AC 2008-217: IMPACT OF COMPUTING POWER ON COMPUTING SCENARIO

S.K. Sen, Florida Institute of Technology

Syamal K Sen is currently a professor in the Dept. of Mathematical Sciences, Florida Institute of Technology (FIT), Melbourne, Florida. He did his Ph.D. (Engg.) in Computational Science from the prestigious Indian Institute of Science (IISc), Bangalore, India in 1973 and then continued as a faculty of this institute for 33 years. He was a professor of Supercomputer Computer Education and Research Centre of IISc during 1996-2004 before joining FIT in January 2004. He held a Fulbright Fellowship for senior teachers in 1991 and worked in FIT. He also held faculty positions, on leave from IISc, in several universities around the globe including University of Mauritius (Professor, Maths., 1997-98), Mauritius, Florida Institute of Technology (Visiting Professor, Math. Sciences, 1995-96), Al-Fateh University (Associate Professor, Computer Engg, 1981-83.), Tripoli, Libya, University of the West Indies (Lecturer, Maths., 1975-76), Barbados.. He has published over 130 research articles in refereed international journals such as Nonlinear World, Appl. Maths. and Computation, J. of Math. Analysis and Application, Simulation, Int. J. of Computer Maths., Int. J Systems Sci., IEEE Trans. Computers, Internl. J. Control, Internat. J. Math. & Math. Sci., Matrix & Tensor Qrtly, Acta Applicande Mathematicae, J. Computational and Applied Mathematics, Advances in Modelling and Simulation, Int. J. Engineering Simulation, Neural, Parallel and Scientific Computations, Nonlinear Analysis, Computers and Mathematics with Applications, Mathematical and Computer Modelling, Int. J. Innovative Computing, Information and Control, J. Computational Methods in Sciences and Engineering, and Computers & Mathematics with Applications. Besides, he has coauthored seven books including the most recent one entitled “Computational Error and Complexity in Science and Engineering (with V. Lakshmikantham), Elsevier, Amsterdam, 2005. He had also authored several book chapters. All his research and book publications are in several areas mainly in computational science. He has been teaching several courses in areas such as stochastic and deterministic operations research, applied statistical analysis, and computational mathematics since late sixties. Further, he has been a member of the editorial board of international journals such as Computer Science and Informatics (India), and Neural, Parallel and Scientific Computations (USA). He has also been cited in Marquis Whos Who (Sep 2005).

Gholam Ali Shaykhian, NASA

Gholam "Ali" Shaykhian is a Software Engineer with Application, Simulation and Support Software Branch, Shuttle Processing Directorate, NASA KSC. Shaykhian has worked at KSC since 1986. He joined NASA in April 2000. He is a professional member of the American Society for Engineering Education (ASEE) and senior member of the Institute of Electrical and Electronic (IEEE). He has a Ph.D. in Operations Research from Florida Institute of Technology (FIT), Melbourne, Florida. Ali teaches graduate courses at FIT, University College, Melbourne, Florida.

© American Society for Engineering Education, 2008

Impact of Computing Power on Computing Scenario

Abstract Computing scenario over centuries/millenniums has been changing based on the tools/power of tools — often innovative — available to mankind. We discuss here briefly how the computing scenes go on evolving based on the availability and usage of newer and newer computing tools leading to early twenty-first century computing scene due to electronic supercomputing devices.

1 Introduction

Computing has been the necessity since time immemorial, even before the dawn of civilization. It must have come to exist when a human being realized the concept of his/her possession/property such as children, food items, and places for dwelling. Computing along with counting started evolving since then. We present here the pre-computer age methodology/psychology of computing based on the then available tools [1] along with the computer age changing scenario of computing. The main purpose is to highlight the differences between the pre-computer age ingenuity/mode of thinking based on the then available tools and the computer age innovations/mode of thinking based on the most remarkable tool, viz., the electronic computer. Also, even in the computer age where the computing power — the processing speed, the band width, and the hard disk space along with the executable memory storage space — is steadily increasing, the importance/dominance of algorithms for scientific and engineering computations are also shifting. For example, evolutionary approaches such as the genetic algorithms, ant approaches, and simulated annealing, specifically for NP-hard problems such as the traveling salesman problems, are increasingly becoming more dominant and innovative than the corresponding deterministic procedures. There are exponential-time deterministic algorithms such as the best k-digit rational approximation of a given irrational number, Gomory method for general all-integer programs, and the north-west corner rule combined with shadow cost method for transportation problems. Although such problems are intractable when dimensions are large, the currently available ultra-high computing speed — over one billion flops (floating-point operations per second) sequential speed — along with commensurable memory and band-width permits us to solve many real-world exponential practical problems in a reasonable time for reasonable values of dimensions. Such problems could not have been earlier attempted due to limitation of computing power. The parallel computing, in addition, enhances the prospect of computing deterministically the solution of many hitherto intractable practical problems. However, we have stressed the point that there will still remain many real world very useful NP-hard problems for which no deterministic algorithms will ever be tractable as these will take centuries for the deterministic outputs; no matter how much the computing speed beyond even peta-flops (10^{15} floating point operations per second) is increased [2]. The only algorithms in these problems are evolutionary approaches which are usually polynomial-time. During the pre-computer era over the past many centuries the mathematicians/physicists/astronomers had used tools such as those of geometry and intelligent mathematical derivations to perform many important numerical computations such as those for $\pi, e,$ and golden ratio to an accuracy which though in the high-speed computer era seem to be very normal/easy were definitely milestones of human ingenuity of ancient pre-computer age scientists. Even much before this period of centuries, the ancient scientists developed ways and means to keep track of accounting record by means of marking on stone slabs or on a stick of bone.

