory by comparing the performance of a numerical implementation (a computational algorithm) of the mini-maximal root-searching method with the statistically optimal root searching method and with the secant method. For this the set of non decreasing functions $f : \{1, ..., 30\} \to \{1, ..., 30\}$ was chosen and the extended root problem given by: find $x_z \mid x_z$ is a root of $g_z(x) = f(x) - z$ for $z = 1, ..., 30$ will be studied. This way the average length of the discarded interval for each value of z will be compared.

Methodology:

The constructed algorithm executes two main phases, the first consists of the construction of the statistical characterization of the set of non decreasing functions and a comparison with an estimated characterization $\rho(x, y)$ as a normal distribution with variance and mean proportional to distance to the extremes (as suggested by Exercise 5). The second phase evaluates step 1 in the generic root searching algorithm for the three chosen strategies and compares the value of the average discard given by $\rho(x, y)$ constructed in phase 1. The algorithm also constructs a comparison of the average discard of each strategy with the location of zero z varying from 1 to 30, constructs an equivalent interpolation/root-searching strategy for the three methods and finally s_2 is compared with s_1 for a fixed value of $z = 20$.

The routine was constructed on Matlab 7.10.0(R2010a) according to the following pseudo-code:

Begin algorithm:

"Phase 1:"
calculate and display surface $\rho(i,j)\ \forall(i,j) \in 30 \times 30$
for i = 1:30
calculate $\rho_i^r = \rho^r(i) = \frac{n_q}{n_p} i$ *"Linear estimation for center of distribution"*
calculate $\sigma_i' = \sigma'(i) = \sqrt{\frac{n_p+1-i}{n_p+1}\sigma_0^2(i) + \frac{i}{n_p+1}\sigma_{n_p+1}^2(i)}$
where $\sigma_0^2(i) = \|i\|$ & $\sigma_{n_p+1}^2(i) = \|n_p + 1 - i\|$
"Estimation of standard deviation with variance proportional to distance"
calculate true standard deviation S_i of $\{\rho(i,1), \rho(i,2), ..., \rho(i,30)\}$
end for

plot Normal distribution $N(\rho_i^r, \sigma_i'^2)$ over surface $\rho(i,j)$
plot difference $N(\rho_i^r, \sigma_i'^2) - \rho(i,j)$
plot $i \times S_i$ & $i \times \sigma_i'$

"Phase 2:"
"Coment: With the value of $\rho(x,y)$ the algorithm calculates and displays for each possible value of zero z the following:"
for $z = 1:30$
calculate $s_1(x)$
plot s_1

calculate $\bar{x}(z) =$argmax $\Big(s_1(x)\Big)$ "Statistical Method"
calculate $x_{1/2}(z) = (b+a)/2$ "Bisection Method"
calculate $x_s(z) = \frac{b*f(a)-a*f(b)}{f(a)-f(b)}$ "Secant Method"
evaluate $\bar{s}(z) = s_1(\bar{x})$
evaluate $s_{1/2}(z) = s_1(x_{1/2})$
evaluate $s_s(z) = s_1(x_s)$
end for

plot $s \times z$ for the three strategies
plot $x \times z$ for the three strategies

$z = 20$
calculate $s_1(x)$
calculate $s_2(x)$
plot s_1 & s_2 and mark all maxima

End algorithm.

Results

The following images are the result of the constructed algorithm. The exact script of the routines follows in the addendum section.

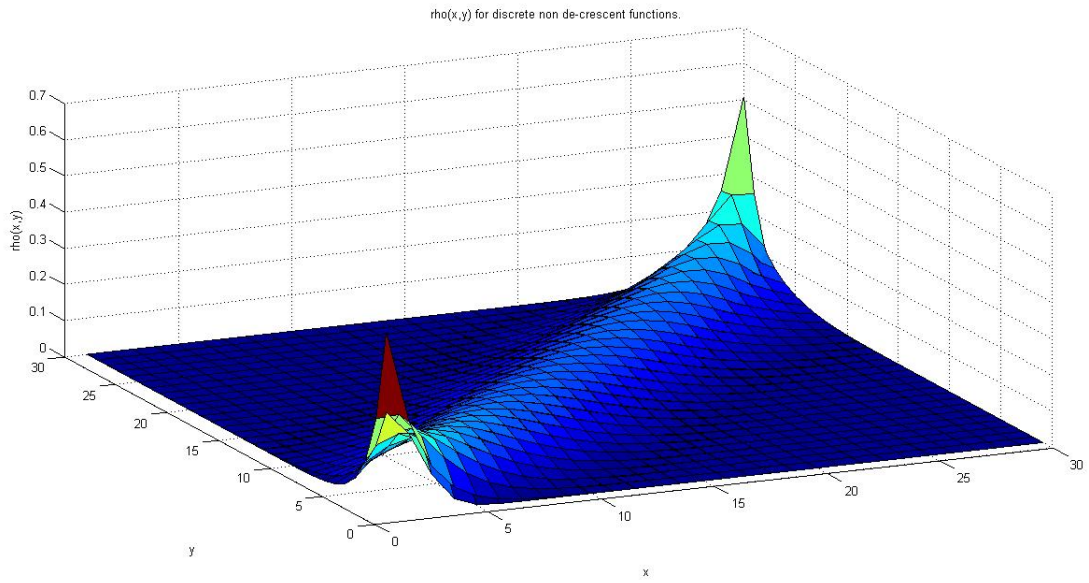


Fig. 3.1. $\rho(x, y)$ and Normal approximation $N(\rho^r, \sigma'^2)$ for non decreasing Functions

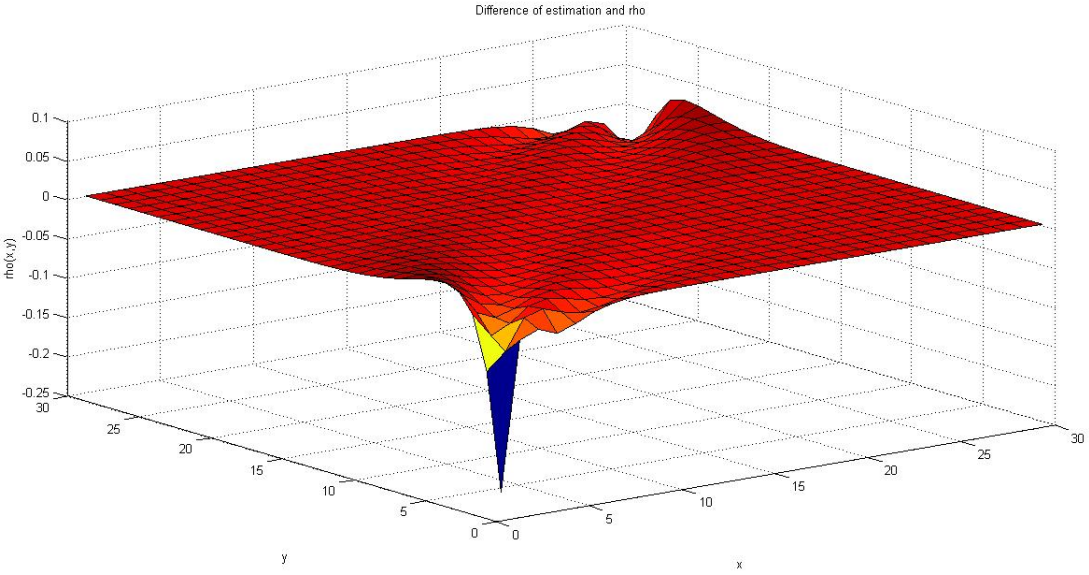


Fig. 3.2. Difference of Normal Approximation $N(\rho^r, \sigma'^2)$ and $\rho(x, y)$

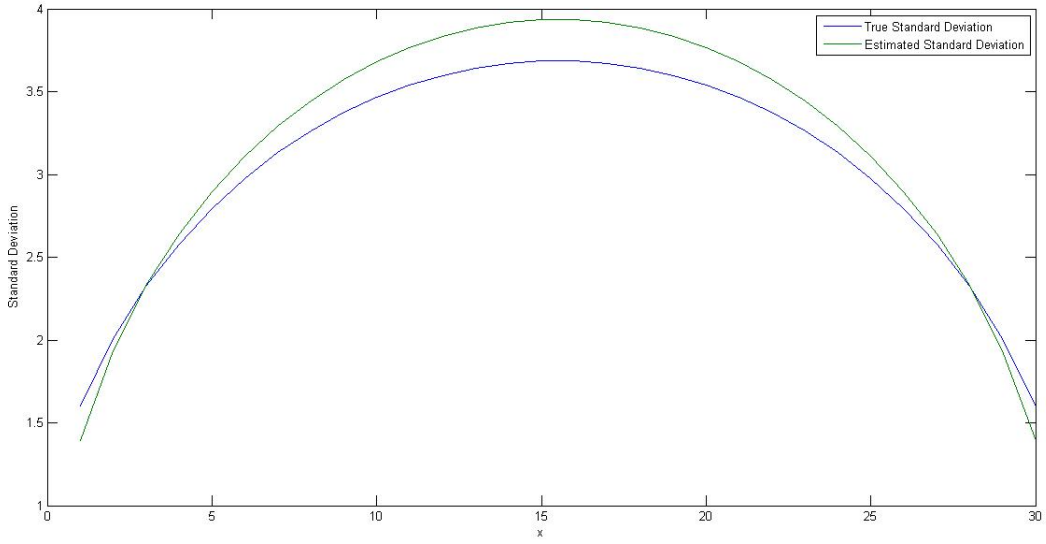


Fig. 3.3. Standard Deviation and Approximation with Variance Proportional to Distance

