FINE-GRAINED PARALLEL GENETIC ALGORITHM: A STOCHASTIC OPTIMISATION METHOD

A. Muhammad[1], A. Bargiela[2], G. King[3]

[1] Southampton Institute, Southampton, UK, solent@biruni.erum.com.pk
[2] The Nottingham Trent University, Nottingham, UK, andre@doc.ntu.ac.uk
[3] Southampton Institute, Southampton, UK, Graham.King@solent.ac.uk

Abstract

This paper presents a fine-grained parallel genetic algorithm with mutation rate as a control parameter. The function of the mutation rate is similar to the function of temperature parameter in the simulated annealing [Lundy'86, Otten'89, and Romeo'85]. The parallel genetic algorithm presented here is based on a Markov chain [Kemeny'60] model. It has been proved that fine-grained parallel genetic algorithm is an ergodic Markov chain and it converges to the stationary distribution.

1. Introduction

Parallel genetic algorithms are becoming more popular among researchers, due to the increased speed and efficiency [Muhlenbein'89]. There are two most popular parallel models of genetic algorithms. These are, distributed or coarse-grained model [Pettey'89 and Tanese'89], and massively parallel or fine-grained model [Manderick'89, Muhlenbein'91, and Tomassini'93]. The coarse-grained (distributed) parallel genetic algorithms assume the division of a large population into several subpopulations which are processed concurrently on different processing nodes. Each processor runs a simple genetic algorithm executing selection, crossover, and mutation on the corresponding subpopulation. There are two basic versions of the distributed parallel genetic algorithms, a) mutually exclusive algorithms, which assume no communication between the processors, b) non mutually exclusive, which permit exchange of the best individuals between processing nodes after few generations. There could be many versions of such algorithms depending on the subpopulations size, rate of the exchange of individuals and the number of individuals exchanged. However, the basic idea behind these algorithms is not different from that of the simple genetic algorithm, since each processor is running a simple genetic algorithm with only infrequent communications between processors. Consequently the convergence properties of the distributed and the simple genetic algorithms are comparable [Davis'91a, Davis'91b, Suzuki'93, and Rudolph'94].

The fine-grained (massively parallel) genetic algorithms assume the placement of only one member of the population on each processing node (Fig. 1). Any individual can only mate with individuals located on the neighbouring processing nodes. The selection is a local phenomenon rather than a global one in contrast to simple and distributed genetic algorithms. Depending on the topology of the network of the processors there could be several versions of massively parallel genetic algorithms. In this study a simple toroidal mesh of processors has been used. Each node in this mesh is connected to four immediate neighbours. The advantage of the toroidal mesh topology is the reduction in the communication radius.

The pseudo code for the fine-grained parallel genetic algorithm is given in the following:

```
for each node do in parallel
    generate an individual randomly
end parallel do
while not stop_criterion_satisfied do
    for each node do in parallel
        evaluate the fitness of the individual
        get the fitness values of four neighbouring
        individuals
        find out the optimum  fitness value
        get the neighbouring individual
        corresponding to optimum fitness
        uniform crossover with the local
        individual according to the crossover rate
        mutate the individual according to the
        mutation rate
    end parallel do
    test the stopping criteria
end while
```

The next section provides the foundations of finite discrete-time Markov chains, with particular reference to genetic algorithms. In the subsequent sections the stochastic equations for local selection, uniform crossover and mutation operator have been developed. These stochastic equations have been summed up in a finite discrete-time Markov chain based model. This model has been used to study the asymptotic convergence properties of the fine-grained genetic algorithm. Finally the results of the theoretical study have been verified using computer simulations.

2. Finite Discrete-time Markov Chains

It is obvious that the fine-grained parallel genetic algorithm is a stochastic process like the sequential counterpart. Likewise, state space of the algorithm is dependent on the representation of the search parameters. Let the population be represented as binary strings of length L then the set S of all possible individuals have 2^L members, and the individuals can be indexed with i such that $0 \le i < 2^L$.

The distribution of the population at each generation of the algorithm could be defined as vector,

$$\bar{q}_k = (q_k(0)\ q_k(1) \cdots q_k(2^L - 1))^T \qquad (1)$$

Where k is the number of the generation and q_k *(i)* is the multiplicity of the individual $i \in S$ at the generation k . The set, *S'*, of all such distributions form the search space or state space of the genetic algorithm. The size of the set *S'* is given as

$$N' = \begin{pmatrix} N + 2^L - 1 \\ N \end{pmatrix} \qquad (2)$$

Where *N* is the population size or the number of nodes in the network.