Section 2 is an overview of computing scenario during the pre-historic era, i.e. during 300,000 BC-250 BC. Section 3, on the other hand, is an exposition of mathematical ingenuity to perform computation during pre-computer era, i.e., during 200 BC till the birth of an electronic digital computer during early twentieth century. Section 4, on the other hand, presents the impact of ever increasing power of computing on the computing scenario since the appearance of the first digital computer during 1940's. Section 5 comprises conclusions.

2. Computing Scenario During Pre-historic Era (300,000 BC – 250 BC)

Universe is a gigantic errorless never-stoppable parallel computer with infinite precision Before 15 trillion BC, the universal errorfree computer boots up with a *Big Bang*. Since then the computing in nature/universe is going on continuously in a massively parallel manner nonstop using the infinite precision (word-length) real numbers with infinite precision computation. It knows no mistake and error, nor does it know any malfunction. The universal computer computes the path of the hurricane exactly so that the hurricane will follow that path without the slightest deviation. It computes and keeps the birth-death dynamic record of all living beings exactly at all times. The universal computer never violets any law of nature. All materials and living beings including human beings form the physical body (hardware) of the computer. All processing that are continuously going on in these materials and living beings are due to the bugless software perfectly embedded in its hardware. The scope of this universal computing is accessible/known only to a negligible extent to mankind. However, the age of the universal computer will be in dispute every time scientists discover new theory/phenomenon or get new instrument or new mathematical knowledge.

Computing during and before 300,000BC Probably the mankind did not have any idea of numerical units. However, they might have the concept of relative quantity — more or less and also probably had some idea of how much more or how much less when compared between two or more quantities. While we have no proof of such an idea in a human being, it will not be probably too wrong to extrapolate that human beings should have developed their individual possessions.

Possessions need to be portable, since they were required to be carried around. Things that could not be carried, were left behind. One need to have good reflex and to be quick in movement; being slow could mean instant death. The then animals were much bigger and stronger than the present day animals kept in zoos or national parks. They were fast and regarded a human just as another piece of delicious meat. There were no fences or enclosures to keep them back. In this environment humans tended to not to transport/carry too much possessions around for their own safety. We would not be wrong to say that the need to count or compute was not present as possessions were negligible and scarce.

However, when mankind started settling, they began started gathering possessions. At that time certainly they could tell that Mr. X has more or less possessions such as oranges. Let there be a heap of oranges. One could presume that they visualized the form (heap of oranges) and estimated the size mentally, to have an idea of quantities. When the heap of oranges had become larger then some oranges were added. This implies that larger the image of the heap they had in memory, the more they had. Probably this is a logical explanation of human beings interacting with the surroundings/environment.

With the passage of time, humans learnt how to make and use fire. They transformed themselves from nomadic hunting life style to domestic lifestyle, occupied and settled on pieces of land and started farming. In addition, hunting still continued away from settlements. The number of humans grew and they learnt various professions such as farming, fencing, shoe/dress-making, and doing the job of a blacksmith. Individual/family wealth/possessions started to grow and accumulate.

While ways of visualizing quantities are subjective, humans felt the need for improved means of assessing quantities and also keeping a track/record of them. The necessity prompted them to innovate or improve upon the means.

50,000 to 20,000 BC: *Computing using fingers/pebbles/bones* The first tool used as computational aids were most certainly man's own fingers. Thus it is no coincidence that the name "**digit**" comes from the fact that the 10 digits (ancient Latin digita referring to a finger or a toe as well as a numerical quantity) of the hands correspond to the 10 symbols of the common base. With the growing need to represent larger numbers than those represented/calculated by fingers and toes, the ancient man employed readily available materials such as tiny stones/pebbles. These tools had the additional advantage of storing intermediate computations for later use. It may be pointed out that the world "calculate" was derived from the Latin word "calculus" meaning pebbles.

The oldest (30,000 BC) known objects[1] employed to represent numbers were bones with notches carved into them. These bones dated around 30,000 BC were discovered in western Europe. Specifically, an over 20,000 year old wolf's jaw bone discovered in Czechoslovakia in 1937 had fifty-five notches in groups of five. For a common human being larger numbers are even today written in, say, statistics in groups of five for easy comprehension as well as for reduced chance of committing mistakes. This is probably the first evidence of the *tally system*, which was still used extensively almost up to the mid-twentieth century. With the advent of a digital computer, such a system is rarely used in the twenty-first century. Thus such a system could, therefore, qualify as one of the most enduring of all human innovations /inventions.

8500 BC: *Bone notches representing prime numbers* A piece of bone dated about 8500 BC , discovered in Africa seemed to have notches representing the prime numbers[2] 11, 13, 17, and 19. Ancient man would have viewed them as numbers with a special importance although prime numbers have probably of no relevance to day-today problems of collecting the most important necessity, viz., food. However, it was indeed amazing that man of that era had the sophistication to recognize this advanced mathematical concept and even took the trouble of writing it down in this almost non-destructible form.