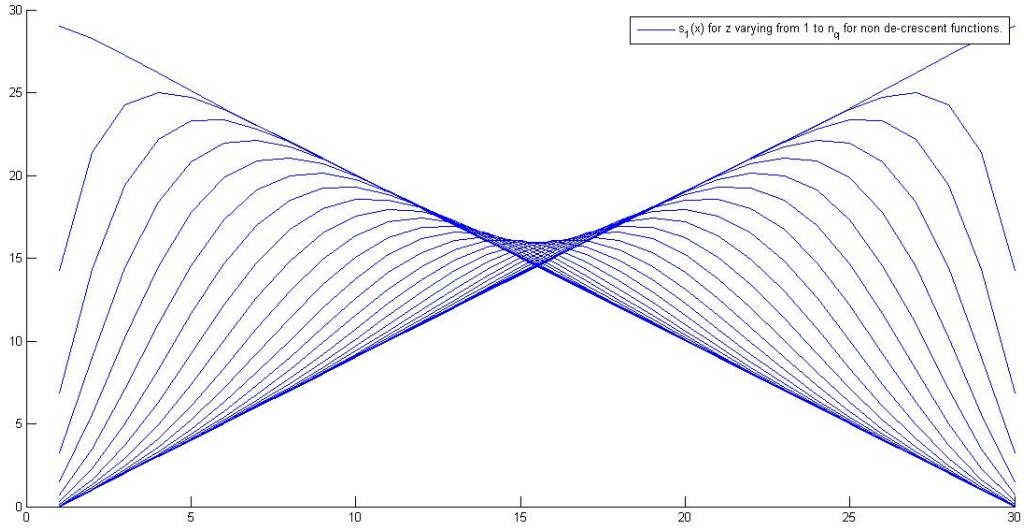


Fig. 3.4. $s_1(x)$ for zero varying from 1 to 30

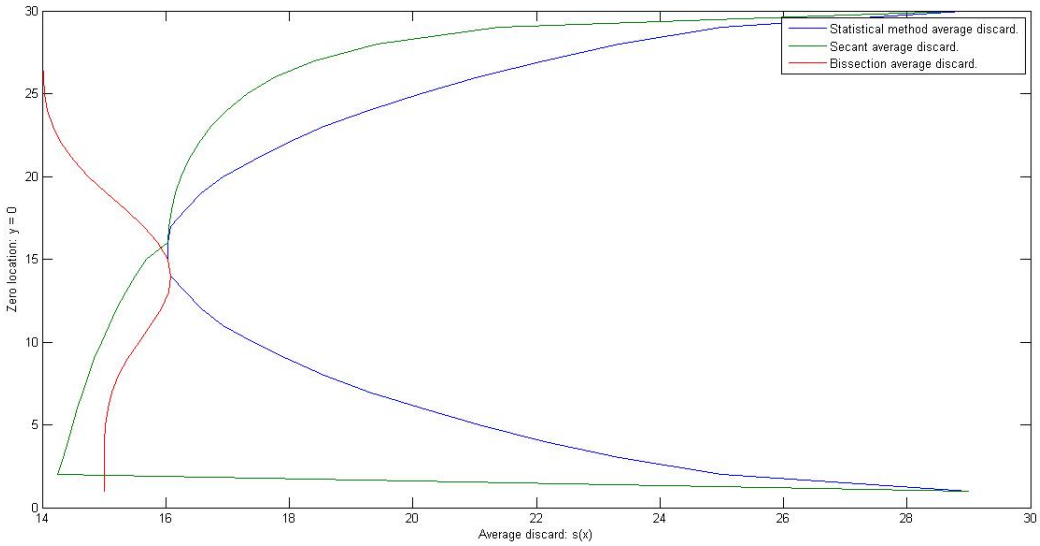


Fig. 3.5. Performance of Strategies with zero Varying from 1 to 30

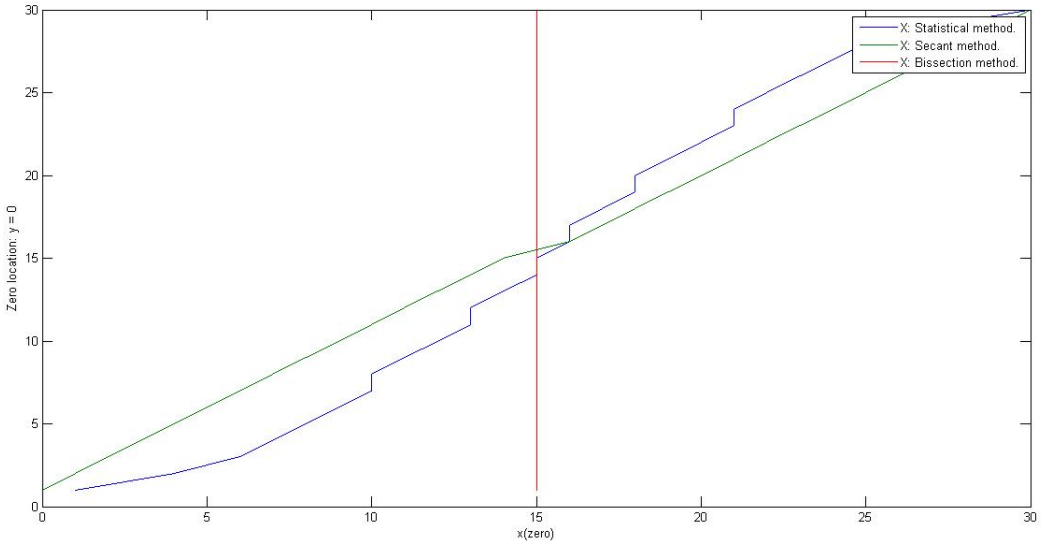


Fig. 3.6. Equivalent Root Problem, or Equivalent Interpolation

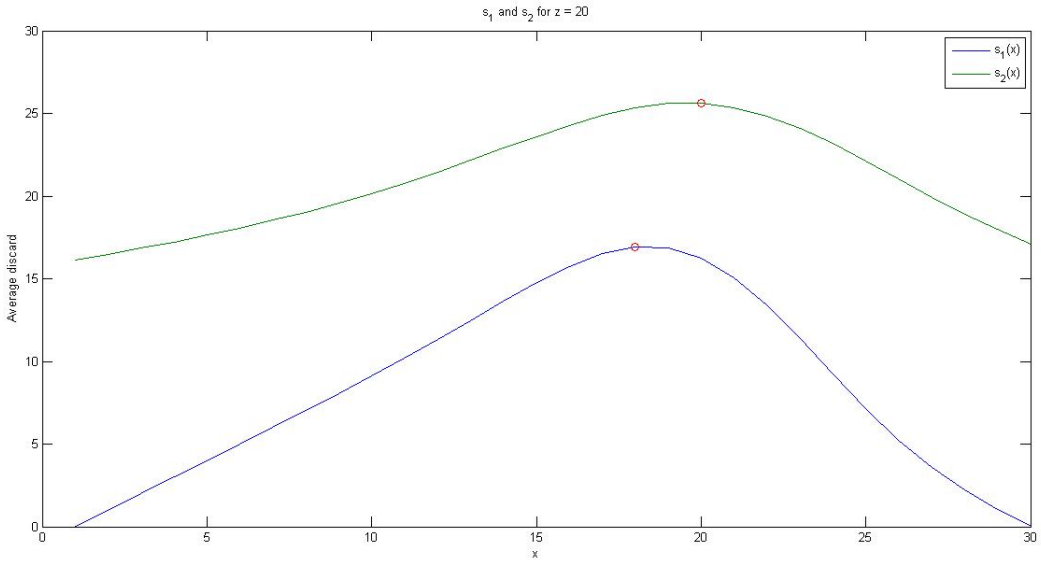


Fig. 3.7. $s_1(x)$ & $s_2(x)$ for zero = 20

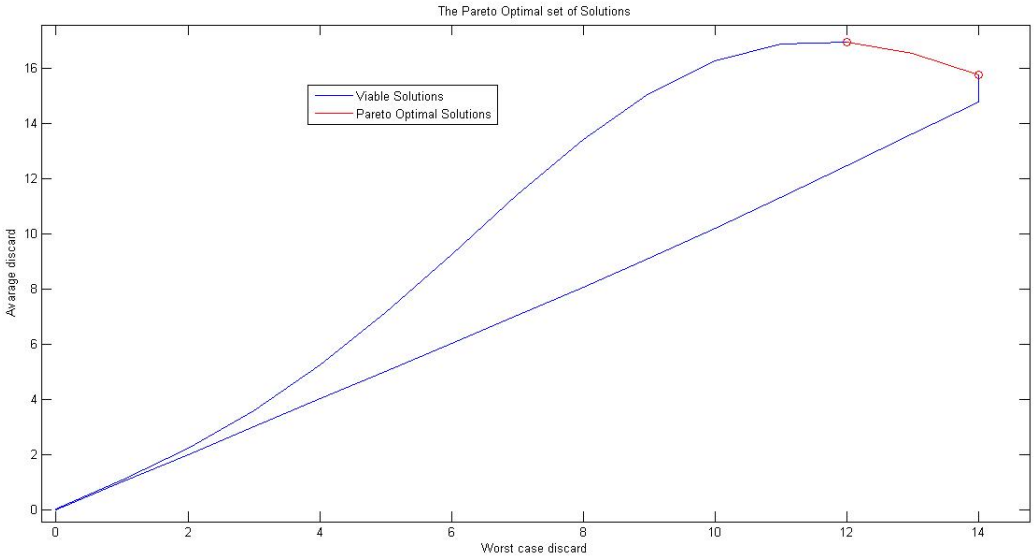


Fig. 3.8. Pareto set of strategies for zero = 20

In order to quantify the time necessary to run each of these steps of the constructed routine, the time for each step excluding the time necessary to plot each figure is displayed:

1. Construction of $\rho(x, y)$ and the estimation $N(m, \sigma^2)$ took: 0.175818 seconds
2. Construction of $s_1(x)$ for each value of zero $z = 1 : 30$ took: 0.146363 seconds
3. Construction of $s_2(x)$ for zero $z = 20$ took: 43.203490 seconds

Discussion

As can be observed in figures 3.1 and 3.2 the intuitive approximation suggested by variance proportional to distance to extremes is very precise(figure 3.1 has both $N(\rho^r, \sigma'^2)$ and $\rho(x, y)$ overlapped), $N(\rho^r, \sigma'^2)$ and $\rho(x, y)$ are almost indistinguishable in most points. The resemblance increases as the number of points n_p, n_q increase although close to the edges (1,1) & (n_p, n_q) the approximation continues rough. The standard deviation also presents a satisfactory approximation and is also better the greater n_p & n_q are. The results depicted in figures 3.1 to 3.3 resemble the consequences of the Central Limit Theorem just as the hypothesis of the set of functions resemble the hypothesis adopted by the theorem.

The approximation given by a variance proportional to distance can be extended to $\Re^n$(for optimization) and to problems where the construction of a statistical characterization is difficult to obtain. This approximation is both simple and intuitive although it can result in poor approximation when compared to the precise statistical characterization such as given by Theorem 2 (as in Example 1) therefore, when possible, a statistical characterization given by Theorem 2 should be preferred.

Figure 3.4 displays the set of curves given by $s_1(x)$ with zero varying from 1 to 30. As demonstrated by the constructed theory, evaluating Step 1 of the generic root searching algorithm at the maximum of $s_1(x)$ will yield maximum discard. An example of how to use figure 3.4 is: Suppose an instance has zero $z = 4$, by using figure 3.4 it is possible to identify the fourth curve (from the left to the right) and verify the maximum point of the curve $s_1(x)$ to evaluate step 1 with maximum average discard. Another use of figure 3.4 is to evaluate the statistical performance of a given strategy; to evaluate the bisection method, for example, the fourth curve evaluated at $x_{1/2}$ will give the average discard operation of the this method for $z = 4$. Figure 3.5 has the performance of each of the three evaluated strategies with zero varying from 1 to 30.

The statistical performance of each strategy depicted in figure 3.5 shows the statistical superiority of the statistical method. When z = 15 the three strategies coincide and therefore have an optimal discard. Figure 3.5 also shows the lack of symmetry of the average discard of the bisection method and the secant method. This lack of symmetry observed is probably explained by the non-symmetry of the set of non decreasing functions; for the graphic of a function admits horizontal lines, but not vertical lines; and probably the effects of integer approximations also interfere in the results, especially of the secant method. A symmetric performance is expected to be observed over the set of strictly increasing methods ($\rho(x, y)$ can be easily constructed for this set with very similar arguments used in this chapter).

Figures 3.6 and 3.7 bring quite interesting insights. The first shows how each strategy selects x in Step 1 for zero varying from 1 to 30. In fact the curve that represents the statistical method in figure 3.6 illustrates what can be used as an abacus pre-calculated to follow a statistically optimal sequence of steps in the root bracketing strategy. What isn't clear in the figure is that the statistical method tends to a smooth curve when the number of n_p & n_q grow, the indentations in its curve is partly due to numerical approximations that are still rough with $n_p = 30$ & $n_q = 30$. As discussed previously, a Pareto Optimal set of strategies is defined between the bisection method and the statistical method in each step, therefore evaluating last step situations figure 3.6 shows that the secant method is not Pareto Optimal. Figure 3.7 on the other hand shows that although in a two step evaluation $\bar{x}_2 \neq \bar{x}_1$, the average discard obtained by evaluating twice over maximum of $s_1(x)$ will result in almost maximum efficiency.

Another information contained in figure 3.7 is that $\bar{x}_2$ is closer to the secant method than $\bar{x}_1$. Intuitively (although far from proven) it is possible to say that although the secant method isn't Pareto Optimal the statistical method does seem to converge to it when the number of steps n grows in a similar way that the statistical method converges to the the secant method when n_p & $n_q \to \infty$(this was discussed at the end of section 3.2.1). If the statistical method indeed converges to the secant method for discrete non decreasing functions then a strong analogy can be done to the Fibonacci Method as it converges to the Golden Section Method in unidimensional optimization (this will be discussed in the next chapter).