During a genetic search the distribution of the population at generation *i+1*, $\overline{q}_{i+1}$, depends only on the distribution of the previous generation, $\overline{q}_i$. This implies that the fine-grained parallel genetic algorithm as a stochastic process is a Markov chain (App., Def. 1). The transition rule of a genetic algorithm is divided into three basic transition rules, due to the stochastic genetic operators selection, crossover and mutation, and are developed in the following sections and further aggregated into a transition rule for the fine-grained parallel genetic algorithm. It has been observed from empirical studies of genetic algorithms, that regardless of the initial (random in general) population the genetic search converges to the best region of the search space.

3. The Stochastic Transition Rule due to Selection

There are number of strategies available for the selection of individuals in sequential algorithms. However they all depend only on the probability distribution. The selection mechanism in the massively parallel model is entirely different from the sequential one. In particular the topology of the computing nodes influences the selection of individuals in the subsequent generations.

There is a need for a functional representation of the topology of the population in order to explain the selection method. For a toroidal mesh topology we can write,

$$(\alpha, \beta, \gamma, \delta \} = nbhd(i, j) \qquad (3)$$

The function *nbhd* takes the location of the node as an argument and returns the individuals in the four immediate neighbouring nodes.

$$m = optimum(\alpha, \beta, \gamma, \delta \} \qquad (4)$$

The function *optimum,* as the name suggests, returns the fittest of the four arguments.

$$\begin{aligned} \sigma\ (i, (j, k)) &= 1,\ f(i) = optimum\ (nbhd\ (j, k))\ )) \\ &= 0,\ otherwise \end{aligned} \qquad (5)$$

The function σ takes two arguments *i* and *(j,k)* where *i* is the index of the individual and *(j,k)* is the location of the node, it returns *1* if fitness value of individual *i* is the maximum in the neighbourhood of *(j,k)* and it returns *0* otherwise.

The probability of selecting an individual $i \in S$ from a population given the distribution of the current generation is

$$\frac{\sum_{m,n} \sigma\ (i, (m, n))}{N} \qquad (6)$$

Where *N* is the population size, subscripts *m* and *n* represent the physical locations of the nodes and the summations goes through all nodes. It is obvious from the above expression that the probability of selection ranges from *0* to *1,* inclusive.

4. The Stochastic Transition Rule for Crossover

In the fine-grained parallel genetic algorithm the rate of crossover has been fixed to 1, so two selected individuals are always 'crossovered'. A uniform crossover has been used in this model. The crossover of two individuals is controlled by a random binary string called mask. The mechanism the crossover could be explained as multiplexing two binary numbers using a control mask.

example:

s_1 = *01010101*
s_2 = *11110000*
m = *10101011*
c = *01010001*

A 1 at position *i* of mask *m* selects the i-th bit from the parent s_1 and *0* select the i-th bit from the parent s_2. Since, the length of the mask is same as the length of the individuals in the population, *L*, the number of all possible masks is 2^L.

Let α *(i,j,k)* returns *1* if *k* is the result of uniform crossover between *i* and j, and returns 0 otherwise. The probability of obtaining an individual $i \in S$, as a result of the selection of two parents and then their uniform crossover, from the current generation given the distribution, is

$$\frac{1}{2^L} \sum_{i,j} \left(\frac{\sum_{m,n} \sigma\ (i, (m, n))}{N} \right) \left(\frac{\sum_{m,n} \sigma\ (j, (m, n))}{N} \right) \alpha\ (i, j, k) \quad (7)$$

5. The Stochastic Transition Rule for Mutation

The mutation operator used in this method is the same as in any sequential algorithm for binary population. The mutation rate is defined as p_m and unlike the crossover rate it is variable. Search space of the binary strings could be visualised as a hypercube, which is a metric space. A metric function can be defined on this search space as the number of differing bits in the two strings, this function is generally known as the Hamming distance. Let *i* and *j* are two binary strings then the hamming distance between these two strings is defined as

$$H(i, j) = |i\ XOR\ j|$$

Where *XOR* is the bitwise *exclusive-OR* function and the modulus | | is the number of 1's in the string.

The probability of obtaining individual *i* from individual *j* or vice versa, just by using mutation operator is $p_m^{H(i,j)}(1 - p_m)^{L - H(i,j)}$, the first factor is the probability of differing bits to flip and the second factor is the probability of the survival from mutation of the same bits of the two individuals.

The probability of obtaining an individual $i \in S$, as a result of selection and then mutation from the current generation given the probability distribution is

$$\sum_{j\in S}\left(\frac{\sum_{m,n}\sigma(j,(m,n))}{N}\right) p_m^{H(i,j)}\,(1-p_m)^{L-H(i,j)} \tag{8}$$

The mutation rate could be fixed to a value or it could be changed with generations.

