AN EFFICIENT ALGORITHM FOR SOLVING SINGLE VARIABLE OPTIMIZATION PROBLEMS

Article · January 2013

2 authors:


Opeyemi Abisoye
Federal University of Technology Minna


Abisoye Blessing
Federal University of Technology Minna

AN EFFICIENT ALGORITHM FOR SOLVING SINGLE VARIABLE OPTIMIZATION PROBLEMS

O.A. Abisoye

Department of Computer Science, and

Blessing Abisoye

Information Technology Services, Federal University of Technology, Minna, Nigeria

Abstract

Many methods are available for finding $x^*E\ R^n$ which minimizes the real value function f(x), some of which are Fibonacci Search Algorithm, Quadratic Search Algorithm, Convergence Algorithm and Cubic Search Algorithm. In this research work, existing algorithms used in single variable optimization problems are critically reviewed. The performance comparison of these algorithms is also examined. The algorithms are implemented using flowcharts and codes in Turbo-C programming language. These algorithms are subjected to convergence text to ascertain their efficiency. The result of the study shows that the algorithms used in single variable optimization problem such as Fibonacci, Quadratic and Cubic Search Method almost coincident. It is concluded that of the three optimization Algorithms, cubic search is the most effective single variable optimization technique.

Keywords: Optimization, Cubic, Quadratic, Fibonacci, Algorithm

Introduction

The need of optimization cannot be over emphasized in many aspects of real life system. It's of great importance in solving engineering problems, design and construction. Construction engineers and other personnel have to take many decisions aimed at either minimizing the efforts required or maximizing the desired benefits. In real life situations such desired benefits or required effort could be expressed as a function of several decision variables.

Optimization is a noun meaning "the act of rendering optimal". *Optimal* is an adjective that can be defined as "the most desirable solution possible under a restriction expressed or implied" (Emercomm, 2005)

In our world of global economies, tight supply chains and rising customer expectations the need for an organization to be able to consistently meet exact delivery dates for their customers' requirements is becoming more and more mandatory. Some customers have specific time windows for delivery to be met (e.g. the truck has to be at a specific receiving door between 10:00 and 10:15am). Many customers are now imposing financial penalties for shipments that do not arrive on time. These customers range across the spectrum from heavy manufacturers to retail chains to service industries and more. The issue for suppliers is how to be sure that product can be manufactured, get into the distribution warehouse, get onto a truck and be delivered on time and still make a profit. When one thinks of the variables involved with making sure that 100 custom-made widgets that were ordered last week make it to a customer in another country on the promised date of 12 days hence and to do it cost-effectively, it can be a rather daunting task. The issues involve incoming material supply constraints, machine and personnel availability, warehouse constraints and the lead times and constraints of the outbound logistics function (e.g. transportation type/cost, documentation requirements, etc.). In a word, organizations now must ***optimize*** their supply performance (Emercomm, 2005).

There is a type of optimization solution available for each part of the supply links in Supply Chain

Management. Manufacturing, warehousing, distribution and transportation all have their unique challenges, demands and load factors but they all have the common issue of being able to plan as accurately as possible in situations that are far from finite. Using these solutions will decrease inventory, increase productivity, reduce costs and dramatically increase on-time delivery performance.

Most TCP/IP-based networking applications were never designed specifically for operation over wireless connections. While today's 3G and tomorrow's 4G networks can deliver IP packets reliably and efficiently, in a congested situation, or even with just a very weak radio signal, throughput rates can go down significantly, delays can increase, packets may be dropped, and connections can be lost entirely. Getting reconnected might be with a different IP address, which can confuse an application that is in mid-transaction. Moving rapidly such as in a train or car also stresses connections but optimized application helps out. Benefits of optimized applications are multifold: they impose a lower network load; transactions complete more quickly and, hence, are more likely to succeed in congested situations; a lower amount of communications translates to better battery life; and users incur lower monthly service charges, especially as the industry moves in the direction of usage-based pricing models. RIM BlackBerry is a prominent example of an extremely efficient wireless application environment that benefits both operators and its users (Rysavy, 2009).