At last figure 3.8 shows the set of viable solutions. In multi-objective theory mid term solutions tend to be preferable in convex sets and extremes tend to be preferred in concave sets. The set of optimal solutions (highlighted in red) is, as can be noticed, slightly convex(almost linear); therefore it is a set in which no solution should naturally be preferred and any strategy in between can be chosen without excessive losses in any objective.

This experiment shows that with a small list of tables it is possible implement with little computational effort optimal steps in root-searching. This can be of interest of mathematical programming platforms and of built-in routines of various software's that crave for computational efficiency. Even the explicit calculation of s_1 or s_2 are, as shown by the experiment, computationally viable and therefore the very construction of these functions and tables can be part of common routines that use root-searching algorithms.

4

Optimal Maximum Searching

"It is the glory of God to conceal a matter; to search out a matter is the glory of kings." Pv 25:2 NIV

Maximum Search

Maximum/minimum search or non-linear optimization algorithms are interactive sequential methods constructed to solve the following problem:

Maximization Problem. *Given* $f : \Re^n \to \Re$ *a black-box function. Maximize:* $f(x)$ *Subject to* $g_i(x) \leq 0$ *and to* $h_j(x) = 0 \mid i = 1, ..., n_g$ *and* $j = 1, ..., n_h$.

In optimization, function f is called the objective function. A minimum of hypothesis is commonly assumed over the black-box function f. In optimization, the sense of "best" strategy to solve a maximization problem is, as in other black-box sequential searching problems, understood as those that simultaneously minimize at a maximum rate the possible location of the solution with a minimum amount of evaluations of the objective function. The reader will soon notice the close relation there is between root-searching and maximum-searching methods due to this same nature of black-box sequential searching.

Line Search

Line Search strategies, have served as a backbone to various optimization methods [8]. These strategies are divided into two main steps.

1. Choose from a starting point a descent direction, defining line r.
2. Maximize function f over line r.

This second step of the Line Search methods is a unidimensional sub-problem known as unidimensional search. Thus unidimensional optimization strategies are of great importance in the optimization of multidimensional problems.

Although many years where dedicated to research on optimization methods, popular unidimensional methods where proposed from the beginning of the 1950's to the end of the 1970's. The Fibonacci method and the Golden Section method that are known for their optimality [3], Armijo and curve fitting methods, from which I highlight Brent's modified method[5, 10](that is originally used for root searching problems) are examples of the unidimensional optimization strategies born

in these three decades. These unidimensional methods, with minor modifications, have remained as the main tools in modern optimization algorithms.

Fibonacci's method, as well as the Golden Section method, stand for presenting a guarantee of a predictable convergence rate that does not depend on the nature of the objective function. As demonstrated by Kiefer [3] these methods are minimaximal; this minimaximality can be comprehended as being optimal in a worst case situation, in a sense similar to what the bisection method is in root-searching.

Parallel to this list of Line Search methods are the limited step methods such as the Trust Region methods [9, 2]. These methods may be identified as descendants of Levenberg-Marquardt algorithms that arose at the end of the 70's and have produced a great amount of literature till present date. Although the increasing importance of these Trust Region methods, because it is a limited step method and doesn't necessarily use concepts of discard and bracketing, these escape the scope of this work.

4.1 Classical Unidimensional Optimization

Now the unidimensional optimization problem and the classic non-randomized sequential maximum-finding algorithm with a bracketing strategy will be presented:

Unidimensional Maximization Problem. *Let* $f : [a, b] \to \Re$ *be a black-box function. Given* a, b *and* $c \in (a, b)$ *and* $y_a = f(a), y_b = f(b), y_c = f(c)$ *with* $y_c > (y_a \ \& \ y_b)$. *Find* x^* *that maximizes:* $f(x)$

Without loss of generality f may be assumed to be limited, for if f isn't, it is possible to construct $f' = \tan^{-1} \circ f$, and $\mathrm{Im}(f') \subset [-\frac{\pi}{2}, \frac{\pi}{2}]$ is limited and if x^* maximizes f then it maximizes f' as well. Function f is usually assumed to be uni-modal or continuous. Either way the existence of a solution is guaranteed, in the first case by definition and in the second case by Weierstrass Theorem.

The classic non-randomized sequential maximum-finding algorithm with a bracketing strategy is:

1. Chose $x \in (a, b) \mid x \neq c$
2. Evaluate $f(x)$
3. If $f(x) > y_c$ and $x > c$ make $a \leftarrow c$ and $c \leftarrow x$
   else if $f(x) > y_c$ and $x < c$ make $b \leftarrow c$ and $c \leftarrow x$
   else if $f(x) < y_c$ and $x > c$ make $b \leftarrow x$
   else if $f(x) < y_c$ and $x < c$ make $a \leftarrow x$
   else if $f(x) = y_c$ then $\to *$ [1]
4. If stopping criteria is met, return $\{a, y_a, b, y_b, c, y_c\}$ and stop.
   Else go to step 1.

[1] Different hypothesis over function f redound in different actions over the finding of $f(x) = y_c$. In the case of f being uni-modal with strictly increasing function for $x < x^*$ and strictly decreasing for $x > x^*$ then redefining the boundaries of $[a, b] \leftarrow [min(x, c), max(c, x)]$ and recalculating x and c is the best measure. Some hypothesis over f leave no alternative other than simply returning to step 1 and choosing a new value for x different from the previous.

It is imperative to say that many unidimensional optimization problems don't start with a, b, c, y_a, y_b and y_c values and in this case, the algorithm starts by choosing adequate values in f's domain to evaluate and assign the respective a, b, c, y_a, y_b and y_c values.

As in root-searching algorithms different strategies vary mostly in how to chose $x \in [a, b]$ and as can be imagined, interpolation is commonly used to predict the location of the maximum. Probably the most popular unidimensional maximization strategy is the Golden Section method. The Golden Section method is a particular case of the Fibonacci Sequence Method that is explained bellow:

4.1.1 Fibonacci Sequence Method (S_n^*)

Let U_i be the i'th Fibonacci number given by:

$$U_0 = 0; U_1 = 1; U_i = U_{i-1} + U_{i-2} \mid i \geq 2$$

And let ϕ_i^+ and ϕ_i^- be defined by:

$$\phi_i^+ = U_i/U_{i+1} \quad \& \quad \phi_i^- = U_{i-1}/U_{i+1} \mid i > 2$$

$$\phi_2^+ = \epsilon + U_2/U_{2+1} = \epsilon + \tfrac{1}{2} \quad \& \quad \phi_2^- = U_{2-1}/U_{2+1} = \tfrac{1}{2}$$

The Fibonacci Sequence method S_n^* with n steps supposes a starting condition of having only values of a and b and is defined by:

Starting with $i \leftarrow n$

1. Calculate $c = a + (b - a)\phi_i^+$ and $x = a + (b - a)\phi_i^-$
2. Execute steps 2-3 from the classic non-randomized sequential maximum-finding algorithm with a bracketing strategy.[2]
3. If $i > 2$ update $i \leftarrow (i - 1)$ and go to step 1.
   Else return $\{a, y_a, b, y_b, c, y_c\}$ and end.

As the reader may notice, although step 1 of the Fibonacci Sequence method calculates c and x, it only needs to evaluate c and x at the first iteration, where at the subsequent, the values of either c or x are already known due to the following property:

$$\phi_i^+ \times \phi_{i-1}^+ = \frac{U_i}{U_{i+1}} \times \frac{U_{i-1}}{U_i} = \frac{U_{i-1}}{U_{i+1}} = \phi_i^-$$

The Golden Section S_∞^* method is nothing more than the Fibonacci Sequence Method S_n^* defined by:

$$S_\infty^* \equiv \lim_{n \to \infty} S_n^*$$

At each step of S_n^* the value of ϕ_i^+ and ϕ_i^- must be updated, whereas S_∞^* uses the fixed value given by:

$$\phi = \lim_{n \to \infty} \phi_n^+ = \frac{\sqrt{5} - 1}{2} = 0,6180339887498948...$$

[2] If step 2 is being executed for the first time it is necessary to evaluate function value not only of $f(x)$ but also of $f(c)$.

$\mathscr{E}$

$$\phi^{-1} = 1 - \phi = \frac{3 - \sqrt{5}}{2} = 0,3819660112501052...$$

These two methods introduced by J.Kiefer [3] are mini-maximal over the set of the uni-modal functions with a maximum at x^* and strictly increasing from $[a, x^*)$ and strictly decreasing from $(x^*, b]$. More precisely, Kiefer's methods are, in his nomenclature:

Theorem 5. *Given any* $\epsilon > 0$ *and* $\forall n = 1, ..., \infty$*:*

$$\sup_{f \in F} L(D(f, S_n^*)) \leq \inf_{S \in \mathbb{S}_n} \sup_{f \in F} L(D(f, S)) + \epsilon$$

$\mathbb{D}$: set of all closed intervals within [a,b].
$D \in \mathbb{D}$: Terminal decision.
$n \in \mathbb{N}$: An integer.

Let $g_k : [a, b]^{k+1} \times \Re^{k+1} \to [a, b]$ be functions $\mid k = 1, ..., n$.
And s $\mathscr{E}$ $t : [a, b]^{n+2} \times \Re^{n+2} \to [a, b]$: be functions $\mid s \leq t$.
A strategy S_n, will be the set $S_n = \{a, b, y_a, y_b, g_1, ..., g_n, s, t\}$ that can be computed sequentially as the root searching algorithms:

$$x_k = g_k(a, b, x_1, ..., x_{k-1}, y_a, y_b, f(x_1), ..., f(x_{k-1})) \mid k = 1, ..., n$$
$$D(f, S) = [s(a, b, x_1, ..., x_n, y_a, y_b, f(x_1), ..., f(x_n)),$$
$$t(a, b, x_1, ..., x_n, y_a, y_b, f(x_1), ..., f(x_n))]$$

With $\mathbb{S}_n$ the set of strategies $\{S \mid x^* \in D(f, S) \forall f \in F$ where F is the set of non decreasing C^0 functions.$\}$ and L is the length function.

A full demonstration of this result is presented in J.Kiefer's [3] works.

4.1.2 Modified Bisection Method

Some modifications to the Bisection method where proposed for use in optimization, all of which, to the authors knowledge, have a performance necessarily worse than the Golden Section [4]. The variations of the modified bisection method seem to be preserved by literature in order exclusively to show superiority of the Golden Section method.

In this section an original modification to the Bisection method shall be presented. This variation presents a superior performance in sets of functions with particular characteristics as will be shown. This method's starting condition supposes that a, b, c, y_a, y_b and y_c are known, and if not, these values must be calculated to begin the following algorithm:

If $c = (a + b)/2$ then go to step 1 - A.
Else go to step 1 - B.

Modified Bisection Method - Instance A

1. If $y_a \geq y_b$ calculate $x = (a + c)/2$
   Else if $y_a < y_b$ calculate $x = (c + b)/2$
2. Execute steps 2-3 from the classic non-randomized sequential maximum-finding algorithm with a bracketing strategy.
3. If stopping criteria is met, return $\{a, y_a, b, y_b, c, y_c\}$ and end.
   Else if $c = (a + b)/2$ then go to step 1 - A.
   Else go to step 1 - B.

Modified Bisection Method - Instance B

1. If $\|(c - a)\| \geq \|(b - c)\|$ calculate $x = (a + c)/2$
   Else if $\|(c - a)\| < \|(b - c)\|$ calculate $x = (c + b)/2$
2. Execute steps 2-3 from the classic non-randomized sequential maximum-finding algorithm with a bracketing strategy.
3. If stopping criteria is met, return $\{a, y_a, b, y_b, c, y_c\}$ and end.
   Else if $c = (a + b)/2$ then go to step 1 - A.
   Else go to step 1 - B.

