A Parallel Evolutionary Approach for Solving Single Variable Optimization Problems

PritiPunia
Thapar University, Patiala, India

ManinderKaur
Assistant Professor, Thapar University, Patiala, India

ABSTRACT
This paper presents the role of Parallel Evolutionary approaches (PEAs) to solve Single Variable Optimization Problems(SVOP). The proposed algorithm follows a parallel global approach to GA in which the master distributes individuals among slaves for evaluation. The master performs all genetic operators and slaves only evaluate individuals. Parallelization of Genetic Algorithm is done using PVM (Parallel Virtual Machine). The obtained results indicate that a SVOPGA (Single Variable Optimization Parallel Genetic Algorithm) is effective and represents an efficient approach to solve the systems of Single variable optimization problems. The experimental results present a successful implementation of synchronous master-slave model.

Keywords
Evolutionary algorithms, Single Variable Optimization Problems, Parallel Computing,PVM,Genetic Algorithm, Parallel Genetic Algorithm, Master-Slave.

1. INTRODUCTION
Optimization of single objective and multi-objective problems has been a long time challenge all around the world for the scientists and researchers. There have been many mathematical programming techniques that have been presented by various mathematicians to solve optimization problems. Till now, there has not been any single totally efficient and robust method to cover all optimization problems that arise in the different engineering fields. Presently, there are many research articles about optimization methods; the typical ones are based on calculus, numerical methods, and random methods [15]. The calculus based methods have been intensely studied and are subdivided in two main classes: 1) the direct search methods which find a local maximum moving over a function on the relative local gradient directions and 2) the indirect methods which usually find the local ends solving a set of non-linear equations, which are the resultant of equalizing the gradient from the object function to zero. Traditional numerical methods to solve optimized problems have their own problems attached with them like getting stuck in suboptimal solution. In order to get a global optimum solution, an approach to Evolutionary Algorithms has been made.

This paper presents a method of global optimization based on genetic algorithms. Genetic Algorithms (GAs) are a common probabilistic optimization method inspired by natural evolution. GA is easy to implement to non-differentiable functions and discrete search space. As mentioned, due to the shortcomings of the calculus based techniques and the numerical ones, the random methods have increased in popularity.

The random methods include evolutionary algorithms. The EA are robust, self-repair, self-adaptable optimization methods and are globally robust optimization methods. These methods of random search are based upon the principles of natural selection of Darwin [6] and genetic theory of Fisher [7]. Evolutionary techniques may be seen as recursive approaches based on population, which use selection strategy and random variation among individuals to generate new population. Genetic Algorithm has been the subject of discussion from long duration [2, 3, 5, and 17].GAsare the search tool that uses random selection for optimization of given function. GAs were developed by Holland in 1940s [10]. Goldberg [9] and recently Bäck[2, 3] contributed a lot

further. GAs proved to be successful in robust searches in complex search space. GAs have many applications in different domains of science, engineering and industry.GAs have the power to parallelize itself which proved them to be different from conventional mathematical techniques [16].

1.1 Single Variable Optimization Problems
Optimization of any problem is done to obtain the best result under the given circumstances. Optimization may be defined as an act of finding the conditions that give the maximum or minimum value of a given function.

The general single-objective (constraint) combinatorial optimization or a mathematical programming problem can be stated as follows:

Minimize f(x) where i<x<j

Where f(x) is the objective function and x is a real variable. The purpose of an optimization algorithm is to find a solution x, for which the function f(x) is minimum. Similarly the problem formulation for maximization is done.

1.1.1 *Optimality Criteria*
The three different types of optimal points are:

(i) Local Optimal point:
A point or solution x* is said to be a local optimal point, if no point in the neighbourhood has a function value smaller than f(x*).

(ii) Global Optimal point:
A point or solution x** is said to be a global optimal point, if no point in the entire search space has a function value smaller than The work proposes a variant of an evolutionary approach for solving f(x**).