6. The Transition Rule for the Fine-Grained Parallel Genetic algorithm

In this section the combined effect of the three genetic operators selection, crossover, and mutation are added to the one Markov transition rule in order to complete the model. The probability of obtaining an individual $i \in S$, as a result of fine-grained parallel genetic algorithm based search from the current generation given the distribution, is

$$\frac{1}{N^2 2^L}\sum_{j\in S} p_m^{H(i,j)}\,(1-p_m)^{L-H(i,j)} \sum_{k,l\in S}\left(\sum_{m,n}\sigma(k,(m,n))\right)\left(\sum_{m,n}\sigma(l,(m,n))\right)\alpha(k,l,i) \tag{9}$$

The above probability could be easily extended to obtain the distribution of next generation using the multinomial probability distribution. The resultant is a Markov transition matrix for fine-grained parallel genetic algorithm.

$$\bar{\mathbf{P}} = \frac{N\,!}{\prod_{i\in S} q_k(i)\,!} \prod_{i\in S}\left\{\frac{1}{N^2 2^L}\sum_{j\in S} p_m^{H(i,j)}\,(1-p_m)^{L-H(i,j)} \sum_{k,l\in S}\left[\sum_{m,n}\sigma(k,(m,n))\right]\left[\sum_{m,n}\sigma(l,(m,n))\right]\alpha\,(k,l,i)\right\}^{q_k(i)} \tag{10}$$

Once the transition matrix for the Markov chain representing the genetic algorithm with three genetic operators selection, crossover and mutation operators is obtained, the convergence properties of this stochastic optimisation method could be studied.

7. Convergence

It is evident from the probability transition matrix for fine-grained parallel genetic algorithms given by Eq. 10, that the mutation rate p_m is the control parameter for the corresponding Markov chain. If the mutation rate p_m is fixed for all generations then the algorithm and the corresponding Markov chain is called time-homogeneous (App., Def. 4). The homogeneous algorithm could be defined as a sequence of homogeneous Markov chains, such that each Markov chain is generated at a fixed value of p_m and it is decreased between two subsequent Markov chains. The non-homogeneous algorithm could be defined as a single non-homogeneous Markov chain and the value of p_m is changed at each step of transition. The value of p_m (the analogue of temperature in a simulated annealing) could be used to drive the genetic algorithm towards the global optimal solution. In this study the convergence of the homogeneous algorithm will be presented and as such it is essential to prove the existence of stationary distribution (under certain conditions). The Markov chain is required to be irreducible (App., Def. 6) and aperiodic (App., Def. 7) in order to be convergent to the stationary distribution. A Markov chain is irreducible if it is not absorbing, i.e., the elements of the transition matrix are less than 1. In order to be aperiodic the elements of the Markov chain's transition matrix should be greater than zero.

Let

$$\lambda = \frac{p_m}{1-p_m} \text{ (Davis'91)}$$

then

$$p_m = \frac{\lambda}{1+\lambda}$$

The transition rule for mutation, Eq. 8, can be rewritten as

$$p_3 = \frac{1}{(1+\lambda)^L}\sum_{j\in S}\left(\frac{\sum_{m,n}\sigma(j,(m,n))}{N}\right)\lambda^{H(i,j)}$$

and similarly Eq. 9 can be rewritten as

$$p = \frac{1}{(1+\lambda)^L}\sum_{j\in S}\lambda^{H(i,j)}\;p_2$$

since $0 < p_m < 1$ and $0 \le H(i,j) \le L$ therefore

$0 < \lambda < \infty$, and

$$\lambda^L \sum p_2 \le \sum \lambda^{H(i,j)}\;p_2 \le \sum p_2$$

Therefore,

$$\left(\frac{\lambda}{1+\lambda}\right)^L \le p \le \left(\frac{1}{1+\lambda}\right)^L$$

and consequently,

$$\frac{N\,!}{\prod_{i\in S} q_k(i)\,!}\left(\frac{\lambda}{1+\lambda}\right)^{NL} \le \bar{P} \le \frac{N\,!}{\prod_{i\in S} q_k(i)\,!}\left(\frac{1}{1+\lambda}\right)^{LN}$$

The irreducibility and aperiodicity follows as a result of the above inequality. From the theorem (App., Theorem) it could be concluded that since the Markov chain representing parallel genetic algorithm is time-homogeneous and aperiodic, a unique stationary distribution exists. The fine-grained parallel genetic algorithm is an ergodic Markov chain. That is there exists a unique limit distribution for the states of the chain which has a nonzero probability of being in any state at any time regardless of the initial distribution.