If the algorithm is initiated in instance A then the following iterations will necessarily either return to instance A or execute one iteration of instance B and return necessarily to A at the following iteration; this cycle will be called a steady state regime. If the algorithm starts at instance B with $c \neq a + \frac{1}{3}(b-a)$ and $c \neq a + \frac{2}{3}(b-a)$, then the algorithm will run sequentially n_t times in instance B until its first iteration in instance A, and from then on the algorithm will follow a steady state regime. Before the steady state regime is reached the cycle of n_t iterations at instance B will be called the transient regime.

The resulting convergence rate of this version of the modified Bisection method can be studied by a diagram in which each arrow represents an evaluation of the objective function and its respective discard operation. The number above each arrow is the convergence rate given by the discard operation and the number below represents the value of the probability p or $(1-p)$ of this operations occurrence. For simplicity probability p will be considered constant; this assumption is a scale invariance assumption that makes viable the analysis of the performance of the modified bisection method. The upper row represents instance A and the lower one instance B.

Diagram 1 - Bisection Steady State Regime

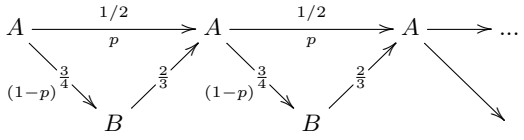
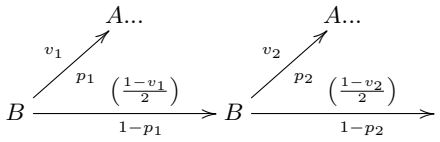


Diagram 2 - Bisection Transient Regime



Modified Bisection Convergence Rate

The above diagrams serves of aid to calculate the convergence rate of the modified Bisection method during the steady state regime based on probability p. A more thorough analysis may be done including a similar convergence rate calculus with the transient regime although the reader may notice that in cases where the initial conditions don't give c and y_c, they may be chosen as to start the algorithm in the steady state regime, being in this case, the analysis complete:

Theorem 6. *Given p, the probability at which the bisection method enters instance A from instance A, the value of the average convergence rate of the modified bisection method is given by:*

$$r_{\frac{1}{2}} = \left(\frac{1}{2}\right)^{\frac{1}{2-p}}$$

Proof. The average convergence rate of a strategy $S_n \in \mathbb{S}_n$ is defined as the number $r \in \Re \mid$ the new length of the region $[a, b]$ after n iterations is equal to $r^n \times (b-a) = (b-a) - s_n$.

This way, supposing the algorithm executed n iterations over instance A, then $p \times n$ times it whet directly from instance A to instance A and $(1-p) \times n$ times it went through instance B before. This results in the convergence rate given by:

$$r_{\frac{1}{2}} = \left[\left(\frac{1}{2}\right)^{p} \times \left(\frac{3}{4}\right)^{1-p} \times \left(\frac{2}{3}\right)^{1-p}\right]^{\frac{1}{p\times 1+(1-p)\times 2}}$$

$$= \left[\left(\frac{1}{2}\right)^{p} \times \left(\frac{1}{2}\right)^{1-p}\right]^{\frac{1}{p+2-2p}} = \left(\frac{1}{2}\right)^{\frac{1}{2-p}}$$

□

The convergence rate of the Golden Section method is of $\phi = (\sqrt{5}-1)/2 = 0.6180...$ so for any set of functions with $p \geq 2 + \log_{\phi}(2) = 0.5596...$ the Modified Bisection method will have in average a faster convergence rate $r_{\frac{1}{2}} = \left(\frac{1}{2}\right)^{\frac{1}{2-p}} \leq \phi$ than the Golden Section method. In other words, the more favourable function set $\mathbb{F}$ is in keeping the algorithm running from instance A directly to instance A, the better the convergence rate of the Modified Bisection method is to evaluate a function $f \in \mathbb{F}$ chosen at random. It may be useful to remember more precisely what probability p is before the end of this section:

Given a set $\mathbb{F}$ of functions, $f \in \mathbb{F}$ a function chosen at random and evaluated at three points a, b and c with $c = (a+b)/2$, and supposing without loss of generality that $f(b) > f(a)$. Then p is the probability that function f has function value $y(x) > y(c)$ at $x = (b+c)/2$.

This "inertia" condition is perfectly possible in sets of functions that x function values carry information of the location of neighbour values such as in differentiable functions. Function sets that have tendency to have maximum at extremes will also be favoured by this method over the Golden Section method.[3]

Sub-Optimality of the Modified Bisection Method

As discussed in the previous section, the Fibonacci sequence is a mini-maximal algorithm with a starting condition in which c and y_c isn't given over the set of uni-modal functions. One may ask how to obtain a method that is mini-maximal over the same set of functions but with a starting condition in which a, b and c and y_a, y_b and y_c are given.

At first all mini-maximal strategies $S_1 \in \mathbb{S}_1$ with given a, b, c, y_a, y_b and y_c will be analysed. As usual the following assumptions are made: $y_c > y_a$ and $y_c > y_b$.

As in root searching mini-maximal strategies "minimize" [4] objective function $\underline{O}_n$ defined by:

$$\underline{O}_n : \mathbb{S}_n \to \Re \mid \underline{O}_n(S_n) = \sup_{f \in F} L(D(f, S))$$

If $c > \frac{1}{2}(a+b)$ it is easy to perceive that any strategy $S_1 \in \mathbb{S}_1$ that evaluates $x \in [a+b-c, c)$ is then mini-maximal. If $c < \frac{1}{2}(a+b)$ the argument is analogous, but if $c = \frac{1}{2}(a+b)$ then the Fibonacci Sequence strategy S_1^* is the "minimizer" of $\underline{O}_n$. It is possible to notice that $\underline{S}_1$ that minimizes $\underline{O}_1$ isn't unique, this is also true for $\underline{O}_n$ and the number of "minimizers" of $\underline{O}_n$ are infinite $\forall n \in \mathbb{N}$.

Considering once more the case when $c > \frac{1}{2}(a+b)$, the set of minimizing strategies of $\underline{O}_1$ can evaluate a randomly selected function f and execute the discard operation over either $[a, x]$ or $[c, b]$. The second possibility is the same for all minimizers, but evidently $l[a, x]$ varies with x and can be chosen to be δ maximum, this is achieved with $x = c - \delta$. A first version of the bisection method for optimization(that is not the case of the presented method) executes consecutive evaluations of the objective function over $c = (a+b)/2$ and $x = c - \delta$ which is to say very similar to what could be called a reinitialization of a $\underline{S}_1$ method with δ maximum possible discard operations.

What is interesting about the presented modified bisection method is that it is easy to demonstrate that at every step it evaluates the objective function f at x given by $\underline{S}_2$ that "minimizes" $\underline{O}_2$. This way while the classical bisection method for

[3] The reader may verify that some subsets of the polynomials have such "inertia". As an example of such subsets the author suggests a numerical verification of the Modified Bisection method between the first two roots of randomly generated tenth degree polynomials $P(x) = \sum c_i x^i$ by uniformly generating coefficients $c_i \in [-1, 1] \mid i = 1, ..., 9$ and $c_0 \in [0, 1]$ and $c_{10} \in [-1, 0]$.

[4] "minimize" is between quotation marks because as in Kiefer's demonstrated situation, there isn't a minimizer in strict sense. Kiefer's theorem guarantees that $\exists S^* \mid \sup_{f \in F} L(D(f, S_n^*)) \leq \inf_{S \in \mathbb{S}_n} \sup_{f \in F} L(D(f, S)) + \epsilon$

optimization is a reinitialization of a strategy $\underline{S}_1$ that minimizes $\underline{O}_1$, the modified bisection method is a reinitialization of a $\underline{S}_2$ that minimizes $\underline{O}_2$.

Exercise 6. *Find the set of strategies that minimize $\underline{O}_3$ with given initial values of a, b, c, y_a, y_b and y_c . (Hint: What is S_3^*?)*

For a better illustration on the modified bisection performance a numerical experiment shall be constructed at the end of this chapter to compare the efficiency of this method with the efficiency of the golden section method.

4.2 Statistical Performance

4.2.1 Statistical Optimization Method

Statistical Optimization is very similar to Statistical Root searching. As discussed in the Statistical Performance section of root searching methods, to obtain a statistically optimal convergence over a determined set of function $\mathbb{F}$ it is necessary to construct a Statistical Characterization of the set of functions being investigated.

In root search two main theorems where demonstrated to obtain the characterization of different sets of functions. Theorem 2 gives the means to characterize sets of functions $\mathbb{F} = \{f_q(x) = g(q, x) \mid q \in [0, 1]^p \subset \Re^p\}$ while Theorem 3 gives the characterization of the set of discrete non decreasing functions. In both Optimization and in Root Search Theorem 2 can be used for any set of functions that may be described by parameters $q \in \Re^n$. On the other hand Theorem 3 has little value in optimization, since non decreasing functions have a pre-determined location of the maximum. For optimization it is of more interest the characterization of uni-modal functions with constraints of $f(a) = y_a, f(b) = y_b$ and $f(c) = y_c$ since the Fibonacci sequence is mini-maximal over this set, and at each step of the classical non-randomized bracketing strategy y_a, y_b and y_c are known. This way the performance of these two methods can be compared.

To construct the characterization of uni-modal functions the following lemma will be necessary:

Lemma 5. *The number of (strictly) increasing functions from $A = \{1, ..., n_p\}$ to $B = \{1, ..., n_q\}$ with $n_q \geq n_p$ is given by :*

$$\#f\Big\Uparrow_{\{1,...,n_p\}}^{\{1,...,n_q\}} = \#solutions\{c_1 + c_2 + ... + c_{n_p+1} = n_q - n_p \mid c_i \in \mathbb{N}^*\} = \frac{n_q!}{(n_q - n_p)!n_p!}$$

Proof. Given a solution function f, without loss of generality it is possible to consider that $f(0) = 0$ and $f(n_p + 1) = n_q + 1$. Let $c_i^* = f(i) - f(i-1) \mid i = 1, ..., n_p + 1$, so $c_1^* + c_2^* + ... + c_{n_p+1}^* = n_q + 1$ for each increasing function f and $c_i^* \in \mathbb{N}$ for f is strictly increasing. Define $c_i = c_i^* - 1$, therefore $c_1 + c_2 + ... + c_{n_p+1} = n_q - n_p$ and $c_i \in \mathbb{N}^*$ this implies that for each increasing function f defined from $\{1, ..., n_p\}$ to $\{1, ..., n_q\}$ there is a unique set of values for c_i satisfying $c_1 + c_2 + ... + c_{n_p+1} = n_q - n_p$, and for each solution to $c_1 + c_2 + ... + c_{n_p+1} = n_q - n_p \mid c_i \in \mathbb{N}^*$ there is a unique increasing function defined from $\{1, ..., n_p\}$ to $\{1, ..., n_q\}$.

Therefore the first equality of the Lemma yields.