(iii) Inflection point:
x* is an inflection point if f(x*) increases locally as x* increases & decreases locally as x* reduces or f(x*) decreases locally as x* increases and increases locally as x* decreases .

1.2 Genetic Algorithms
Based on the theory of genetics, the GA encodes each individual in the given population as a chromosome [4]. This encoding describes the parameters for the objective function being optimized. The alteration stage includes Crossover and

Mutation. In this paper the method which is used selects a random sample as parents from the population with a specified probability. A two-point crossover is then performed on these individuals which creates offspring.

Afterwards mutation is performed with a probability whichalters the parent parameters. While crossover leads to the convergence of the algorithm to one solution, mutation works against this goal, enabling the algorithm to look for better solution and exploit the unexplored areas.

1.2.1 Biological Background

All living organisms are made up of cells. In each cell there is the same set of strings of DNA collectively known as chromosome.The genes determine a chromosome's characteristic. Alterations in genes leads to differences in the set of characteristics associated withthat gene. The set of chromosome is further called the genotype. Genotype defines a phenotype (the individual) with certain fitness. During biological reproduction first occurs recombination (or crossover). Genes from parents form the whole new chromosome with recombination. The newly created offspring can then be mutated. Mutation means bit wise change in the elements of DNA. The fitness of an organism is calculated by success of the organism in its life. As per Darwinian theory the highly fit individualsget opportunities to "reproduce" and the least fit members of the population get less chance for reproduction and hence they die out [8, 18].

1.2.2 Background

Genetic algorithm (GA) is a heuristic search for solution or an optimization technique, originally inspired by the Darwinian principle of evolution through (genetic) selection. A GAsmake use of a highly abstract version of evolutionary algorithm to produce solutions to given problems. Each GA operates on a random population of artificial chromosomes. These are the finite binary strings and each one of them represents a solution to a problem. Each has fitness, a real number, which is a measure of the quality of the solution; to the given problem. The process starts with a randomly generated population of chromosomes, a GA then carries out a process of fitness-based selection and recombination to evolve a successor population, known as the next generation. During recombination, choosing parent chromosomes and their genetic material is then recombined to produce offspring.

These are then passed into the successor population. As this process is repeated, successive generations are produced and the average fitness of the chromosomes increase until some stopping criterion is reached. In this way, a GA produces a best solution to the given problem, [13]. GA was first introduced by John Holland's, to find good solutions to problems that were otherwise computationally intractable[11]. Holland's schema theorem [10] and the related hypothesis of building block, provided a conceptual and theoretical basis for the design of an efficient GAs[9]. GA theory is an active and growing field, with a range of approaches being used to describe and explain phenomena never anticipated by earlier theory. It is now one of the major pillars of the wider field of computational intelligence, which encircles techniques such as neural networks and artificial immunology. Genetic algorithms are search tools that can be used for both problems solving and modeling evolutionary systems. Since it is heuristic (it estimates a solution), GAs proceeds differently from other heuristic methods in several ways. Most importantly it works on a random population of possible solutions, where as other heuristic methods keep working on a single solution in their iterations. Another important point is that GAs is not a deterministic approach rather a probabilistic one.

A. Selection

From the current population, the algorithm selects individuals that will evolve to the next generation. A temporary population is made from the current population by selecting randomly N individuals; the chances of selection of every individual are proportional to its fitness – fitter individuals have high chance to be selected. Each individual can be selected more than once. There are different selection strategies available, but among them the most popular in the literature is roulette wheel selection. Here, the individuals are selected based upon a probability of selection given in Equation 1 where f (parent i) denotes the fitness of the i^{th} parent [1].

$$P_{selection} = F(\text{parent } i) / \sum F(\text{parent } i) \quad (1)$$

This selection method demands the objective function to be strictly positive.