Many ancient (tens of thousands of years old) artifacts which were relatively recently discovered support the idea that humans used various ways to keep track of numerical data, quantities, numbers, and possessions.

6000 BC: *Human curiosity to know about the ancient computing scientists* As the quality of life improved, people in various parts of the world wanted to know if there were more on computing behind the clouds. Scientists by virtue of indomitable curiosity ventured forward and came up with many new findings and facts of life.

[1] The age of an object can be approximately measured using carbon-14 dating method. Radiocarbon dating is a radiometric dating method that uses the naturally occurring isotope **carbon**-14 (14C) to determine the age of carbonaceous materials.

[2] Prime numbers which are divisible by 1 or by itself and not by any other number. The number 1 is not a prime number. The number 2 is the only even prime number while all the other prime numbers 3, 5, 7, 11, 13, … are odd and extends to infinity.

In Vedas originated from Indian subcontinent (i.e., Bharata Varsha meaning the Greater India) and compiled around 6000 BC, the numerals 12 (dwadwash) and 300 (trishat) were mentioned. This was the earliest known record of the decimal number system. The use of zero showed that the positional number system with base 10 was in use at that time [1]. We do not have yet any logical basis to show how long before this date (6000 BC) the decimal system including zero was invented. We are faced with the question "who invented the zero at this pre-historic date?" although the Indian mathematician Aryabhatta (b. 2765 BC) [3] was credited by most people with the invention of zero.

5500 BC: *Fractional system in Egyptian mathematics* An illustration of the use of fraction in mathematics is seen in the picture of the eye of Horus, an Egyptian deity represented as the falcon-headed god according to the Egyptian legend. They used the fractional units to represent the fractions of hekat (approximately 4.8 liters) — theunit of measure for grains. [4] may be referred to for the use on this system of mathematics. The system is based on halves. Half of 1 equals 1/2, half of 1/2 equals 1/4 and so on until the smallest value of 1/64. By adding together the values of different sections fractions are created. This system was used to record prescriptions, land, and grain.

5000 BC: Use of Abax or Abaq Using the Abax (Latin) or Abaq (Sumeric) meaning writing/calculating in dust, gave the general idea of an algorithmin unit (ALU) of a computer. This was being used in the far East. The Abaq is a wooden tabletop or a flat stone with carved straight lines. Computations are done using small pebbles of various sizes, each representing different numbers/values. Around 800 AD the Abax appeared in Europe [1].

4000 BC: *Use of clay tablets* The people of the Sumerian civilization kept records of commercial transactions on clay tablets. The Sumerian civilization emerged upon the flood plain of the lower reaches of the Tigris and Euphrates Rivers. The first man who recorded numbers using the number system based on 6 and 10 in a storage medium in 3200 BC probably was a Sumerian accountant. The discovery of this form of arithmetic helped the Sumerians with modeling their products of their economy. They used positional number system. Their commerce grew making Mesopotamia (the region that consists of Iraq and surrounding areas) the creator of western civilization [1].

3000 BC: *Abacus* The Abacus — a computing device — constructed using beads on wires was used in China. The Abacus was described for the first time in Babylon. Babylon, a city of ancient Mesopotamia. Its ruins can be found in present-day Al Hillah, Babil province, Iraq, about 52 miles south of Baghdad. An improved version was known to have been in use around 1300 BC and still in use in some parts of Asia and those of Balkan (Greece, Turkey, Slovenia, Romania, and Bulgaria). It is interesting to mention that in 1950 AD a well-trained human still beat the fastest electronic computer of 1950 AD by doing arithmetic on Abacus in a contest between man and machine. In 3000 BC, the Hindu culture flourished and large numbers were used.

2500- 250 BC: *Chronology of events* We briefly mention in the following table (Table 1), the important events that took place in this pre-historic era. It is certainly a sketchy one based on available information in literature although many significant inventions must have gone to oblivion due to lack of modern sophisticated publishing/recording machinery.

Table 1. Important events that took place during 2500 BC – 250 BC

Era	Invention
2500 BC	Egyptians came up with the idea of a thinking machine
2400 BC	Babylonians approximated π as $3\frac{1}{8}$ using the Abacus
2000 BC	Chinese writing system was developed. It was codified around 1500 BC. Indo-European tribes were moving from the North-West towards India.
1900 BC	Stonehenge situated in Wiltshire, 8 miles north of Salisbury, is made up of earthworks surrounding a circular setting of large standing stones. It could be a calendar or probably a place for spiritual events.
1850 BC	Egyptian scribe Ahmes stated that $\pi = 256/81 \approx 3.160$ and recorded in a scroll. The scroll was purchased by the Egyptologist Rhind in 1858 AD and is now in the British museum of Art.
1800 BC	An additive number system was in use in Egypt.
1438 BC	The first water clock known was probably constructed in Egypt. One of the oldest was found in the tomb of Amenhotep I, the second king of Egypt's 18^{th} dynasty, buried in 1500 BC. Others were built in China (1086 BC), Korea (1438 BC), Syria (700 BC), and Greece (500 BC).
1350 BC	Chinese used a precision of 1 decimal. They calculated beyond the precision of whole numbers and started to divide the number in parts.
1300 BC	Chinese used positional number system.
800 BC	Binary properties were exhibited.
600 BC	Pythagoras rediscovered that the sum of the squares of the sides of any right angled triangle equals the square of its hypotenuse. This was known in Babylonian time (i.e. 2400 BC or even earlier). Abacus was used in Greece. Lots of activities in Chinese arithmetic started.
500 BC	Indian mathematician Pingala, described the binary number system.
300 BC	Ten symbols representing ten numerals 0, 1, …, 9 were known to have been used in different languages such as Arabic, Devanagari (Hindi), and Tamil. The Babylonian Salamis tablet, the oldest surviving counting board probably belonging to 300 BC was discovered in 1846 on the island of Salamis near Greece.
250 BC	Ctesibius (285 BC-222 BC), possibly the first head of the museum of Alexandria, invented an automata to represent a whistling clock.