The second equality is a classic combinatorics analysis result of which a demonstration may be found in [6] or in most books in combinatorics. □

Lemma 6. *The number of uni-modal functions defined from* $\{1,...,n_p\}$ *to* $\{1,...,n_q\}$ *with* $f(1)=y_1$ *is given by:*

$$\#f\wedge_{\{1,...,n_p\}}^{\{1,...,n_q\}}\Big|_{f(1)=y_1} = \#f\Big\Uparrow_{\{2,...,n_p\}}^{\{1,...,y_1-1\}} + \sum_{\substack{x=2,...,n_p\\ y=y_1+1,...,n_q}} \left(\#f\Big\Uparrow_{\{2,...,x-1\}}^{\{y_1+1,...,y-1\}}\right)\left(\#f\Big\Uparrow_{\{x+1,...,n_p\}}^{\{1,...,y-1\}}\right)$$

Proof. The number of uni-modal functions with $f(1)=y_1$ and $(x,y)=$ maximum of function, is given by $\left(\#f\Big\Uparrow_{\{2,...,x-1\}}^{\{y_1+1,...,y-1\}}\right)\left(\#f\Big\Uparrow_{\{x+1,...,n_p\}}^{\{1,...,y-1\}}\right)$. Therefore summing all possible locations of the maximum $x=2,...,n_p$ and $y=y_1,...,n_q$ the lemma is obtained. □

Theorem 7. *The Statistical Characterization of the set of uni-modal functions defined from* $\{1,...,n_p\}$ *to* $\{1,...,n_q\}$ *with* $f(1)=y_1$ *and* $f(c)=y_c>y_1$ *is given by:*

$$\rho(x,y)=$$

$$\left(\#f\Big\Uparrow_{\{2,...,c-1\}}^{\{y_1+1,...,y_c-1\}}\times\#f\wedge_{\{c,...,n_p\}}^{\{1,...,n_q\}}\Big|_{f(c)=y_c}+\#f\wedge_{\{2,...,c\}}^{\{y_1+1,...,n_q\}}\Big|_{f(c)=y_c}\times\#f\Big\Uparrow_{\{c+1,...,n_p\}}^{\{1,...,y_c-1\}}\right.$$
$$\left.-\#f\Big\Uparrow_{\{2,...,c-1\}}^{\{y_1+1,...,y_c-1\}}\times\#f\Big\Uparrow_{\{c+1,...,n_p\}}^{\{1,...,y_c-1\}}\right)^{-1}\delta_q^{-1}\times$$

$$\begin{cases}
\#f\Big\Uparrow_{\{2,...,x-1\}}^{\{y_1+1,...,y-1\}}\left(\#f\Big\Uparrow_{\{x+1,...,c-1\}}^{\{y+1,...,y_c-1\}}\times\#f\wedge_{\{c,...,n_p\}}^{\{1,...,n_q\}}\Big|_{f(c)=y_c}+\#f\wedge_{\{x+1,...,c\}}^{\{y+1,...,n_q\}}\Big|_{f(c)=y_c}\times\#f\Big\Uparrow_{\{c+1,...,n_p\}}^{\{1,...,y_c-1\}}\right) & \text{if } x<c \;\&\; y<y_c \;\&\; y>y_1\\
\#f\Big\Uparrow_{\{c+1,...,n_p\}}^{\{1,...,y_c-1\}}\left(\#f\Big\Uparrow_{\{2,...,x-1\}}^{\{y_1+1,...,y-1\}}\times\#f\wedge_{\{x,...,c-1\}}^{\{y_c+1,...,n_q\}}\Big|_{f(x)=y}+\#f\wedge_{\{2,...,x\}}^{\{y_1+1,...,n_q\}}\Big|_{f(x)=y}\times\#f\Big\Uparrow_{\{x+1,...,c-1\}}^{\{y_c+1,...,y-1\}}\right) & \text{if } x<c \;\&\; y>y_c\\
\#f\Big\Uparrow_{\{x+1,...,n_p\}}^{\{1,...,y-1\}}\left(\#f\Big\Uparrow_{\{2,...,c-1\}}^{\{y_1+1,...,y_c-1\}}\times\#f\wedge_{\{c,...,x-1\}}^{\{y+1,...,n_q\}}\Big|_{f(c)=y_c}+\#f\wedge_{\{2,...,c\}}^{\{y_1+1,...,n_q\}}\Big|_{f(c)=y_c}\times\#f\Big\Uparrow_{\{c+1,...,x-1\}}^{\{y+1,...,y_c-1\}}\right) & \text{if } x>c \;\&\; y<y_c\\
\#f\Big\Uparrow_{\{2,...,c-1\}}^{\{y_1+1,...,y_c-1\}}\left(\#f\Big\Uparrow_{\{c+1,...,x-1\}}^{\{y_c+1,...,y-1\}}\times\#f\wedge_{\{x,...,n_p\}}^{\{1,...,n_q\}}\Big|_{f(x)=y}+\#f\wedge_{\{c+1,...,x\}}^{\{y_c+1,...,n_q\}}\Big|_{f(x)=y}\times\#f\Big\Uparrow_{\{x+1,...,n_p\}}^{\{1,...,y-1\}}\right) & \text{if } x>c \;\&\; y>y_c
\end{cases}$$

Lemma 7. *The average length of the discarded region in one step of evaluation of function* f *in the generic optimization algorithm in point* x *is:*

$$s^1(x)=\int_{y_a}^{y_b}\rho(x,y)l(x,y)\mathrm{d}y$$

Where:

$$l(x,y)=\begin{cases}(b-c)\mid for\ y>y_c\ \&\ x<c\\(c-a)\mid for\ y>y_c\ \&\ x>c\\(x-a)\mid for\ y<y_c\ \&\ x<c\\(b-x)\mid for\ y<y_c\ \&\ x>c\end{cases}$$

Proof. Step 3 of the bracketing strategy will perform at each iteration of the algorithm the following discard operations:

For $x < c$

1. If $f(x) > y_c$ discard interval $[c, b]$
2. If $f(x) = y_c$ discard interval $[a, x] \cup [c, b]$
3. If $f(x) < y_c$ discard interval $[a, x]$

Therefore the average length of the discarded region is given by:

$$s^1(x) = P[f(x) > y_c] \times (b-c) + P[f(x) = y_c] \times [(x-a)+(b-c)] + P[f(x) < y_c] \times (x-a)$$

Using the definition of a probability distribution and integrals in discrete spaces the result is obtained.

For $x > c$ the demonstration is analogous. □

Therefore, if $\bar{S}_1$ is the strategy that in one step the greatest discard operation is done in average, then $\bar{S}_1$ shall evaluate function f in x_1:

$$x_1 = argmax_{x \in [a,b]} s^1(x)$$

Theorem 8. *$\bar{S}_n \in \mathbb{S}_n$ is given by a sequence of evaluations of function f on points $x_n, ..., x_1$ given by:*

$$x_i = argmax_{x \in [a,b]} s^i(x)$$

Where a and b are updated at each iteration given by step 3 of the generic bracketing strategy and $s^i(x)$ for $i = 2, ..., n$ is given by:

$$s^i(x) = \int_{y_a}^{y_b} \rho(x, y) \left[l(x, y) + \max_{\alpha \in [a^*, b^*]} \left(s^{i-1}(\alpha) \right) \right] \mathrm{d}y$$

The term s^{i-1} in the above equation is evaluated in $[a^*, b^*]$, that is the updated interval of function f when $f(x) = y$, and $l(x)$ is the same as given by lemma.

Proof. When there are i steps left for evaluation, x_i must be chosen to maximize the average sum of the discard operation given by that step $l(x, y)$ plus the average discarded quantity for the next $i - 1$ steps. Once the strategy of choice of x_k is always to evaluate at maximum point of s^k, the average discarded quantity for the next $i-1$ iterations is given by $\max_{\alpha \in [a^*, b^*]} \left(s^{i-1}(\alpha) \right)$ where $[a^*, b^*]$ is the updated interval by step 3 of the bracketing strategy given by the current function interval $[a, b]$ when evaluated at x and obtaining $f(x) = y$. □

As in root-searching, the value of ρ, s^i and x_i for some sets of functions may be pre-calculated in order to avoid expensive calculations of the updated values of these parameters. For optimization, sets that may be considered for these pre-calculations are the set of discrete uni-modal functions and n^{th} degree polynomials, with n even, for example.

Now a combinatorial result that may prove itself useful to count and compare uni-modal functions to increasing functions is presented:

Lemma 8. *The number of uni-modal functions defined from* $\{1, ..., n_p\}$ *to* $\{1, ..., n_q\}$ *is given by:*

$$\#f\Big\Uparrow_{\{1,...,n_p\}}^{\{1,...,2n_q\}} = 2\#f \wedge_{\{1,...,n_p\}}^{\{1,...,n_q\}} + \#f \wedge_{\{1,...,n_p-1\}}^{\{1,...,n_q\}}$$

Proof. Let $\mathbb{C} = \{c : \{1, ..., n_p\} \to \{1, ..., 2n_q\} | c \text{ is an increasing function}\}$ and $\mathbb{U} = \Big\{u : \{1, ..., n_p\} \to \{1, ..., n_q\}\Big\}$

Consider the following reflection operation:

$$\lambda : \mathbb{C} \to \mathbb{U}$$

$$u(i) = \lambda(c(i)) = \begin{cases} c(i) & | \text{ if } c(i) \leq n_q \\ 2n_q - c(i) & | \text{ if } c(i) > n_q \end{cases}$$

This way for every increasing function in $\mathbb{C}$ by λ a uni-modal function in $\mathbb{U}$ or a bi-modal function with two neighbouring maxima in $\mathbb{U}$ is obtained. It is possible to find by studying λ that for every uni-modal function u in $\mathbb{U}$ there are two increasing functions c_1 and c_2 in $\mathbb{C}$ that $\lambda(c_i) = u \mid i = 1, 2$ and that for every bi-modal function b in $\mathbb{U}$ with neighbouring maxima there will be an increasing function c_3 in $\mathbb{C}$ with $\lambda(c_3) = b$.

For each function $u \in \mathbb{U} \mid u$ is uni-modal, let x^u be maximum of u; and for each function $b \in \mathbb{U} \mid b$ is bimodal with neighbouring maxima, let x^b be the first maximum and $x^b + 1$ be the second. Then c_1 , c_2 and c_3 are given by:

$$c_1(x) = \begin{cases} u(x) & | \text{ if } x < x^u \\ 2n_q - u(x) & | \text{ if } x \geq x^u \end{cases}$$

$$c_2(x) = \begin{cases} u(x) & | \text{ if } x \leq x^u \\ 2n_q - u(x) & | \text{ if } x > x^u \end{cases}$$

$$c_3(x) = \begin{cases} u(x) & | \text{ if } x \leq x^b \\ 2n_q - u(x) & | \text{ if } x > x^b \end{cases}$$

Therefore λ is a bijection between $\mathbb{C}$ and $\{f \in \mathbb{U} \mid f \text{ is either uni-modal or bi-modal with neighbouring maxima}\}$. It is easy to notice that the set of bi-modal functions with neighbouring maxima defined from $\{1, ..., n_p\}$ to $\{1, ..., n_q\}$ is almost identical to uni-modal functions defined from $\{1, ..., n_p - 1\}$ to $\{1, ..., n_q\}$ and are equal in size; this proves the lemma. □

4.2.2 Statistical Method x Mini-Maximal Method, a Pareto Set

In root search the existence of a Pareto optimal set of bracketing strategies was discussed. At this point the reader must have noticed the strong similarity between root searching and unidimensional optimization. In order to avoid repeated effort to discuss a matter that varies little between these two families of problems only a small remark onto the validity of the discussion presented in the former chapter is made.

As in root search it is possible to define objective functions $\underline{O}_n$ and $\overline{O}_n$. As naturally as the Pareto optimal set comes from the vector optimization problem of minimizing $O = \left(\underline{O}, \overline{O}\right)$ in root searching, the same set arises in unidimensional optimization with a slight observation. The observation is the same as discussed in the optimality of the Fibonacci Sequence method, that is, that optimality is understood as belonging to the set of δ optimal solutions.

With the proper care to consider this remark on what is understood to be δ optimal the conclusions that arise from analysing the Pareto set of optimal optimization strategies follow a similar path as that of root searching strategies.

4.3 Numerical Experiments on Maximum-Searching

This section will exemplify the constructed theory by comparing the performance of numerical implementations (a computational algorithm) of the statistically optimal maximum-searching method and the modified bisection method to the mini-maximal maximum-searching strategy. For this two experiments were elaborated. The first experiment was constructed to evaluate the convergence rate of the modified bisection method over two distinct sets of functions and compare it with the predictable convergence rate of 0.618 given by the golden section method. The second experiment evaluates a Fibonacci $\underline{S}_2$ initial condition of $a, b, c = a + \frac{1}{3}(b-a)$ and $f(a), f(b)$ and $f(c)$ known and compares the the predictable performance of the Fibonacci method to a numerical implementation of the statistical $\overline{S}_2$ method.