B. Crossover

From temporary population, individuals are selected, with the probability, which will act as parents for crossover. Crossover refers exchanging the genetic material between individuals by imitating parts of the chromosome from one parent to another to produce children. The process would ideally combine "good" portions from every chromosome, evolving a child with a much superior fitness than its parents, but it is not known in advance which portions are "good". The crossover which is used is two-point crossover. Two random locations (locus) in parents' chromosomes are selected, and then exchange the portion of chromosome preceding the selected point 1 from parent 1 to child 1 and from parent 2 to child 2. Portion of chromosome following the selected point is copied from parent 1 to child 2 and from parent 2 to child 1. Similarly at point 2 procedure is followed. After crossover, children are put back into the temporary population in place of their parents. $P_{crossover}$ is set to an arbitrary value within range [0.6, 0.8] and these have been found to work best in most situations [1].

C. Mutation

Mutation is a phenomenon for exploiting the new areas of search space. It includes random alteration of bit values inside chromosome with a given probability $P_{mutation}$. Random number P is generated for each allele in each chromosome and if it satisfies relation $P < P_{mutation}$, then the selected allele is altered [1]. $P_{mutation}$ is assumed as the probability of mutation and it is typically set to very low values within range [0.01, 0.1]. While crossover works towards convergence of the algorithm to one solution, mutation works against this goal, enabling the algorithm to look for better solution in yet unexplored areas. Finally, the current population with temporary population and reiterate the process till the specified termination condition (such as closeness to the solution is replaced, maximum number of generations, maximum number of simulations, and so on) is met[1]. More details of individual techniques are given in [12].

1.3 PARALLEL GENETIC ALGORITHM

The three basic types of parallel Genetic algorithms are:

1) *Master-slave:* In this model, a single processor 'master' performs the genetic operations and ask other processors to evaluate individuals as shown in Figure 1. This model is quite useful when dealt with a few number of processors or when dealt with computationally intensive and complex evaluations

2)Island model: Here, every processor runs an independent Evolutionary Algorithm (EA) while using a separate sub-population. The processors cooperate in the cluster by regularly exchanging migrants (good individuals). The island model is particularly used where communication is limited. 3*) Diffusion model*: In this model, the individuals are arranged spatially, and use local neighborhood only for mating with other individuals. Here,while parallelizing, the inter-processor communication (as every individual has to communicate with its neighbors in every iteration), is only local[14]. Thus this approach is particularly suitable for heavily parallel computers with a fast local intercommunication network. In this paper, the synchronous master-slave model is used, when the master waits to receive the fitness values for all the population before proceeding to next generation. This approach is used since evaluations take quite long compared to communication time. Another reason for choosing this approach is ease of implementation and the advantage of producing the same result as the corresponding serial genetic algorithm.

1.3Parallelization Method

Often the numerical models for evaluationof fitness function are computationally intensive, with each requiring a large amount of processor time. This problem may be addressed with the help of Master Slave parallelization. In this model, the master distributes the fitness function evaluation among all the available 'slaves' processors [14, 15].The slave processors evaluate the given fitness function and returned the evaluated values to the master processor which is meant to implement the genetic operators. The similarity between sequential genetic algorithm (GA) and Master-Slave model is that it searches the decision space in the same way as the sequential GA does, although it does much faster.

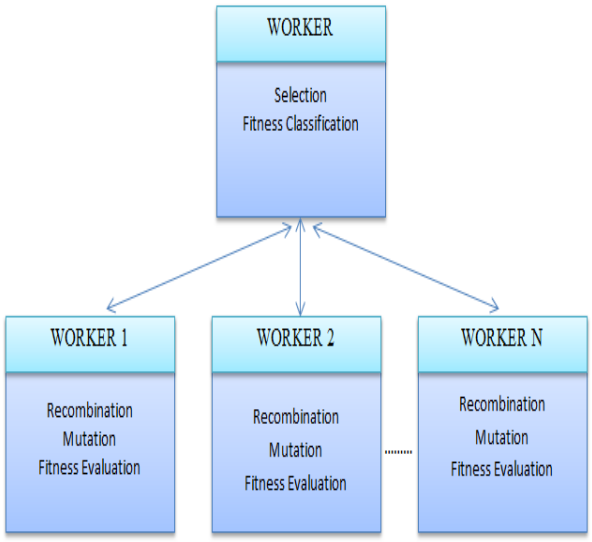

Fig 1: A schematic representation of a master-slave parallel GA

2. PROBLEM FORMULATION

One of the Single variable problem taken as test case to be solved by SVOPGA is:

Find the dimensions of the rectangle of largest area which can be inscribed in the closed region bounded by the x-axis, y-axis, and graph of $y=8-x^3$. Constraints: x lies between 1 and 2 (Figure 2).

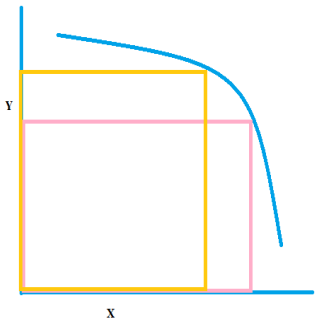

Figure 2: Rectangle inscribed in the given curve.

3.THE PROPOSED METHODOLOGY OF SOLUTION: SVOPGA ALGORITHM

Slave Algorithm:

initialize( )
while(true)
accept instruction *i* from master node
if(*i* is Evaluation_command)
do evaluation and return the fitness
else if (*i* is Finish_command)
process exit
end while

Master Algorithm:
randomly initialize the population
evaluate the whole population
while(not meet stop criteria) //number of Generations
Roulette_wheel_selection( ) //select pop(t+1) from pop(t)
Crossover and Mutation( ) // Recombine pop(t+1)
Evaluate_Population( ) // Evaluate pop(t+1); t =t+1
end while

evaluate_Population:
while(not all the individuals evaluated)
wait until a slave node N_i is available
send new individual to N_i
wait for the fitness return
end while

3.1 Coding Scheme

Binary encoding is used for ease of calculating fitness. For instance, **x** lies between 1 and 2. So, **x** may be 1.2, 1.5 or 1.99. Taking 1.99 in worst case, following 8 bits in an array are required:a = [11000111].
Here, 1.99 means multiplying it with 100 while encoding and dividing by 100 while decoding. Hence, 8 bits are required as the length of chromosome.
A sample population of 2n to 4n is generated, where n is the number of solutions needed. As n here is 1, so 2 to 4 random individuals in the present generation population are initialized. This trend will be followed in all the future generations that is the number of individuals will be four only and the genetic operators will be applied on these four individuals only to produce next generation.

3.2 CALCULATION OF FITNESS VALUE

Fitness value corresponding to *a* is:

x = p* (b-a)/(pow(2, chromlen-1)-1)

wherechromlen is the length of chromosome

and, $p = \sum_{i=0}^{chromlen} a_i . 2^{7-i}$

and, a and b are the intervals in which x lies.

5.SIMULATION RESULTS

The SVOPGA algorithm is implemented on the following set of hardware and software as mentioned in table 5.1. The work is carried out by writing the code in Turbo C in LINUX.

Table 5.1 Experimental setup for SVOPGA

Processor	Intel Core i5 Processor , Intel Core i3 Processor
Processor Speed	2.5 GHz , 2.10 GHz
Cache	3 MB
Operating System	LINUX (Ubuntu 12.04**,** Ubuntu 11.10 **)**
RAM	2 GB, 2GB
RAM Support	Up to 8 GB
Hard Drive	1 TB , 320 GB
PVM Version[17, 19]	Version 3.4.6

Simulated output results: In this section, the SVOP are solved taking the following parameters as in table 5.2:

Table 5.2: List of Parameters

Population Size	4
Number of Generations	80
Chromosome Length	8
Mutation Probability	0.001
Crossover Probability	0.66
Packet Size	10

The fitness is observed with the number of generations as in figure 3. x comes out to be 1.26 with the end of number of generations.