Benefits of optimized applications are multifold: they impose a lower network load; transactions complete more quickly and, hence, are more likely to succeed in congested situations; a lower amount of communications translates to better battery life; and users incur lower monthly service charges, especially as the industry moves in the direction of usage-based pricing models. RIM BlackBerry is a prominent example of an extremely efficient wireless application environment that benefits both operators and its users (UMTS, 2009)

Optimization can therefore be defined as the procedure for obtaining the "best solutions" to certain mathematical, engineering and industrial problems which are often models of physical reality (Dixon, 1993)

Optimization entails study of optimality criteria for problem, the determination of algorithm methods of solution, the study of the structure of such methods both under trial conditions and on real life problems

Of course, as would be expected there is no single method available for solving different types of optimization problems. In the broad sense, optimization can be applied in almost every activity in which numerical information is processed (science, engineering, mathematics, economics, management, commerce etc). Optimization is very useful for anyone in the art and science of compiler. Any interesting fact about the code that the compiler can be recognized becomes a candidate for optimization.

In Process Engineering, typical applications include; inventory control, planning of maintenance and replacement of equipment to reduce operating cost, optimum design of distillation column, production, planning control & scheduling, chemical reactor design, design of pumps turbines & heat transfer equipment for maximum efficiency, design and sieve tray for maximum efficiency, data fitting and optimal design of chemical plants.

In military optimization is applied to study radar deployment policies, utilization of limited resources, anti aircraft fire control, sizing and detection of enemies of submarines and fleet convoy.

The need of optimization is great in embedded environment for the 8-bit micro controller successful optimization primarily reduces the amount of ROM & RAM used. This is acid test of code generator.

Statement of the problem

A problem is posed concerning the search for the minimum point **x*** assume of a continuous unimodal function F on a given interval **[a,b].** Unimodal, in the sense that it was the only turning point. We also assume throughout that the objective function is differentiable whether or not the derivative is available for computation. The function is not only unimodal and differentiable but also univariate, which is a one variable function. However many methods are available for finding $x^*E\ R^n$ which minimize the real value function F(x). The problem is to suggest the most effective algorithm for one variable optimization from those studied in this research work. The algorithms include Fibonacci Search Algorithm, Quadratic Search Algorithm and Cubic Search Algorithm.

Objectives

- **i.** To study or examine the Fibonacci, Quadratic and Cubic Search optimization techniques
- **ii.** To write a computer program for the Fibonacci, Quadratic and Cubic Search optimization techniques
- **iii.** To subject Fibonacci, Quadratic and Cubic Search optimization techniques to a convergence test.
- **iv.** To compare the effectiveness of Fibonacci, Quadratic and Cubic Search optimization techniques
- **v.** To recommend the most effective of the three algorithms

iv.

Research methodology

i. The computer implementation of the Fibonacci, quadratic and cubic search techniques is effected by developing the algorithm and the computer programs (codes) in C language for each of the three single variable optimization techniques under study.

ii. Two different single variable optimization problems each are now repeated or posed to each of the developed or written programs, the outputs or results being tabulated respectively

iii. Performance comparism of the tabulated result; the different output for each of the posed single variable optimization problem are now compared to determine the most effective single variable optimization.

Characterisation of Optimization

An optimization problem can be defined as follow:

Find x which minimizes f(x) between a given interval known as a bracket to an optimum value where x is an n-dimensional that is x=(x1, x2, …..xn).

In optimization, problems are often concerned with global maximum or global minimum.