4.3.1 Modified Bisection Method Performance Over Two Sets

Methodology:

The average convergence rate of the modified bisection method was evaluated over two distinct sets of functions. The fist is the set of tenth degree polynomials $P(x) = \sum c_i x^i$ with coefficients $c_i \in [-1, 1] \mid i = 1, ..., 9$ and $c_0 \in [0, 1]$ and $c_{10} \in [-1, 0]$; these coefficients where selected randomly with uniform probability and the domain of the function was selected to be the interval between the first two real roots(therefore guaranteeing uni-modality). The second set was chosen with the opposite principle of the first, for it has a fixed function and a varying interval; for this set the cosine function was chosen and the interval was selected randomly in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ for each iteration.

The routine was constructed using Matlab 7.10.0(R2010a) and for each of the chosen sets the algorithm sequentially chooses a function within the set and evaluates the convergence rate[5] of the modified bisection strategy after solving the maximization problem defined over the selected function with a stopping criteria of bracketing the solution within an error of 10^{-5} units. The same problem is not solved using the classical bisection strategy neither the golden section strategy for their constant convergence rate is already known and are given below. At each iteration, defined by one randomly selected function, the average convergence rate is updated and

[5] The convergence rate is defined as $\sqrt[n]{l_f/l_o}$ where n is the number of function evaluations, l_f and l_o are the final and initial lengths of the bracketing strategy.

another function is then drawn once again until five consecutive calculations of the average remain within a difference of 10^{-5} (stagnation criterion); this process is programmed to repeat 3 times and the same stagnated average is expected for the three processes.

As reference the following nomenclature will be adopted:

1. Convergence rate of the golden section strategy: $r_\phi = 0.6180$
2. Convergence rate of the classical bisection strategy: $r_{1/2} = 0.7071$
3. Convergence rate of the modified bisection strategy: $r^*_{1/2}$

The code of the routine may be found in the addendum to this work.

Results

Experiment 1.

For the set of 10^{th} degree polynomials:

1^{st} **Run.**

1. Number of functions evaluated before stagnation: 10344
2. Average convergence rate obtained: 0.5667
3. Proportion of functions with $r^*_{1/2} \leq r_\phi$: 0.6257

2^{nd} **Run.**

1. Number of functions evaluated before stagnation: 9992
2. Average convergence rate obtained: 0.5672
3. Proportion of functions with $r^*_{1/2} \leq r_\phi$: 0.6221

3^{rd} **Run.**

1. Number of functions evaluated before stagnation: 9088
2. Average convergence rate obtained: 0.5681
3. Proportion of functions with $r^*_{1/2} \leq r_\phi$: 0.6166

Overall best convergence rate obtained: $r^*_{1/2} = 0.5000 = 0.8090 r_\phi = 0.7071 r_{1/2}$
Overall average convergence rate obtained: $r^*_{1/2} = 0.5673 = 0.9179 r_\phi = 0.8023 r_{1/2}$
Overall worst convergence rate obtained: $r^*_{1/2} = 0.6944 = 1.1236 r_\phi = 0.9820 r_{1/2}$

Experiment 2.

For the set of cosine functions:

1^{st} **Run.**

1. Number of functions evaluated before stagnation: 9644
2. Average convergence rate obtained: 0.5654
3. Proportion of functions with $r^*_{1/2} \leq r_\phi$: 0.6468

2^{nd} **Run.**

1. Number of functions evaluated before stagnation: 8855

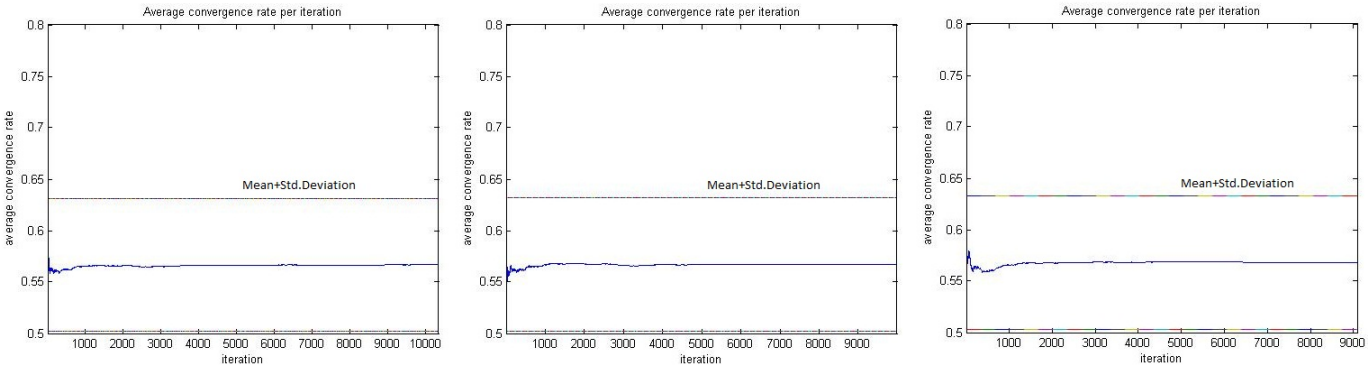


Fig. 4.1. First, Second and Third Run, Modified Bisection Method on 10^{th} Degree Polynomials

2. Average convergence rate obtained: 0.5660
3. Proportion of functions with $r^*_{1/2} \leq r_\phi$: 0.6407

3^{rd} **Run.**

1. Number of functions evaluated before stagnation: 10329
2. Average convergence rate obtained: 0.5658
3. Proportion of functions with $r^*_{1/2} \leq r_\phi$: 0.6413

Overall best convergence rate obtained: $r^*_{1/2} = 0.5000 = 0.8090 r_\phi = 0.7071 r_{1/2}$
Overall average convergence rate obtained: $r^*_{1/2} = 0.5657 = 0.9154 r_\phi = 0.8000 r_{1/2}$
Overall worst convergence rate obtained: $r^*_{1/2} = 0.7071 = 1.1441 r_\phi = 1.0000 r_{1/2}$

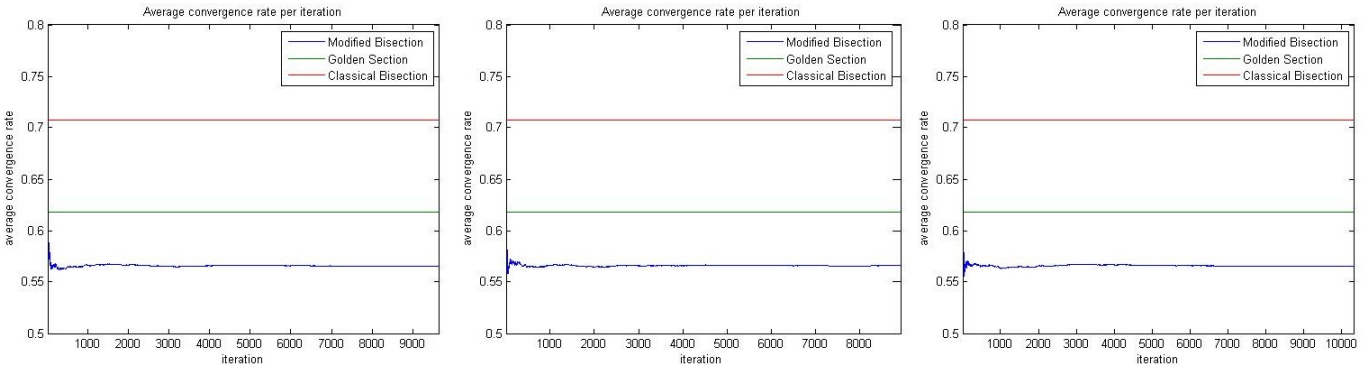


Fig. 4.2. First Run, Modified Bisection Method on Cosine Functions

Discussion

Both experiment 1 and experiment 2 show similar results. The modified bisection method presents an average convergence rate close to 0.566 that is clearly better that the golden section method in a little less than 10%. It's worst case performance presented a value very close to, but slightly higher than, the convergence rate of the classical bisection method and a possibility of achieving a maximum convergence rate of 0.5.

Due to the simplicity of the modified bisection method a computational implementation of the algorithm presents little or no complication and can be compared to the simplest algorithms in terms of facility.

The modified bisection method can vary in performance according to the set of functions being evaluated but as shown by this experience and by theory the value of the convergence rate is safe above a minimum value. In addition the modified bisection method has an expected higher average according to the probability of prediction that the contour conditions give of the location of the maximum. This statistical performance that has shown to be better than that of the golden section method is better understood under the constructed theory of statistical characterization of functions. As discussed in the Pareto optimality section, the gain in average performance implies in loss of worst-case performance and vice-versa.

4.3.2 Statistical Performance Over Unimodal Discrete Functions

In order to study the performance of the statistical method for unidimensional optimization problems constructed in this chapter two numerical experiments will be presented.

The first numerical experiment is exploratory and qualitative. It constructs the statistical characterization for 3 selected conditions and graphically compares the projection of the contour of ρ and the curve given by a polynomial interpolation. This experiment's objective is also to find possible numerical obstacles that are not contemplated in the constructed theory.

The second experiment is quantitative. It constructs s_2 for an initial condition of $a, b, c = a + \frac{1}{3}(b-a)$ and $f(a), f(b)$ and $f(c)$ known (the same as Fibonacci $\underline{S}_2$ initial condition). This way the statistical method may and will be compared to the mini-maximal method.

Methodology:

The three selected conditions for the first experiment are:

Condition 1.
$n_p = 21; n_q = 50; f(1) = 3; c = 11; f(c) = 30$
Condition 2.
$n_p = 30; n_q = 30; f(1) = 5; c = 13; f(c) = 19$
Condition 3.
$n_p = 30; n_q = 30; f(1) = 10; c = 13; f(c) = 19$

The image of ρ for condition 1 will be displayed[6] and the expectation that $\sum_{j=1}^{n_q} \rho(i_0, j) = 1$ for $i_0 = 1, ..., n_p$ will be verified graphically.

The projection of ρ's contour for conditions 2 and 3 will be compared graphically to the classical second degree polynomial interpolation.

The second experiment has the following conditions (observe that $c = 1 + \frac{1}{3}(n_p - 1)$ and not $c = \frac{1}{3}n_p$ because $f(1)$ is known.):

Condition 4.

$n_p = 22; n_q = 40; f(1) = 15; c = 8; f(c) = 25$

For this initial condition s_1 and s_2 will be displayed and so will ρ. Posteriorly the graphics will be interpreted and so the performance of the Fibonacci sequence method $\underline{S}_2$ will be compared to the statistical method $\overline{S}_2$.

[6] For convenience $\rho(c, f(c))$ and $\rho(1, f(1))$ will be displayed as zero(their actual value is 1) to facilitate the visualization of the image.