3. Mathematical innovation for calculation during Pre-computer Era (200 BC-1940 AD)

Electronic digital computer was nonexistent during this pre-computer era (200 BC-1940 AD). Yet scientists and mathematicians did not leave any stone unturned to perform calculations using mathematical properties and their innovative ideas along with the then available computing tools such as the Abacus, and (memorized) multiplication tables along with arithmetic operations performed mentally. Certainly these were not only time-consuming but also prone to computational errors since humans (living beings) are involved. It may be remarked that "To err (committing mistake) is human (living being)" and "Not to err is computer (nonliving being)". Also, the amount of computation that had been performed is simply a big numerical zero compared to what we can do using a ultra high-speed computer in twenty-first century. The computing culture today is almost infinitely taller than that during 300,000 BC-1940 AD.

Consider, for instance, the problem of solving linear ordinary differential equations (ODEs) associated with initial value problems. During pre-computer days, we would definitely use an available analytical solution procedure and obtain an analytical solution imposing the initial conditions, which remains, in general, an algebraic/transcendental expression (function in the independent variable x). This expression is of no use in real world implementation until this is translated into numbers performing numerical computation on the expression for specified values of x. During the computer era, one may follow the foregoing procedure if more convenient and common humanly possible. Alternatively, one may directly use numerical procedure to solve numerically the ODEs without bothering or without using human intelligence for obtaining an analytical solution (which may not be always possible for most real-world problems involving nonlinear ODEs/partial differential equations (PDEs)). In the later case, very little or no analytical ability on the part of the humans is called for. Further, while the domain of numerical solution space is almost infinitely larger than that of the analytical solution space. That is, while in most practical problems, analytical solutions simply cannot be obtained. For instance, there are numerous integration problems where analytical integration is impossible (in spite of all our extensive knowledge of analytical integration of a large body of functions) while numerical integration is always possible, much easier, and usable directly in real-world implementation. The simple integration

$$I = \int e^{\cos x} dx$$

cannot be analytically integrated while it can be numerically readily integrated given the limit of integration $[a,b] = [1,3]$, say, using, for example, the Simpson's 1/3 closed quadrature formula.

Golden ratio φ in nature, artfacts, and architecture The Greek mathematicians Pythagoras (about 582 BC−507 BC) and Euclid (about 330 BC−275 BC), the Italian mathematician Fibonacci (about 1175 −1250), also known as Leonardo of Pisa, the German Lutheran mathematician J. Kepler (1571−1630), the British mathematical physicist R. Penrose (1931) are just a few names over the past 25 centuries, who have spent countless hours over this simple yet amazing number, the golden ratio and its properties. Not only mathematicians but also musicians, psychologists, architects, historians, biologists, artists, and mystics have pondered over the omnipresence of this number. Although it is not as well-known as π, the golden ratio has stimulated the thought process of intellectuals of all disciplines like possibly no other number in mathematics. For a brief account as well as further links, refer [5]. The Italian mathematician Luca Pacioli (about 1445−1517) may be credited with starting the modern history of golden ratio φ in around 1509. We just mention below some of the numerous connections of φ in nature, artifacts, and architecture.

The golden ratio appears in the geometry of regular pentagrams and pentagons. Phidias built, in 5th century BC, Parthenon (a temple of Athena) statues that appear to embody the golden ratio. Plato (427 BC-347 BC) proposed five regular solids — tetrahedron, cube, octahedron, dodecahedron, and icosahedron — some of which have golden ratio connection. For example, an icosahedron (polyhedron with 20 faces) with edge length 2 in three dimensional Cartesian coordinates has the 12 vertices $(0,\pm1,\pm\varphi), (\pm1,\pm\varphi,0), (\pm\varphi,0,\pm1)$, where φ is the golden ratio [6]. The Swiss naturalist C. Bonnet (1720–1793) discovered that there were two successive Fibonacci series in the spiral *phyllotaxy* (arrangement of the leaves on the shoot) of a plant going clockwise and anticlockwise. R. Penrose (b. 1931)discovered a symmetrical pattern which uses golden ratio in *aperiodic tiling* (tiling which never repeats itself) resulting in new discoveries on quasicrystals (aperiodic structures that is capable of producing diffraction).