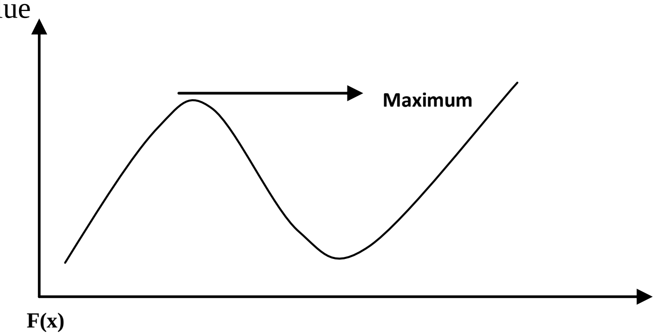
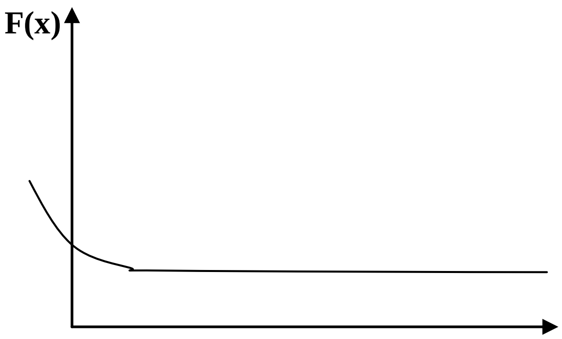


Fig 1

Optimization Concept

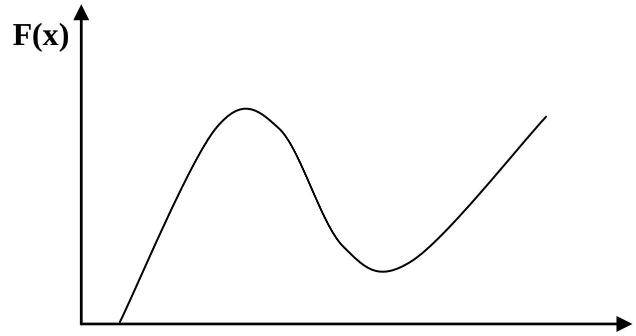
Sometimes a mathematical relationship between variables cannot be found and in order to obtain values of f(x) we have to carry out an experiment or a series of calculation. This means that we want to keep the number of evaluations to a minimum. The strategy will be to start from a base point and select where next to make an evaluation of f(x); depending on whether we get a value of f(x) nearer to the optimum (e.g. a lower value if we seek a minimum) we make a decision as to where next to choose x for a further evaluation. The values of x that we select will form a sequence {xi}; the problem then arises to as to how we detect convergence; this can be a complex problem and there is a danger in using too simple criterion for convergence.

In addition to the sequence {xi}, we have a companion sequence {f(xi)} . Therefore we have two possible criteria; for seeking a least value of f(x) these are



2.2 (a)

$\left| f(x) - f(x_{i+1}) \right| < \varepsilon$



2.2(b)

$\left| (x) - (x_{i+1}) \right| < \varepsilon^*$

Consider fig 2.2a above, the first diagram represent a shallow valley where large changes in x gives rise to only small changes in f(x) and the criterion (2.2) would be more useful, the fig 2.2b shows a steep-sided vally and changes in x now cause large changes in f(x); here the criterion will be the better choice.

Optimization problems

The travelling salesman optimization problem

Mathematicians have long amused themselves with very difficult problems that are treated as puzzles. One of the more recent of these is the traveling salesman problem it is usually formulated as follows a salesman has a certain number of cities that he must visit. He knows the distance (or time or cost) of travel between every pair of cities. His problem is to select a route that at this home city, passes through each city once and only once and return to his city in the shortest possible distance (0r in least time, or at the least cost).

If three cities are involved one of which (A) is the home base their two possible roots; ABC and ACB. For four cities there are six possible routes. But for eleven cities there are more than three and a half million possible routes. In general if there n cities there are (n-1) possible routes. Clearly, the problem is to the best route without trying each one. Although there have been many effort to solve the problem analytically, no satisfactory general method exist. However several computational techniques for solving the problem have been suggested. Competition among them has been in terms of the amount of computational time required.

The travelling salesman problem is similar to the assignment problem, except there is an additional restraint. Let Cij be the cost of going from one city I to city j and let Xij =1 if we go directly from I to J and zero otherwise. Then we wish to minimize ∑ij XijCij. However, the Xij must be so chosen that no city is visited twice before the tour of all cities is completed. In particular, we cannot go from I directly to i.