Results

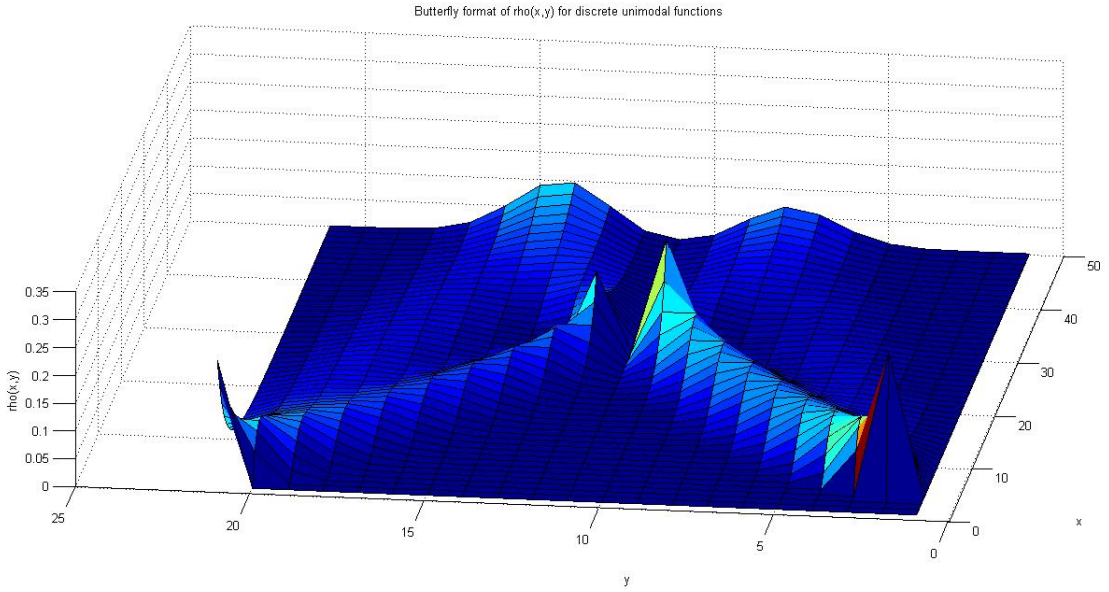


Fig. 4.3. Condition 1, Uni-modal Functions Butterfly Format

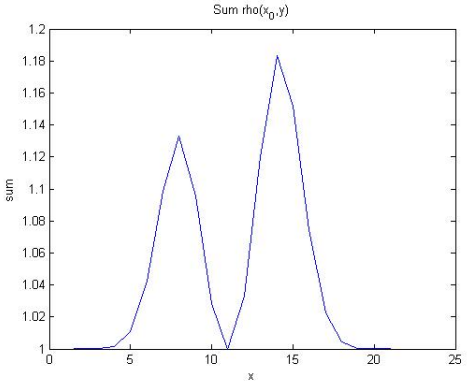


Fig. 4.4. Verification of $\sum_{y=1}^{n_q} \rho(x_0, y) = 1$ for $x_0 = 1, ..., n_p$

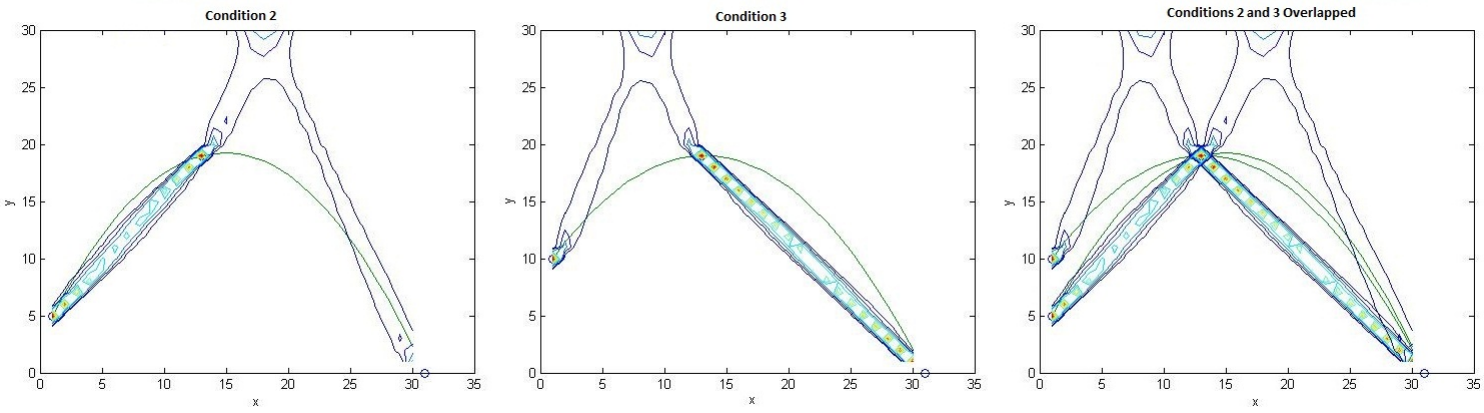


Fig. 4.5. Conditions 2 and 3, Contour of ρ of Uni-modal Functions and Polynomial Interpolation

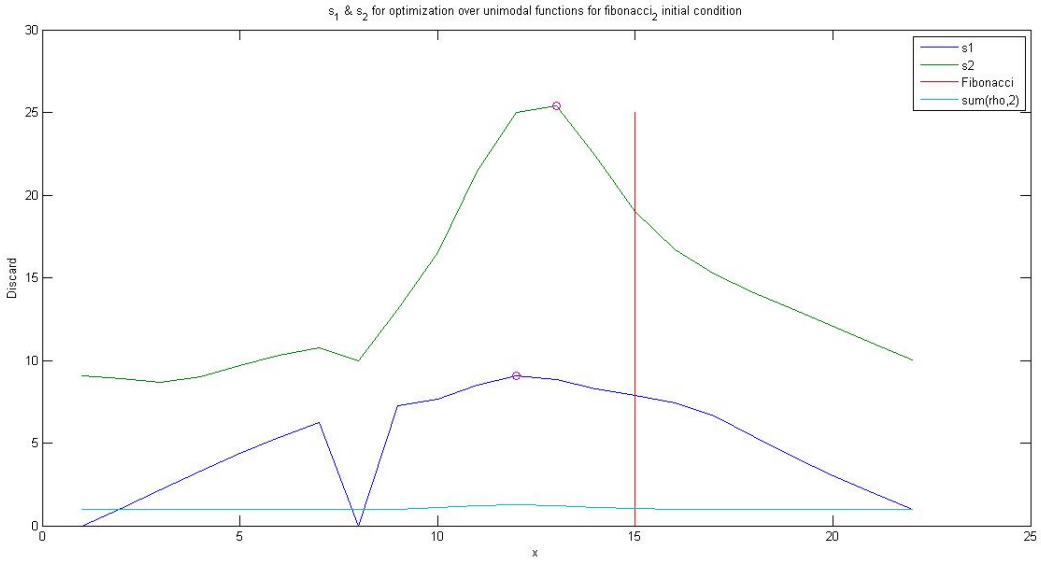


Fig. 4.6. s_1 and s_2 and $\sum_{j=1}^{nq} \rho(i,j)$ for Fibonacci Initial Conditions

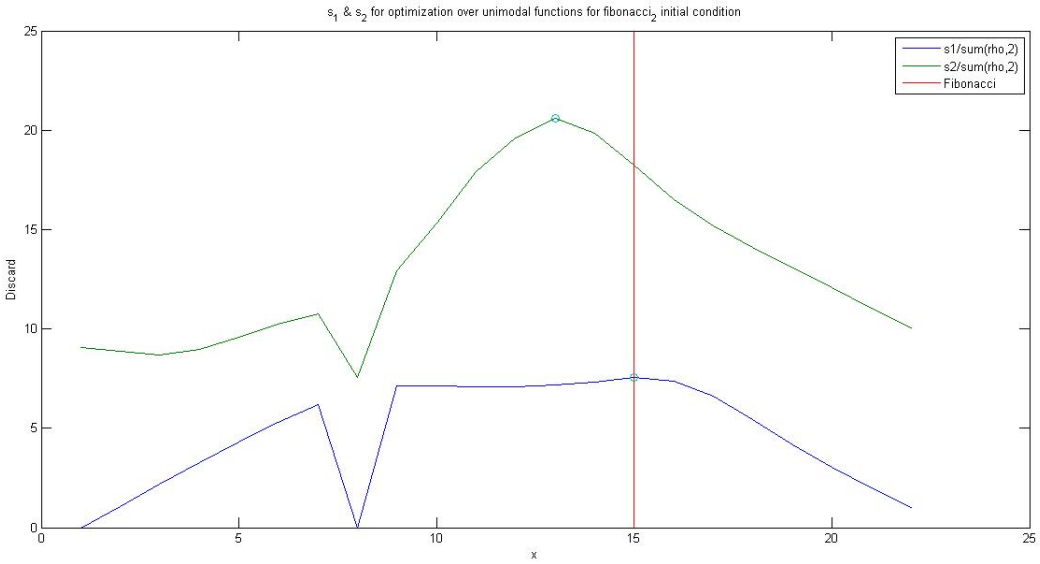


Fig. 4.7. s_1 and s_2 "Corrected" for Fibonacci Initial Conditions

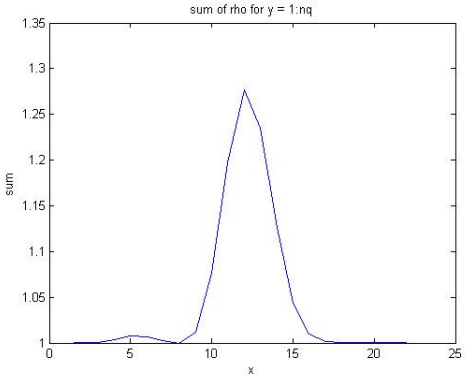


Fig. 4.8. $\sum_{j=1}^{nq} \rho(i,j)$

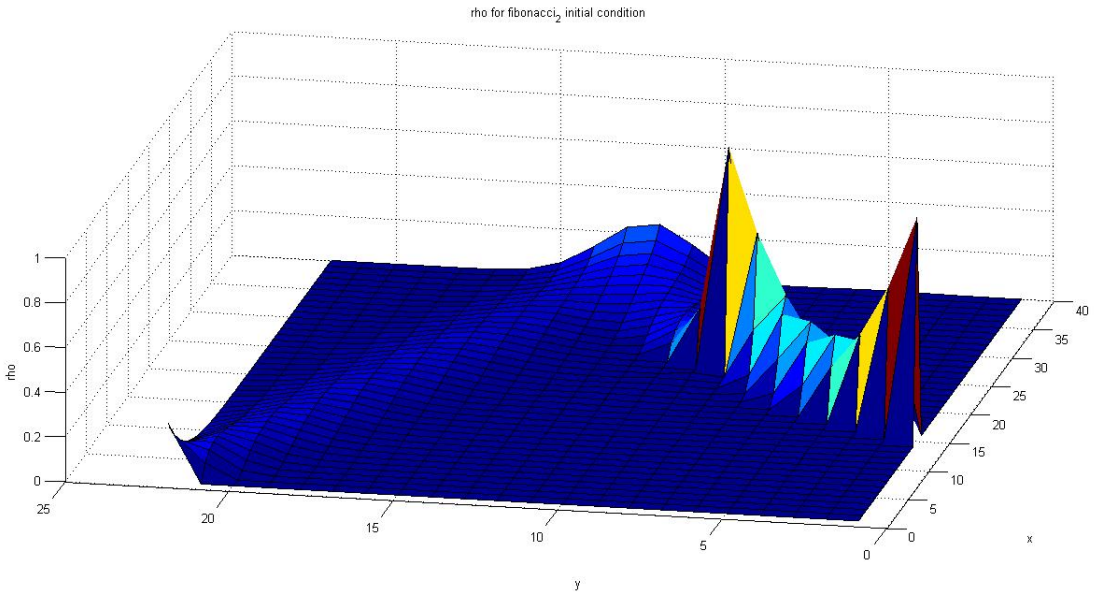


Fig. 4.9. Statistical Characterization of Uni-modal Functions for Fibonacci Initial Condition

Discussion

By observing figure 4.3 it is possible to notice that the general tendency of uni-modal functions may be described by what may be called a "butterfly" format. Either the function is an increasing function before c and uni-modal after or the contrary; this fact builds up high probability in the regions depicted by the image. In these contour conditions it is easy to see both tendencies, but in others such as in the images depicted in figure 4.5 one tendency prevails over the other leading to a clear tendency to the location of the maximum.

Figures 4.4 and 4.3 show that numerical error accumulates close to the maximum's probable location. In fact the very construction of the characterization of the set of uni-modal discrete functions requires care with numerical representation, for a grid of $n_p = 30, n_q = 30$ implies in calculations with number of the order of $\sim 30!$, and therefore the order in which the sequence of multiplications and divisions are executed imply in increase or decrease of error.