During 14–16th century, the aesthetics (a branch of philosophy of art known as *axiology* or *value* theory) of golden ratio developed. Consequently, book designers, artists, and architects were encouraged to adopt golden ratio in the dimensional relationships of their works yielding pleasing harmonious proportions. The golden ratio is some times used in modern artifacts such as stairs, buildings, and woodworks.

As to architecture, the front structure of the Parthenon (temple) depicts golden rectangles in its proportions. It is probably not that the architect consciously made the design keeping golden rectangles in mind. It is possibly because of other consideration such as the stability and the aesthetic sense. Archeologists have found that the Acropolis (edge of a high city) of Athens including Parthenon that several of its geometric proportions are golden ration approximately. A dimensional analysis of the Mosque of Uqba (an oldest mosque located in Kairoun, Tunisia and built in 670 AD) reveals that the designers had consistently applied golden ratio throughout the design.

As to art/painting, the canvas of Sir Lawrence Alma-Tadema (b. 1836), a finest Victorian Dutch painter "The Roses of Heliogabalus (1888)" has the dimensions 213 cm×132 cm — an almost perfect golden rectangle. In an illustration, Leonardo Da Vinci probably consciously applied golden ratio to the human face. Some thinks that in his creation of Mona Lisa, he employed the golden ratio. Piet Cornelis Mondrian (1872-1944), a Dutch painter employed the golden section in his geometrical paintings. However, a dimensional/geometrical study on 565 works of art of various eminent painters performed in 1999 inferred statistically that the mean ratio of the two sides of their paintings is 1.34 with a minimum value of 1.04 and a maximum value of 1.46.

In nature, the rabbit population seems to grow in such a way that we tend to get a feel that there is a similarity of the rabbit sequence with the Fibonacci sequence which, in the limit, produces the golden ratio. However, such an observation is crude and not possibly very enchanting.

There are even more fascinating and extensively useful number π. At different times mathematicians/scientists have estimated it value (without any electronic computer) using, say, geometrical means. Further there are many more important computations that were done using mainly human ingenuity and some kind of simple geometrical/computational tools. However, compared to today's computational achievement that does not often need human ingenuity, the computation during the pre-computer era is a numerical zero [2].We still would do better if we impose the past knowledge/analytical capability of the pre-computer era scientists on our modern computing procedures to solve complex real-world problems.

4. Impact of Increasing Computing Power on Computing Scenario

Let us explain this with an example. Let $x = [x_1 \quad x_2 \Lambda \; x_n]^t$ be an n dimensional vector, where t denotes the transpose. The problem is to find/compute a vector x that globally minimizes the function $f(x)$ which is given in a tabular form or in an analytical form. The term *optimization* implies either minimization or maximization. Minimizing the function $f(x)$ is the same as maximizing the function $[-f(x)]$. Unless otherwise specified, we will imply by the term *minimization* the global minimization. If the constraints or, equivalently, the numerical bounds on each element/variable x_i are specified, then the problem is called a constrained function optimization problem. Else, it is an unconstrained function optimization problem. Although a real world/practical design problem is rarely unconstrained, a study of this (unconstrained) class of problems is important for the following reasons.

(i) The unconstrained minimization algorithms provide a deeper insight required for the study of constrained minimization techniques.
(ii) Some of the robust[3] algorithms for constrained minimization need the use of unconstrained minimization methods.
(iii) The constraints do not have significant effect in certain design problems.
(iv) The unconstrained minimization algorithms can solve certain engineering analysis problems such as the nonlinear displacement response problems involving a structure under a specified load. Here the potential energy is minimized.

Conversion of constrained to unconstrained problem A constrained function optimization problem can be made an unconstrained function optimization problem. For instance, the constraint $a_i \le x_i \le b_i$ is equivalent to $x_i = a_i + (b_i - a_i)\sin^2 \varphi$, where φ is unconstrained. Thus, this equation can be used where x_i appears. However, such procedures can become very complicated. Hence several simpler procedures have been developed [7-14].

Gradual increase in importance of genetic algorithms over deterministic ones Always we need a numerical solution for the real world implementation. An analytical solution is of no use to an engineer until it is translated into numbers. The most important tool for a numerical solution is a computer, rather a digital computer, although an analog computer has its own usage in certain problems such as obtaining a discrete Fourier transform through the fast Fourier transform. However, the accuracy in an analog computer unlike that in a digital computer is very limited, i.e., it is usually not more accurate than 0.005%. This figure translates to four significant digits. This limitation is due to that of the device measuring a physical quantity in an analog computer, which is usually not more accurate than 0.005% [2]. A digital computer may be used to practically any finite precision (word length) and is so dominant that by the term *computer*, we would imply a digital computer and not an analog one. An analog computer may be much faster than a digital computer as in the case of a discrete Fourier transform. But the digital computers over years are progressively improved in speed, memory, as well as band width. Every 18 months the CPU (central processing unit) speed is doubling, every 12 months band width is doubling while every 9 months hard disk space is doubling. Consequently, many problems which were posed earlier and could not be solved because of computing resource limitations are now being solved. We provide below a brief account of past computing years

[3] A robust algorithm is one that must produce correct output regardless of whether the input actually belongs to the restricted domain or not, i.e., whether it is an inlier or not. In fact, a subjective implication of robust algorithm is the insensitivity to an outlier/noise.

and the gradual increase in importance of randomized algorithms such as the genetic algorithms over deterministic algorithms such as the gradient methods.