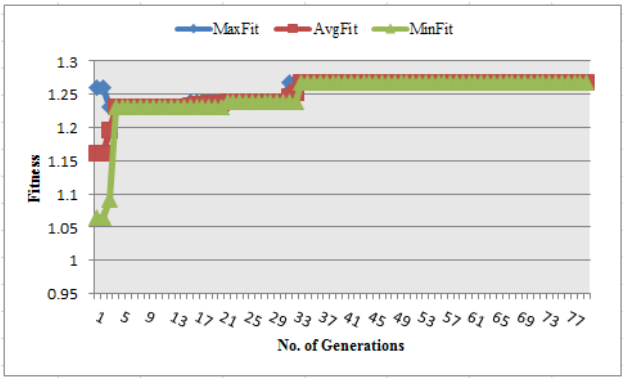

Fig 3:The fitnessversus number of generations by SVOPGA

Figure 3 shows the convergence of maximum fitness, minimum fitness and average fitness along with the increase in number of generations. It was concluded that as the number of generations' increases, the minimum and maximum value of fitness becomes comparable with average fitness.

The quality of the solution is analysed with the help of changes in the crossover and mutation probability. Figure 4 shows the effect of change of rate of crossover before and beyond 0.66.

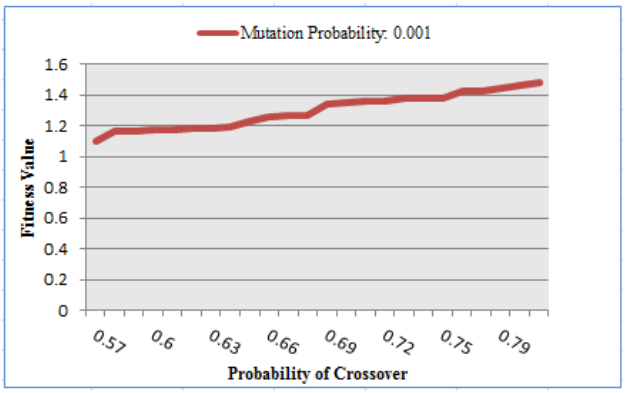

Fig 4: Effect of change of rate of crossover

In figure 4 the results showed that the increase in crossover rate beyond 0.66 has an adverse effect on the quality of crossover. A lower crossover rate that is less than 0.66 showed that more chromosomes being copied as it is in the next population from the current population resulting in the less diversity in the population in generation of fewer offspring. In figure at some points the cause of little improvement in solution quality is a result of increase in more fit individuals in newly generated population.

Figure 5 shows the change in fitness value with the change in probability of mutation where the crossover probability is kept same that is P_c =0.66.

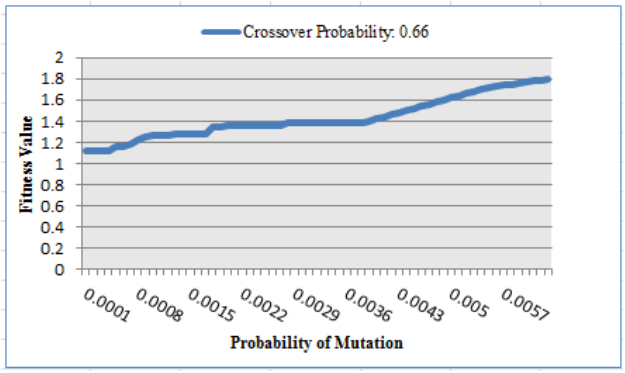

Fig 5: Effect of change of rate of mutation

Figure 5 shows the effect of change in the probability of mutation. The plot depicts that the increase in mutation rate has a negative effect on the average fitness. As the mutation rate increases, each offspring undergoes more mutation thus causing them to become genetically less likely to be the future parent for the next generation. Hence, this caused the system to search the present solution space randomly and limits the convergence of the present population. The default rate of mutation ($P_{mutation}$ = 0.001) is helpful in achieving the optimal fitness value.