This may be avoided in the minimization process by setting Cij = ∞. Notice that one Xij = 1 for each value of i and for each value of j. thus, we could solve the assignment problem and hope that the solution satisfies the additional constraint, we often adjust the solution by inspection. This is frequently satisfactory procedure for small problems, and the reader will be able to device his own empirical rules (Ackoff, 1968). However, the objectives is to minimize the total transportation or delivery cost while ensuring that (1) the number of units transported from any origin does not exceed the number of units available at that origin and (2) the demand at each destination is satisfied (Budnick, 1994).

Allocation problems

Allocation problems involve the allocation of resources to jobs that need to be done. They occur when available resources are not sufficient to allow each job to be carried out in the most efficient manner. Therefore, the objective is to allot the resources to the jobs in such a way that as to either minimize the total cost or maximize the total return.

Consider an assignment problem where each job requires one and only one job. Resources are not divisible among job, nor are jobs divisible into resources. If resources can be divided among jobs, some jobs may be done with combination of resources; if both jobs and resources are expressed in units on the same scale, we have what is generally called the distribution problem.

The problem faced by the distribution department of a company that has three plants and four regional warehouses. Each month a list of requirements for each warehouses is available and the production capacities of plants are known. In addition, the cost of shipping a unit from each plant to each warehouse is known. The problem is to determine which plant should supply which warehouse in such in such a way as to minimize the total distribution costs.

Inventory problem

An inventory consists of usable but idle resources. The resources may be of any type e.g. men, materials, machines or money. When the resource involved is material goods in any stage of completion, inventory is usually referred to as "stock".

An inventory problem exists if the amount of resources is subject to control and if there is at least one cost that decreases as inventory increases. Normally the objective is to minimize total(actual or expected) cost. If however inventory affects demand, the objective may be maximize (actual or expected) profit.

The variables that may be controlled , separately or in combination are the following:

i. The quantity acquired(by purchasing production or some other means); that is how much

ii. The frequency or timing of aqusition; that is how often of when.

iii. The stage of completion of stocked items.

Constrained optimization algorithms

In constrained minimization problem, a subset K in Rn is prescribed and a point zEK is sought for which

$F(z) \leq F(x)$ for all xEK

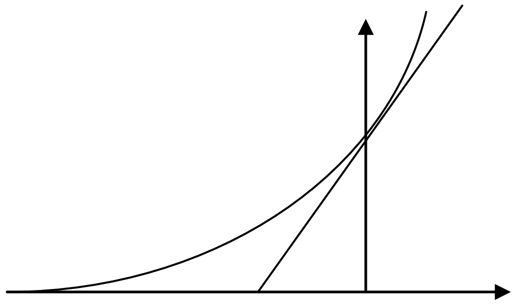
Such problems are usually more difficult because of the need to keep the points within the set K. Sometimes the set K is defined in a complicated way.(Phiips,1994).

a. Powell's Algorithm: The function is evaluated at an initial point x_1 and $x_2=x_1+d$ where d is the increment. Let the corresponding function value be f1 and f2. We choose $x_3=x_1+2d$ and if $f_1>f_2$ and $x_3=x_1-d$ if $f_1<f_2$. The optimum of the quadratic fitted through the three points is given by

$$X_m=1(x_2^2-x_3^2)f_1+(x_3^2-x_1^2)f_2-(x_1^2-x_2^2)f_3/1(x_2-x_3)f_1+(x_3-x_1)f_2-(x_1-x_2)f_3$$

If the smallest of the values$|x_1-x_m|,|x_2-x_m|,|x_3-x_m|$is less tan the required distance, we have to approximate he optimum by x_m. If this is not so then we evaluate the function at x_m and discard that point of $\{x_1, x_2, x_3\}$ which corresponds to the largest function value. The cycle is repeated until the desired accuracy is achieved. The method can lead to a point far distant from the minimum and we shall return to the minimum only slowly.(Bajpai,1997).