Figure 4.5 shows graphically that the classical polynomial interpolation has no predictive value over this set of uni-modal discrete functions. Certainly the contour of the "butterfly" format with linear edges can be used for better interpolative strategies.

The second experiment, given the due care to its limitations, propitiates the means to compare the $\overline{S}_2$ statistical method to the $\underline{S}_2$ mini-maximal Fibonacci sequence method. $n_p = 22$ for the chosen initial conditions, therefore the two steps of the Fibonacci sequence method will yield a worst case discard of 15 and an average of ≤ 16 for by the nature of the discrete problem there is no combination of function values such that a discard greater that 15 can be done choosing to evaluate function at $x = 15$.

Before an interpretation of figures 4.6, 4.7 is given it is necessary to consider the implications of the numerical error involved. The distance of $\sum_{j=1}^{nq} \rho(i,j)$ to the expected value 1 depicted in figure 4.8 and 4.6 can be understood as an estimate of the error involved in the construction of $\rho(x,y)$ (figure 4.9). Following this interpretation, $\rho(x,y)$ was calculated, in average, above its true value and therefore the error was propagated to each sequent value dependent on ρ, including s_1 and s_2 depicted in figure 4.6. This being said a correction proposed to decrease error of the value of s_1 and s_2 was to simply divide each value of $s_1(i)$ and $s_2(i)$ by $\sum_{j=1}^{nq} \rho(i,j)$ and thus resulting in figure 4.7. The following analysis will be done assuming that this approximation is satisfactory.

By figure 4.7 the $\overline{S}_2$ statistical method evaluating function at $x = 13$ obtains in average a discard of ~ 2 units more than the mini-maximal method. And by evaluating the objective function at $x = 13$ instead of $x = 15$ the resulting worst case discard is consequently two units bellow the Fibonacci sequence method. As can be noticed, any intermediate value of x can be chosen without simultaneous loss of mini-maximal discard and average discard and therefore as already noticed in root-searching methods the intermediate solutions constitute the Pareto set of solutions.

For better computational and numerical efficiency, attention should be given to the aspect that the finer the grid the greater the numbers and errors involved in the process. This seems to be inherent to sets of discrete functions and shouldn't be a concern in sets such as those that the Fundamental Theorem of Statistical Charac-

terization is capable of describing once $\rho(x, y)$ is obtained without a combinatoric approach.

5

Appendix: Extending Mini-maximal and Statistical Method to Multidimensional Optimization Problems

A natural question arises when studying unidimensional optimization methods: Is it possible to extend the "Fibonacci Sequence Method" or the "Statistical Method" to multidimensional optimization?

The answer is yes, but to do so it is first necessary to understand some key differences between unidimensional and multidimensional optimization that interfere drastically in how to bracket a solution (if possible) in n dimensional problems. For this the measure, given n points $p_1, p_2, ..., p_n$, of how well the n points bracket a solution to a root-searching problem and a unidimensional maximum-searching problem is displayed.

1. Root-Searching Problem: The smaller the length of two consecutive points that bracket a root the better the collection of points $p_1, p_2, ..., p_n$ is in root-searching. This results in $m(p_1, p_2, ..., p_n) = \min\{p_i - p_j \mid f(p_i) > 0 \ \& \ f(p_j) < 0\}$.
2. Unidimensional Maximum-Searching Problem: The smaller the length of two consecutive points that bracket a maximum the better the collection of points $p_1, p_2, ..., p_n$ is in unidimensional maximum-searching. This results in $m(p_1, p_2, ..., p_n) = \min\{p_i - p_j \mid \exists p_k \in p_1, p_2, ..., p_n \text{ that } f(p_k) > f(p_i), f(p_j)\}$.

These measures contain the notion of how good is an estimate of the location of the solution given by the points $p_1, ..., p_n$. Both measures are induced by the known method of bracketing a solution, and is it possible to ask if the same can be done in multidimensional optimization problems? The problem of maximizing a uni-modal function $f : [0, 1]^d \to \Re \mid d > 1$ will be analysed.

Given a set of points $p_1, p_2, ..., p_n \in [0, 1]^d$ and $y_1, y_2, ..., y_n \mid y_i = f(p_i)$ for $i = 1, ...n$. It is easy to demonstrate that any point of $[0, 1]^d$ with exception of $p_1, ..., p_{j-1}, p_{j+i}, ..., p_n$ where $y_j > y_i \mid i \neq j \ \& \ i = 1, ..., n$ can be a maximum of f therefore proper care must be taken in the construction of a measure that can effectively quantify in how "small a region" the maximum is located.

Before the measure is displayed, the following definitions will be needed:

Def. Given a set of functions $\mathbb{F}$ defined from sets P to Q and a problem $\mathbb{P}$ defined on function $f \in \mathbb{F}$ (a root searching problem or a maximum searching problems for example). A statistical characterization of the solution $\rho^* : P \to [0, \infty]$ is a function that $\rho^*(x) \mid x \in P$ measures the density of probability of the solution to problem $\mathbb{P}$

be x.

When problem $\mathbb{P}$ is a root searching problem, the statistical characterization of the solution is easily obtained by $\rho^*(x) = \rho(x, 0)$, where ρ is the statistical characterization that the mechanisms to construct have already been developed in previous sections.

Lemma 9. *The statistical characterization of the solution to a unidimensional optimization problem defined over the of the set of discrete uni-modal functions defined from* $[1, ..., n_p]$ *to* $[1, .., n_q]$ *with* $f(1) = j_1$ *is given by:*

$$\rho^*(x) = \frac{\#f \wedge_{\{2,...,n_p\}}^{\{1,...,n_q\}} \Big|_{\substack{f(1)\,=\,j_1 \\ \&\ x\,=\,x^*}}}{\delta \times \#f \wedge_{\{1,...,n_p\}}^{\{1,...,n_q\}} \Big|_{f(1)=j_1}} =$$

$$= \frac{\sum_{\substack{i\,=\,x \\ j\,=\,j_1,\,...,\,n_q}} \left(\#f \Uparrow_{\{2,...,i-1\}}^{\{j_1+1,...,j-1\}}\right)\left(\#f \Uparrow_{\{i+1,...,n_p\}}^{\{1,...,j-1\}}\right)}{\delta \times \left(\#f \Uparrow_{\{2,...,n_p\}}^{\{1,...,j_1-1\}} + \sum_{\substack{i\,=\,2,\,...,\,n_p \\ j\,=\,j_1,\,...,\,n_q}} \left(\#f \Uparrow_{\{2,...,i-1\}}^{\{j_1+1,...,j-1\}}\right)\left(\#f \Uparrow_{\{i+1,...,n_p\}}^{\{1,...,j-1\}}\right)\right)}$$

Proof. Proof is obtained via Lemma 6. □

The existence of ρ^* for multidimensional problems is evident. The construction of the statistical characterization of the solution depends on the characteristics of set $\mathbb{F}$ and problem $\mathbb{P}$ and should be analysed case by case. The construction of ρ^* is generally not trivial and an equivalent to the generic Theorem 2 for ρ^* is not yet known[1]; nevertheless it shall be assumed that this first problem(to obtain ρ^*) is already solved to define the necessary tools to construct an optimal multidimensional optimization strategy.

Def. Given a set of functions $\mathbb{F}$ defined from $[0, 1]^n$ to $\Re$, a problem $\mathbb{P}$ on $f \in \mathbb{F}$ and a level of significance $\alpha \in [0, 1]$. Let $\rho^* : [0, 1]^n \to [0, \infty]$ be the characterization of the solution of $\mathbb{P}$, then the measure m_α is defined by[2] the volume of the region R in $[0, 1]^n$ that is simultaneously minimum and contains the solution with a probability of at least α, more precisely:

$$m_\alpha = \min \int_R \mathrm{d}v \;\Big|\; \int_R \rho^* \mathrm{d}v \geq \alpha \;\&\; R \in [0, 1]^n$$

[1] It is possible though, given that the set of functions is differentiable, to calculate the probability of the selected function having the first condition of Karush−Kuhn−Tucker being satisfied. The calculations of this estimate of ρ^* will be omitted

[2] A common measure of dispersion may be used such as variance or some form of mathematical entropy, the convenience of this definition is the fact that for $\alpha = 1$ it coincides with the classical measure used in unidimensional problems.

For $\alpha = 1$ m_α in fact coincides with the conventional measures used by the classical unidimensional maximization algorithm and the classical root searching strategy. The fact that a set of points $p_1, ..., p_k \in [0,1]^n$ can't "bracket" a solution no matter what values of $f(p_i) \mid i = 1, ..., k$ over the set of uni-modal functions is nothing more than to say that m_1 is invariant to evaluations of function value at a finite number of points. On the other hand, over the set of quasi-convex functions m_1 isn't invariant, therefore for some sets of functions defined over $\Re^n$ the conventional measure can still be used.

If ρ^* is continuous, then it is easy to see that $m_{\alpha\in(0,1)}$ isn't invariant to function evaluation(for the set of uni-modal functions), this way it is possible to "bracket" the location of the solution with a significance level α by sequentially evaluating the function at points in $[0,1]^n$ and reducing the volume of the possible locations of the solution.

In unidimensional problems an objective was defined at each step to execute the maximum discard operation quantified by function l, this function, at step i is nothing more than $l(i) = m_\alpha(i) - m_\alpha(i+1)$. With this, the same definitions of mini-maximality and of statistically optimal strategy in unidimensional problems can be used in multidimensional optimization by simply using the definitions:

$$\overline{x}_1 = \operatorname{argmax} \int_{y_a}^{y_b} \rho(x,y) l(x,y) \mathrm{d}y = \operatorname{argmax} \int_{y_a}^{y_b} \rho(x,y) (m_\alpha \Big|_{f(x)=y} - m_\alpha) \mathrm{d}y$$

&

$$\overline{x}_n = \operatorname{argmax} \int_{y_a}^{y_b} \rho(x,y) \Big(l(x,y) + \max s_{n-1} \Big) \mathrm{d}y$$

Similarly the mini-maximal strategy naturally comes by definition; that is by choosing at each step a point at which the worst case has a maximum value of $l(x,y)$. The generalization to $\Re^n$ is very difficult and the theory necessary to complete this section of Optimal Black-Box Sequential Searching has still much to be researched.

References

1. Changkong, V.; Haimes, J.; *Multiobjective Decision Making: Theory and Methodology*. North Holland, 1983. 19
2. Gertz, E.M.; *A Quasi-Newton Trust-Region Method*. Mathematics and Computer Science Division, Argonne National Laboratory 2003. 32
3. Kiefer, J.; *Sequential Minimax Search for a Maximum*. Proc. Amer. Math. Soc. 4 (1953), p502-506 . 10, 31, 32, 34
4. Luenberger, D.G.; *Linear and Nonlinear Programming*. Addison Wesley, 2nd Edition, 1984. 9, 34
5. Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P.; *Numerical Recipes: The Art of Scientific Computing (3rd ed.)*. New York: Cambridge University Press, "Chapter 9. Root Finding and Nonlinear Sets of Equations", 2007. 9, 10, 31
6. Santos, J.P.O.; Mello, M.P.; Murari I.T.C.; *Introdução à Análise Combinatória*. Editora Ciência Moderna, p94, 2007. 14, 38
7. Wikipedia, the free encyclopedia *Root-finding algorithm*. `https://en.wikipedia.org/wiki/Root-finding_algorithm` Online; accessed 30-April-2013. 9
8. Yuan, G.; Lu, S.; Wei, Z.; *A Line Search Algorithm for Unconstrained Optimization*. J. Software Engineering and Aplications, 3: p503-509, 2010. 31
9. Yuan, Y.; *A review of trust region algorithms for optimization*. ICIAM 99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, Edinburgh, 2000 Oxford University Press, USA. 32
10. Zhang, Z.; *An Improvement to the Brent's Method*. International Journal of Experimental Algorithms (IJEA), Vol. (2) : Issue (1), 2001. 9, 10, 31