The convergence to the stationary distribution does not prove the convergence to the global optimal solution because even at limiting conditions $k \rightarrow \infty$, $\bar{P} > 0$. The reason behind this is the positive value of mutation rate p_m which implies that, even having reached the global optimal state, the Markov chain will move to some other state. In order to avoid this situation an elitist strategy in normally employed by the users in which the best solution from each generation is preserved.

8. Empirical Evidence

The theoretical result that the fine-grained parallel genetic algorithm is an ergodic Markov chain and a unique stationary distribution exists as p_m tends to zero has been verified through computer simulations. It has also been shown that it is not possible to get the stationary distribution just by repeatedly multiplying the initial distribution with the transition matrix under the limiting condition because, the selection of individuals is not only based on the fitness value but also on the physical distribution of the individuals on the network of processors. The algorithm proposed here was executed on a toroidal mesh of 16 processors connected as shown in the Fig.1. , and the results obtained at each generation were used to get the distribution in the next step. The function used for this simulation was the first DeJong's function $f(x) = x^2$ and the bit length L = 5. The initial distribution of the population is given in the Fig. 2. The distribution after 100 generations for mutation rate 0.0001 is given in the Fig. 3, it is the steady state as there is no change in population distribution after a threshold point.

This is the result of just one homogeneous Markov chain. It could be observed from the results that at the beginning of the simulations there is an accumulation of individuals in the lower end of the fitness range, and after 100 generations most of the individuals display higher fitness values.

9. Conclusion

The transition matrix for the Markov chain suggests that the chain is irreducible and aperiodic. These two conditions establish the fact that this chain is ergodic and a unique stationary distribution of the population exists. These properties provide enough information about the convergence of the algorithm, although they do not guarantee the convergence to the global optimal solution. However, the algorithm provides a basis for the development of variants which will have global convergence. The study of non-homogeneous algorithm and other variations like elitist strategy will be subject of a separate publication.

References

1. Davis, T. E., 1991, A simulated annealing like convergence theory for the simple genetic algorithm, International Conference on genetic algorithms and applications.
2. Davis, T.E., 1991, Toward an extrapolation of the simulated annealing convergence theory onto the simple genetic algorithm, doctoral dissertation, University of Florida.
3. Kemeny, J. G., 1960, Finite Markov Chains, Springer-Verlag.
4. Kitano, H., 1991, GA-1: A parallel associative memory processor for rule learning with genetic algorithms, International Conference on genetic algorithms and applications.
5. Lundy, M., 1986, Convergence of an annealing algorithm, Mathematical Programming, vol. 34, pp. 111-124, North-Holland.
6. Manderick, B., 1989, Fine-grained parallel genetic algorithms, International Conference on genetic algorithms and applications.
7. Muhlenbein, H., 1989, Parallel genetic algorithms, population genetics and combinatorial optimisation, International Conference on genetic algorithms and applications.
8. Muhlenbein, H., 1991, The parallel genetic algorithm as function optimiser, International Conference on genetic algorithms and applications.
9. Otten, R.H.J.M., 1989, The annealing algorithm, Kluwer Academic Publisher.
10. Pettey, C. C., 1989, A theoretical investigation of a parallel genetic algorithm, International Conference on genetic algorithms and applications.
11. Romeo, F., 1985, Probabilistic hill climbing algorithms: properties and applications, Chapel Hill conference on VLSI.
12. Rudolph, G., 1994, Convergence analysis of canonical genetic algorithms, IEEE transactions on neural networks, vol. 5, no. 1.
13. Suzuki, J., 1993, A Markov chain analysis on a genetic algorithm, International Conference on genetic algorithms and applications.
14. Tanese, R., 1989, Distributed genetic algorithms, International Conference on genetic algorithms and applications.
15. Tomassini, M., 1993, The parallel genetic cellular automata: Application to global optimisation, International Conference on genetic algorithms and applications.

APPENDIX

***Def 1 (Markov chain)**: Let $X = \{X_i : i \in S\}$ be a discrete time stochastic process with finite state space S.* If

$Pr\{X_{k+1} = j \mid X_o = i_0 \ldots X_k = i\} = Pr\{X_{k+1} = j \mid X_k = i\}$

then X is a Markov chain.