b. Newton's Method: Newton's method is also called the Newton-Raphson method is one of the most popular techniques for finding roots of non linear equations. In minimizing a single variable function, suppose that x_n is known approximation to a root α if f(x) =0 as shown below.



y = f(x)   α X_{n+1} x_n **(6.1a)**

The next approximation x_{n+1} taken to be the point where the tangent of the graph of y= -f(x) at $x=x_n$ intersects the x-axis. A tangent is a straight line, so its equation is of the form y=mx+d. The condition that it has slope $f'(x_n)$, $d=f(x)-x_nf_1(x_n)$. The value of y is zero when x is –d/m. So Newton's method is defined by

$$X_{n+1}= x_n-f(x_n)/f1(x_n).$$

Unconstrained optimization algorithm

In an unconstrained minimization problem a function F is defined from a n-dimensional space Rn into the real line R and a point $Z\varepsilon R_n$ is sought with the property that F(z)≤F(x) or all x**ε**Rn. It is convenient to write points in Rn simply as x,y,z and so on. If it is necessary to display the components of a point we write $x=\{x_1,x_2,\dots,x_n\}$ Unimodal function; One reasonable assumption is that on some interval [a,b] given to us in advance, F has only a single minimum(turning point)

a. **Fibonacci search convergence**:
The basis of the method is that two points c,d are placed symmetrically in a known bracket[a,b] and the bracket is the reduced to either[a,d] or[c,d] depending on which set of three points a,c,d or c,d,b satisfies the bracket in such a way as to retain the symmetry and the process repeated until the interval length is within the tolerance. For the Fibonacci search, the placing of an additional points in the initial interval is determined by finding the first Fibonnaci number Fn>L/E where L is the length of the initial interval and e is the given tolerance(The Fibonnacci number are given by $F_0=F_1=1$ and $F_n=F_{n-1}+F_{n-2}$. For the first iteration, c,d are given by the combinations
$c=x_2=(F_{n-1}(a) +F_{n-2}(b)/F_n$
$d= x_2=(F_{n-1}(a) +F_{n-2}(b)/F_n$
After n-2 iterations the interval length has been reduced to 2L/Fn and the two central points will round off error. We can deduce from these which halt of this final interval contains the minimum and so achieve the required accuracy.

a. **Quadratic search convergence**
The quadratic search is based on the fact that close to its minimum a differentiable function will behave much like its second order Taylor expansion about the minimum point. The idea is therefore to use three function values, which bracket the minimum to fit a quadratic interpretation polynomial and then to minimize this quadratic, adjust the bracket to take account of this and report this process until convergence is attained. One quadratic Search algorithm thus proceeds as follows:
Input: a known bracket a<b<c satisfying f(b)<f(a)f(c)
Repeat: Find the minimum point of this quadratic d says discard either a or c so that the remaining points bracket the minimum and relabeled.
Until:|c-a|<e-tolerance.

b. Cubic search convergence

Thus far the methods discussed have made no use of the derivative of the objective function (df_0). The derivative is available in the Cubic Search Method. This is based on minimizing an interpolating cubic which agrees with f and f_1 at two points x0<x1 which bracket the minimum so that $f1(x_0)<0<f1(x_1)$. It follows that this cube has exactly one turning point in $[x_0,x_1]$ and that this is its minimum. This point would be used in place of either X_0 or X_1 so that the bracketing property is maintained and the process repeated until the required tolerance is achieved. The formula for the minimum of the cubic is again obtained from the divided difference interpolation formula remembering for example, that $F[x_0,x_0] = f1(x_0)$

It is $X_{min} = X_o + \sqrt{b^2\text{-}3ac}\text{-}b/3a$

Where the coefficients are given by

$a= F[x_0,x_0, x_1,x_1]$ $b = f[x_0,x_0,x_1]\text{-}h_f[x_0, x_0, x_0,x_0, x_1,x_1]$ $c = f[x_0, x_0] = f^10$ $h= x_1\text{-}x_0$ (Turner,1994)

Results

Problem 1

Apply Fibonacci algorithm, quadratic search algorithm and cubic search algorithm to minimize the function $(20/x^2) + x$