Before we proceed, we like to point out that the speed of computing, the storage space, as well as the band width go hand in hand. That is, just increasing the computing speed without increasing the memory and band width could result in an operational bottle-neck since a high speed CPU will be bogged down due to too many data retrieval and storage operations if the memory size as well as band width is not commensurable, i.e., if these are not relatively large. As stated in the previous paragraph, both CPU and memory space is progressively improving.

Pre-high speed computing years (1946-1964) This nineteen year period may be divided into two parts of the first generation (vacuum tubes) computers — early first generation (1946-53) and late first generation (1953-59) — and one part of the whole second generation (transistors) computers (1959-64) [15]. The main memory cycle time was 0.04-40 ms (milliseconds) during the early first generation while it was 0.01-0.02 ms during the late first generation and 0.002 - 0.01 ms during the second generation. An early first generation computer was capable of executing about 10^3 operations per second on an average while a late first generation could execute about 5×10^4 operations per second on an average. Hardly a negligible fraction (compared to modern computers) of practical computing existed during 1946-53. Randomized algorithms were not perceived during these years as a viable alternative to deterministic ones. This is because randomized algorithms were not so much developed as it is today. Also these needed apparently large amount of computation compared to that required by a deterministic one. In reality, however, all randomized algorithms are polynomial-time, i.e., fast [9, 11??]. Specifically genetic algorithms (global search techniques) were almost nonexistent during 1946-64. Psychologically we were more comfortable with deterministic algorithms than with nondeterministic ones. As a matter of fact we did not have much faith/confidence about the result/output which remains variant at each run as usually the case in an evolutionary (genetic/any other randomized) approach. This is because of the seed to generate required random numbers for the randomized/probabilistic algorithm differs from one run to another — a situation very much unlike any deterministic algorithm such as the gradient methods. A deterministic algorithm will always produce exactly the same output on the same computer, no matter how many times the program (algorithm) is run/executed. Thus gradient algorithms were the only practical acceptable means to optimize a multivariable function and very little of these algorithms were computerized nor were there as sophisticated gradient methods/other deterministic methods as we have today (2007).

High-speed computing years (1964-1975) These eleven years may be divided into two parts of the third generation (monolithic integrated circuits) computers — early third generation (1964-69) and late third generation (1969-75). The main memory cycle time was 0.5-2 μs (1 $\mu s = 10^{-6}$ sec.) during the early third generation while it was 0.02-1 μs during the late third generation. An early third generation computer could execute 10^6 (one million) operations per second on an average while a late third generation computer was capable of executing about 20×10^6 (twenty million) operations per second on an average. The enhanced speed allowed the scientists/engineers to explore more compute intensive problems which were hitherto discouraged due to processing/memory speed limitations. Also, they developed newer and newer algorithms suitable for computation for real world problems. Randomized algorithms such as the genetic algorithms and evolutionary approaches started gaining popularity increasingly. These algorithms started gaining momentum and were being considered as possible candidates for practical computations along with the deterministic ones. But these were yet to become sufficiently appealing for extensive computation in lieu of deterministic ones and were yet to be widely accepted means of global function optimization.

Super-high-speed computing years (1975-1990) The term *supercomputer* has a time-dependant non-rigid definition since today's supercomputer tends to become tomorrow's normal computer. Further, there is no generally accepted definition for fourth generation (very large scale integrated circuits and possibly with vector processors) as well as fifth generation (depicting artificial intelligence, which is mainly due to the software simulation of the natural intelligence) computers. It is thus not useful to carry the concept of computer generation beyond the third generation. We consider computers introduced since 1975 as modern computers and refer to the third generation computers as those of the past. However, for the purpose of speed relative to that of the past computers, a modern computer was loosely termed as a supercomputer if its speed exceeds 100 million operations per second (one hundred million floating point operations per second, i.e., one hundred megaflops). Such a technological improvement gave a significant impetus to scientists/researchers to explore much more compute intensive algorithms such as the randomized ones and perceive the scope/utility of these algorithms over the deterministic algorithms such as the gradient methods for global optimization. A gradient method could get stuck at a local minimum unless appropriate measures are taken to get out of this situation while a genetic/evolutionary algorithm has, in general, much less probability to get stuck in a local minimum.