***Def 2 (Stochastic matrix):** A square matrix is called stochastic matrix if all rows are composed of probability vector or probability distribution. The following stochastic matrix*

$$\overline{P}_k = [\Pr\{X_{k+1} = j \mid X_k = i\}]$$

is the single step transition matrix of the Markov chain X at step k. If the probability distribution at step k is given as $\overline{q}_k$ the probability distribution at the next step can be determined as follows

$$\overline{q}_{k+1} = \overline{P}_k\ \overline{q}_k$$

alternatively if the initial distribution is $\overline{q}_o$ then distribution at any step could be determined as follows

$$\overline{q}_k = \left(\prod_{j=0}^{k-1}\overline{P}_j\right)\overline{q}_o$$

***Def 3 (Homogeneous Markov chains):** Let $\overline{P}_k$ be the state transition matrix for Markov chain X at step k. Then, X is called time-homogeneous if $\overline{P}_k\ (=\overline{P})$ is independent of time step k. As a result of this if the initial probability distribution*

$\bar{q}_o$ *is known then probability distribution of the Markov chain X at any step k could be determined as follows*

$$\bar{q}_k = \bar{P}^k \, \bar{q}_o$$

Def 4 (Stationary distribution): *It is defined as the vector* $\bar{q} = \{q_i\}$ *such that*

$$q_i = \lim_{k \to \infty} \Pr \, \{X(k) = i \mid X(0) = j\}$$

for any arbitrary j. It means if the stationary distribution exists then it is independent of the starting point.

Def 5 (Irreducibility): *A Markov chain is irreducible if and only if for any two states i and j there is a positive probability of reaching j from i in a finite number of transitions, i.e.,*

$$\forall \, i, j \quad \exists n : 1 \leq n \leq \infty \quad \& \quad (P^n)_{i\,j} \geq 0$$

Def 6 (Aperiodicity)*: A Markov chain is called apriodic if and only if for all i ∈ S, the greatest common divisor of all integers* $n \geq 1$*, such that*

$$(P^n)_{ii} > 0$$

is equal to 1.

Theorem : *The unique stationary distribution of a finite homogeneous Markov chain exists if the Markov chain is irreducible and aperiodic.*

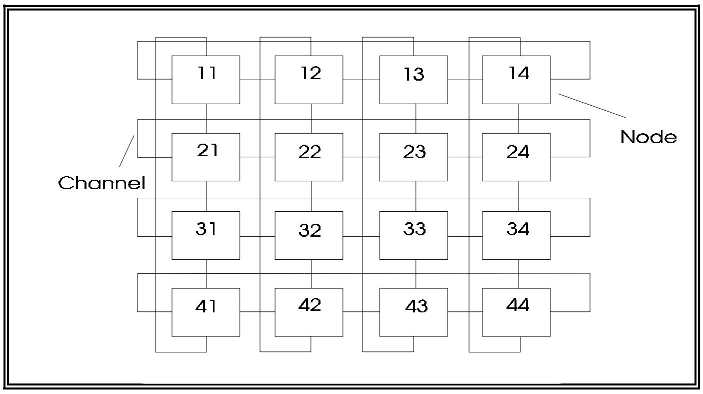

Fig. 1

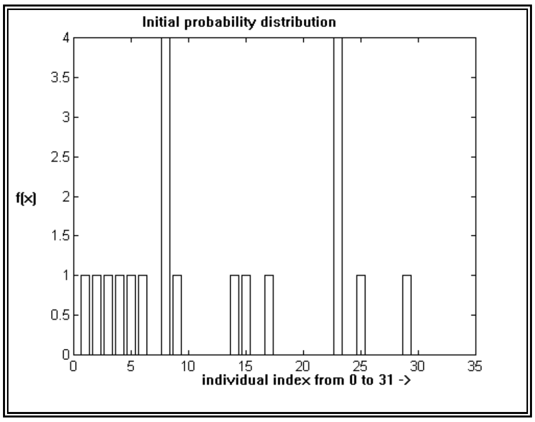

Fig. 2

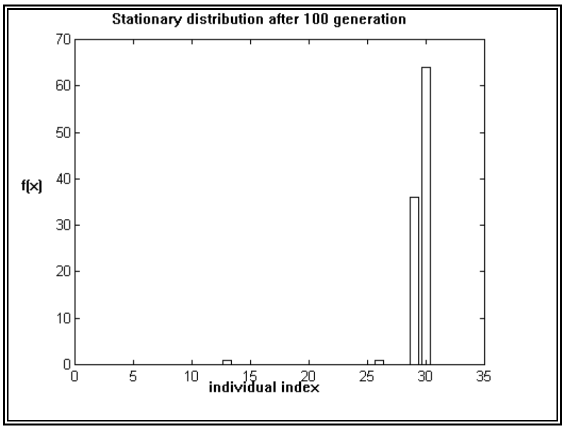

Fig. 3