Program output

Welcome to Fibonacci Search Program environment

Supply interval point a, Supply interval point b, Supply number of iterations

A	B	c	D	f(a)	f(b)	f(c)	f(d)
2.500	7.500	4.328	5.471	10.50	8.230	6.997	7.141
2.500	5.471	3.642	4.328	10.50	7.141	7.410	6.997
3.642	5.471	4.328	4.785	7.410	7.141	6.997	6.968
4.328	5.014	4.785	5.014	6.997	7.141	6.964	6.968
4.328	4.785	4.557	4.785	6.997	1.002	6.964	6.964
4.328	4.557	4.557	4.557	6.997	6.968	6.964	6.997

Optimum Value = Average of the last bracket points = (4.328+4.557)/2

=4.442 on 6th iteration

Error Tolerance (Resolution) = (4.557-4.328) = 0.229

ii. Welcome to Quadratic Search Program environment

Supply interval point a, Supply interval point c, Supply number of iterations

A	B	c	D	f(a)	f(b)	f(c)	f(d)
2.500	5.000	7.500	5.124	5.700	5.886	7.855	5.800
2.500	4.750	6.500	4.826	5.222	5.685	6.973	5.636

3.20	4.250	5.300	4.275	5.153	5.369	6.012	5.357
4.00	4.200	4.400	4.201	5.250	5.334	5.433	5.333
4.360	4.3650	4.38	4.365	5.407	5.415	5.422	5.416
4.381	4.3675	4.359	4.367	5.423	5.417	5.412	5.417

Optimum Value = Average of the last bracket points = (a+c/2

=4.3700 on 6th iteration

Error Tolerance (Resolution) = (c-a) = 0.022

(iv)Welcome to Cubic Search Program environment

Supply interval point x_0, Supply interval point x_1, Supply number of iterations

x_0	X_1	X_2	C	b	A
2.500	7.500	4.015	-5.400	2.2666	-0.2199
3.000	6.500	3.000	-2.7037	1.729	-0.2504
3.200	5.300	3.200	-2.0517	2.382	-0.5762
4.380	4.650	4.580	-0.0555	0.340	-0.1816
4.641	4.641	4.655	-0.000	0.326	-0.237
4.655	4.655	4.655	-0.000	0.326	-0.237

Optimum Value = Average of the last bracket points = (4.655+4.655)/2

=4.655 on 6th iteration

Error Tolerance (Resolution) = = 0.00

Problem 2

Apply Fibonacci algorithm, quadratic search algorithm and cubic search algorithm to minimize the function $x^2 + 6x + 2$

Program output

Welcome to Fibonacci Search Program environment

Supply interval point a, Supply interval point b, Supply number of iterations

A	B	C	D	f(a)	f(b)	f(c)	f(d)
2.500	7.300	2.850	5.460	-10.50	6.996	-7.140	-8.238
2.530	5.460	2.561	4.380	-7.50	7.419	-6.996	-7.140
2.560	3.850	2.992	3.850	-5.31	-6.270	-6.800	-6.821

2.561	3.850	2.561	2.990	-5.31	-6.270	-6.999	-6.999
2.561	3.420	2.992	2.992	-6.807	-6.821	-6.999	-6.999
2.561	3.420	2.992	2.992	-6.807	-6.821	-6.999	-6.999

Optimum Value = Average of the last bracket points = (2.561+3.420)/2

=2.9905 on 5th iteration

Error Tolerance (Resolution) = (3.420-2.561) = 0.859

ii. Welcome to Quadratic Search Program environment

Supply interval point a, Supply interval point c, Supply number of iterations

A	B	C	D	f(a)	f(b)	f(c)	f(d)
2.500	5.000	7.500	5.353	15.75	46.72	80.75	42.00
3.000	4.750	6.500	4.956	20.00	41.44	63.75	38.81
3.200	4.250	5.300	4.343	21.84	33.89	45.99	32.81
4.000	4.200	4.400	4.20	30.00	32.29	34.56	32.23
4.370	4.3675	4.350	4.367	33.97	34.17	34.14	34.18
4.380	4.3750	4.370	4.375	34.32	34.26	34.20	34.26