Ultra-high-speed computing years (1990-onwards) Compared to the foregoing speeds, it would be reasonable to term a processing speed exceeding 1000 million (i.e., one billion) flops as an ultra-high speed. The ultra-high frequency band is generally accepted as 3000-300 megahertz. Electrical signals propagate no faster than the speed of light. A random access memory used to 10^9 cycles per second (one gigahertz) will deliver information at 10^{-10} (i.e., 0.1 nanosecond) speed if its diameter is 3 centimeters since in 10^{-10} seconds, light travels 3 centimeters [2]. It is physics, rather than technology and architecture, that sets up the limits/barriers to increase the computational speed arbitrarily. The physical barriers are the (i) speed of light, (ii) the thermal efficiency, and (iii) the quantum barriers. Per mass of hydrogen atom (1.67×10^{-24} gm), maximum 2.505×10^{23} bits/sec can be theoretically processed/transmitted. Since the estimated number of protons in the universe is 10^{73}, if the whole universe is dedicated to information processing, then no more than 7.9×10^{103} bits per year can be processed [2]. The ultra-high speeds along with ultra-large memory and ultra-large band-width have permitted the scientists to encroach into the realm of hitherto unexplored NP-hard problems such as a large traveling salesman problem (TSP) of immense practical importance in a meaningful way. While a deterministic algorithm for the TSP is combinatorial/exponential-time needing evaluation of $(n-1)!$ paths to obtain the exact minimum cost path, a genetic (heuristic) algorithm which is always polynomial-time would need relatively very little computation to provide us a low cost path, which though may/may not be the exact minimum cost path, that is accepted and used by the traveling salesman. Maybe in future a better (lower cost) path will be found by the algorithm possibly with increased computing power (processing speed, storage, and band width). The purpose is to impress on the fact that randomized algorithms will be the only tool to explore the vast world of NP-hard problems [2]. The deterministic algorithms will have no entry to this world as these could take billions of centuries to produce the required output. Even the estimated age of the universe is a numerical zero compared to this computation time. In the present context, we are involved in only polynomial time deterministic algorithms such as the gradient methods as well as the randomized algorithms such as the genetic algorithms for function optimization which is a polynomial-time problem. Even in such polynomial-time problems, genetic algorithms appear to be the only options for most real-world problems.

Computational complexity The optimization of the function $f(x)$ considered here is a polynomial-time problem and consequently the concerned gradient (deterministic) and genetic (randomized) approaches are all polynomial-time (i.e., fast). In this respect, all these algorithms are attractive and are without any significant edge of one category over the other.

Accuracy, flexibility, and simplicity We are essentially concerned with practical application of function optimization. We have considered typical problems including test ones and found that the genetic algorithms are significantly better than the gradient algorithms in terms of accuracy, flexibility as well as simplicity. Often partial derivatives computation accurately in a gradient method turns out to be a bottle-neck.

Global versus local optimization The gradient methods usually give a local minimum. This is not the goal of our problem. We need to determine the global minimum. So we need to devise a way possibly divide the domain into an appropriate number of sub-domains, each having only one minimum and then apply a gradient method for each sub-domain. On the other hand, genetic algorithms have the tendency to search the global minimum and hardly get stuck at a local minimum.

5. Conclusions

Increasing dominance of genetic/other randomized algorithms over gradient/other deterministic algorithms We have attempted to provide a glimpse of increasing importance of randomized algorithms (such as the genetic algorithms/evolutionary approaches) over the deterministic methods (such as the gradient algorithms) with the steadily increasing capability of computing resources such as the processing speed, band width, and storage space. Today (2007) a situation where a processing speed of over a billion FLOPS are widely and rather cheaply available on a desk top (and even on a lap top) computer has cropped up where randomized algorithms are more attractive than the deterministic ones for large real-world optimization problems both in terms of accuracy (quality of result) as well as inherent simplicity (ease of human comprehension). With further improvement in the capability, these algorithms will have more dominance over the deterministic ones which in past decades (before 1980s) were practically the only means to tackle small not-so-involved function optimization problems. It can be seen that if the computing power would have remained static (of the order of 10000 FLOPS) during the past four decades, then definitely randomized algorithms could not have gained so much of importance as these have today. So far as the time/computational complexity (cost of computation) is concerned, it is often not a serious issue with most real-world problems since all randomized algorithms as well as all deterministic ones for function optimization are polynomial-time [2, 16]. The function optimization problem is inherently polynomial-time (not NP-hard). However, when the search hyperspace is large (say, 7 or over 7 variables) and the function involving, say some combination of transcendental/special functions, is computationally large, then to track down the global optimal solution could involve significant computation though not often a major complexity issue. In such problems, the error involved in gradient algorithms that usually need some kind of knowledge of the derivative of the function could be significant to affect the global optimum value considerably. A genetic algorithm, or for that matter any randomized algorithm, that usually involves only the function computation will, on the other hand, tend to produce more accurate global minimum. Our numerical experiments with various typical problems in section 3 depict this fact both numerically as well as visually.

Genetic/other randomized algorithms are the only ways for NP-hard problems such as TSPs A traveling salesman problem (TSP) [17]. has immense importance in numerous practical applications. Although our subject matter is not connected to an NP-hard problem such as a TSP, we like to stress that no deterministic algorithms are usable because of intractability [2, 16]. The only alternatives are the randomized algorithms, viz., genetic, ant, and other evolutionary approaches such as the simulated annealing [17]. These algorithms are always polynomial-time and hence are always tractable.

Multiprocessing environment (*parallel computation*) We have seen, in section 2, that an n dimensional region is divided into k n-dimensional sub-regions. In a multi-processing environment where we have k or more processors, parallel implementation and computations are straightforward. If we have less than k processors then also a single processor would be performing the search in two or more sub-regions. Thus in a multiprocessing environment, implementation of such a genetic approach is easy for a programmer/user and the consequent computation time is very much reduced — almost by a factor of k.