6.CONCLUSIONS AND FUTURE SCOPE

Results of solving SVOP show that a Genetic Algorithm (GA) is effective and proves to be an efficient approach. The consequence of using SVOPGA as compared to the exact

solutions obtained by some different numerical methods lead to be confirming.The genetic algorithm technique finds the global optimum in relatively few evaluations compared to the size of the search space. For high performance, the SVOPGA may be provided with an associative memory and can be made to work based on experience.

7. REFERENCES

[1] Al Dahoud A. ,Ibrahiem M. M. El Emary, and Mona M., El-KareemAbd.,"Application of Genetic Algorithm in Solving Linear Equation Systems", MASAUM Journal of Basic and Applied Science, p: 179-185 Vol.1, No.2, Sept. 2009.

[2] Bäck, T. and Schwefel, H.P., "An Overview of Evolutionary Algorithms" *Evolutionary Comput.1*: pp. 1-23, 1993.

[3] Bäck, T., *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, N.Y., 1996.

[4] Bajcsy, R. and S. Kovacic, 1989. Multiresolutionelastic atching, Computer Vision, Graphics andImage Processing, 46: 1-21.

[5] Carlos D. Toledo, "Genetic Algorithms for the numerical solutions of variational problems without analytic trial functions", *arXiv:Physics*/0506188, pp. 1-3, June 2005.

[6] Darwin, C., *The Origin of the Species*, Cambridge, Ma., Harvard UniversityPress, 1967.

[7] Fisher, R.A., *TheGenetical Theory of Natural Selection*. Clarendon press, Oxford 1930.

[8] "Genetic Algorithms". http://www.tjhsst.edu/ ~ai/AI2001/GA.htm

[9] Goldberg, D.E., *Genetic Algorithms, in Search, Optimization & Machine Learning. Addison Wesley*, Publishing Company, Inc., Reading, MA, 1989.

[10] Holland, J., (1975), "Adaptation in Natural and Artificial Systems", Ann Arbor: University of Michigan Press.

[11] Jähne, B., 2002. Digital Image Processing", Springer – Verlag Berlin, Heidelberg, 2002.

[12] Michalewicz., Z., 1999. Genetic Algorithms + Data Structures = Evolution Programs 3rd Ed."; Springer; New York.

[13] McCall, J., 2005. Genetic Algorithms for Modeling and Optimization, J. Computational and Appl. Mathematics, 184: 205-222.

[14] Paz, E. C., "Designing efficient master-slave parallel genetic algorithms," IllGAL Report 97004, The University of Illinois, (1997), Available: ftp://ftp-illigal.ge.uiuc.edu/pub/papers/IlliGALs/97004.ps.Z.

[15] Paz, E. C., "Designing efficient master-slave parallel genetic algorithms," IllGAL Report 97004, The University of Illinois, (1997), Available: ftp://ftp-illigal.ge.uiuc.edu/pub/papers/IlliGALs/97004.ps.Z.

[16] Rangel-Merino, A., López-Bonilla, J.L. and Miranda, R."*Optimization Method based on Genetic Algorithms"*.Vol. 12, pp. 393-408, Oct 2005.

[17] Geist A., Beguelin A., Dongaraa J., Jiang W., Manchek R., Geist Al., Dongarra J., Manchek R., Jiang W. "Using PVM 3.0 to Run Grand Challenge applications on a Heterogeneous Network of Parallel Computers". Dec, 1992.

[18] Sandikci, B., Genetic Algorithms. Available at : .www.ie.bilkent.edu.tr/~Lors/ie572/barhaneddin.pdf

[19] Sunderam V., PVM 3 USERS GUIDE AND REFERENCE MANUAL, Oak Ridge National Laboratory/TM-12187, Engineering Physics and Mathematics Division mathematical Sciences Section, Sept., 1994.