Optimum Value = Average of the last bracket points = (4.38+ 4.370)/2

= 4.375 on 8th iteration Error Tolerance (Resolution) = () = 0.010

iii.Welcome to Cubic Search Program environment

Supply interval point a, Supply interval point b, Supply number of iterations

X_0	X_1	X_2	c	b	A
2.500	7.500	2.506	-5.400	362.60	-48.265
3.000	6.500	3.000	-2.700	398.97	-75.90
3.200	5.300	3.200	0.047	437.23	-138.62
3.560	4.650	4.580	-0.011	525.23	-164.59
3.000	2.944	3.000	-0.005	525.23	-446.70
3.000	2.999	2.999	-0.003	525.23	-524.0

Optimum Value = Average of the last bracket points = (3.00+2.99)/2

=2.99 on 8th iteration Error Tolerance (Resolution) = 0.001

Analysis of results

Program	Fibonacci Search run time errors (sec)	Quadratic Search run time errors (sec)	Cubic Search run time errors (sec)
1	1.20	1.00	0.99
2	1.00	0.99	0.98
Average Runtime	1.10	0.99	0.98

After a careful observation of the results it is obvious that Cubic Search algorithm gives a faster convergence as compared with both Fibonacci and Quadratic Search algorithms for the fact that Cubic Search Algorithm made use of the derivative of the objective function, therefore, use of derivative can provide a much more rapid convergence to the minimum. Also, it is obvious from the result of the resolution (error tolerance) of the various function problems that Cubic Search always has the least error.

In term of run time value of the Fibonacci, Quadratic and Cubic Search computer program. It is very obvious from the run time table above that Cubic Search program gives a much more faster and accurate result as compared to both the Fibonacci and the Quadratic Search computer program.

Conclusion and recommendation

In process of computation and in dealing with univariate non-functions that are assumed to be unimodal, the use of Fibonacci, Quadratic, and Cubic Search algorithms are very effective. In practice, these searches can often be utilized over any bounded single variable search, but if the function is not unimodal, then global optimization is not guaranteed. In general, univariate search can be used in multivariate optimization through successive perturbations of each decision variable. The procedure for an n-variable optimization problem would be to fix(n-1) variable at a chosen value and search over the nth decision variable until the minimums values are determined. It is very much obvious from the analysis and results that Cubic search provides a much more rapid convergence to the minimum and gives a lesser error tolerance (resolution) when subjected to the same number of iterations as compared to Fibonacci and Quadratic Search Algorithm.

Based on the findings of this research work that determined at finding the efficient algorithm for solving single variable optimization problems, I hereby recommend Cubic search algorithm as the best algorithm for experts in the professional field where optimization problem is encountered.

References

Ackoff, R. L. and Sasieni, M. W. (1968). *Fundamentals of Operations Research*, Pennsylvania: University of Pennyslvann Book Company.

Bajpai, A.C, Mustoe, L.R. and Walker, D (1997). *Advanced Engineering Mathematics*, John Wiley and Sons Press, pp.

Budnick F. S. (1994). *Applied Mathematics for Business, Economics and the Social Sciences*, University of Rhode Island Press, pp.524-531.

Dixon A.(1993). *Optimization Fundamentals*, Lanchester Polytechnic Press.

EMERCOMM ,(2005). “EMERCOMM Business consultants lean enterprise management Research”.

Philips D. T. Ravindran, S. J. (1994). *Operation Research Principles and Practice*, Macmillin Press Ltd.

Richard, B. (1973). *Differential equations: Schaum's Outline Series*. McGrawHill Book Company.

Rysavy Research(2010). In the research titled "Mobile Broadband Capacity Constraints And the Need for Optimization".

Turner, P. R. (1994). *Numerical Analysis*, The Macmillian Press Ltd.

UMTS Forum,(2009). "A White Paper from the UMTS Forum Mobile Broadband Evolution: the roadmap from HSPA to LTE," February 2009.