Every 18 months processor speed is doubling, Every 12 months band-width is doubling while every 9 months hard disk space is doubling. Currently, in many commercially available computers, we may execute over a billion floating-point operations per second (flops). On the extreme as of now (2006) the processing speed has touched 36 billion flops. The storage (CPU registers, cache, main executable memory, hard disk) sizes and their retrieval speed also have proportionately increased and are increasing. Since for higher processing power, higher storage space is a must to avoid any bottle neck in overall processing speed., we do have terabyte auxiliary storage space currently. The next goal is to achieve peta flops (peta $=10^{15}$) speed. Is there a limit beyond which the speed cannot be increased? Indeed there is a limit set by the speed of light barrier, thermal efficiency barrier, as well as quantum barrier do limit the computational power [2]. The number of protons in the universe is estimated to be around 10^{73}. If the whole universe is dedicated to information processing, then no more than 7.8996×10^{103} bits per year can be processed [2]. However, we are still too too far away from these extreme speed which appears to be a universal maximum in silicon technology.

World's fastest computers as in November 2007 Supercomputers are typically used for highly compute intensive problems such as those in quantum physics, molecular modeling, weather forecasting and climate research, and physical simulation including that of nuclear tests. While the US is the leading consumer of ultra-high speed computing systems with 284 of the 500 systems, Europe follows with 149 systems and Asia has 58 systems. In Asia, Japan leads with 20 systems, Taiwan has 11, China 10 and India 9 [18].

The No. 1 position goes to the BlueGene/L System, a joint development of IBM and the US Department of Energy's (DOE) National Nuclear Security Administration (NNSA) and installed at DOE's Lawrence Livermore National Laboratory in California. Although BlueGene/L has been in the No. 1 position since November 2004, the current system is much faster at 478.2 teraflops compared to 280.6 teraflops six months ago before its upgrade.

BlueGene/P system installed in Germany at the Forschungszentrum Juelich (FZJ) is in the No. 2 position with the processing speed of 167.3 teraflops while the No. 3 system is at the New Mexico Computing Applications Center (NMCAC) in Rio Rancho in New Maxico, having the speed of 126.9 teraflops.

The No. 4 position goes to the Tata supercomputer called EKA (meaning "one" in Sanskrit) situated at the Computational Research Laboratory (CRL) in Pune, India. It is a Hewlett-Packard Platform 3000 BL 460c system integrated with CRL's own innovative routing technology to achieve a speed of 117.9 teraflops. CRL built the supercomputer facility using dense data center layout and novel network routing and parallel processing library technologies developed by its scientists. The second ranked supercomputer in India, rated 58th in the Top500 list, is at the Indian Institute of Science, Bangalore.

India has been making steady progress in the field of supercomputing from the time it first bought two supercomputers from the US pioneer Cray Research in 1988. US strictures on the scope of its use and its demand for intrusive monitoring and compliance led India to devise its own supercomputers using clusters of multiprocessors.

The foregoing information under the heading "World's fastest computers as in November 2007" follows from [18].

References

[1] History of Computing — Pre Historic Era 100.000 bc – 1438 bc, http://www.thocp.net/timeline/0000.htm.

[2] Lakshmikantham, V. and Sen, S.K. Computational Error and Complexity in Science and Engineering, Elsevier, Amsterdam, 2005.

[3] Lakshmikantham, V., Leela, S., and Vasundhara Devi, J. The Origin and History of Mathematics, Cambridge Scientific Publishers, Cambridge, UK, 2005.

[4] http://www.greatscott.com/hiero/eye.html

[5] http://en.wikipedia.org/wiki/Golden_ratio

[6] http://en.wikipedia.org/wiki/Icosahedron

[7] Rao, S.S. (1978). Optimization: Theory and Applications, Wiley Eastern, New Delhi.

[8] Box, M.J., Daview, D., and Swann, W.H. (1969). Nonlinear Optimization, Oliver and Boyd, Edinburgh.

[9] Fox, R.L. (1971). Optimization Methods for Engineering Design, Addison-Wesley, Reading, Massachusetts.

[10] Leitmann, G. (1962). Optimization Techniques with Applications to Aerospace Systems, Academic Press, New York.

[11] Murray, W. (1972). Numerical Methods for Unconstrained Optimization, Academic Press, London.

[12] Walsh, G.R. (1975). Methods of Optimization, Wiley, New York.

[13] Wilde, D.J. (1964). Optimum Seeking Methods, Prentice-Hall, Englewood Cliffs, New Jersey.

[14] Wolfe, M.A. (1978). Numerical Methods for Unconstrained Optimization: An Introduction, Van Nostrand Reinhold, New York.

[15] Ralston, A. and Reilly, E.D. Jr. eds. (1983). Encyclopedia of Computer Science and Engineering (2nd ed.), Van Nostrand Reinhold, New York.

[16] Krishnamurthy, E.V. and Sen, S.K. (2004). Introductory Theory of Computer Science, Affiliated East-West Press, New Delhi.

[17] Samanta, T. and Sen, S.K. (2008): Pseudo-versus quasi-random generators in heuristics for traveling salesman problem, to appear.

[18] Chidanand Rajghatta, The Times of India, TNN, India hosts world's fourth fastes supercomputer, The Times of India Daily News Paper, reported on November 13, 2007 at 2143 hours Indian Standard Time from Washington.