The Impact of the Mutation Strategy on the Quality of Solution of Parallel Genetic Algorithms

MILENA LAZAROVA, PLAMENKA BOROVSKA, SHADA MABGAR
Department of Computer Systems
Technical University of Sofia
8, Kliment Ohridski St., Sofia 1000
BULGARIA

Abstract: The paper is aimed at investigating the influence of the mutation strategy on the quality of solution of parallel evolutionary algorithms. Parallel computational model based on independent subpopulation evolutions on multicomputer platform is suggested. Several parallel strategies for variable mutation rate at subpopulation and individual levels are investigated and their impact on the quality of the solution is evaluated and analyzed for the case study of the traveling salesman problem. Hybrid programming model utilizing both message passing (MPI) and multithreading (OpenMP) is applied. Parallelism profiling and solution quality analysis are made for the purpose of estimating the efficiency of several parallel mutation strategies.

Key-Words: parallel evolutionary computing, genetic algorithms, island model, traveling salesman problem, variable mutation rate, adaptive mutation, quality of solution

1 Introduction
Evolutionary computation (EC) is inspired by the natural evolution and survival of the fittest individuals. Evolutionary algorithms simulate an evolutionary process where the goal is to evolve solutions by means of crossover, mutation and selection based on their quality (fitness) with respect to a given optimization problem. Genetic algorithms (GA) have been a subject of intensive study and have proved to be successful in many application areas.

GA are meta-heuristic methods that prove to be efficient in providing sub-optimal solution of wide variety of hard (NP-hard or NP-complete) optimization problems in reasonable time [1, 2]. The utilization of GA requires proper choice of a set of control parameters. Different problems usually require different combinations of GA parameters (population size, number of generations, choice of genetic operators, etc.) and the success and accuracy of the solution found largely depends on the proper selection of parameters [3].

The optimal values of the control parameters may vary during the run of the evolutionary algorithm since GA rely on dynamic and adaptive evolution processes. In order to improve the performance of the GA several different approaches have been studied in the literature for GA parameter modification and control [4, 5]. Two groups of methods for parameter values setting can be distinguished: static and dynamic genetic parameters tuning. Static parameter tuning implies a method for optimization of the GA parameters before execution of the evolutionary process. Dynamic parameters control involves executing the GA with an initial guess of the parameters and then changing them during the evolution process [3, 6, 7, 8]. Control parameters can also be a subject of self-adaptation and can be modified together with the optimization of the main function [9]. In the later case GA parameters are either coded as chromosomes of the individuals [10], or meta-GA are applied that involve modification of control parameters based on the information for the GA evolution [11].

Empirical optimization of the control parameters of GA leads to adjustment of the parameters for solving particular problem [3]. A disadvantage of this approach is that it is problem specific. The suggested algorithms are rarely practically applicable to wider class of problems and furthermore the optimum parameters selection usually requires enormous amount of resources compared to running GA itself. The dynamic nature of the evolutionary processes induced in GA and the achievements in the biological sciences make adaptive strategies more likely which imply modifications of the control parameters during the evolution [5].

The genetic mutation is a random change that occurs in the characteristics of a gene. Mutation leads to major unpredictable changes in the fitness of an individual. Increasing the number of mutations results in increasing the algorithm’s freedom to search outside the current region of variable space. Mutation rate is a limitation ratio for the number of mutation events at each generation.
There exist various approaches for parallelizing GA for different applications [1]. The island parallel GA, called also coarse-grained or distributed GA, uses local subpopulation evolution on separate processors and thus resembles natural species evolution [2, 12, 13]. The strategy of parallel computations utilized by the island GA attempts not only to decrease the calculation time but also to introduce parallel evolution of subpopulations that prevents premature convergence due to domination of single highly fit individual.

The traveling salesman problem (TSP) is a well-known NP-hard combinatorial optimization problem which aims at finding the minimum cost of a tour in a number of cities. A tour is a path that starts from a city, visits each city exactly once, and returns back to the starting city. The TSP is an optimization problem that is appropriate for solving by GA and is often used as a case study for many metaheuristic algorithms that find a sub-optimal solution in reasonable time.

The purpose of this paper is to investigate the influence of the mutation strategy on the quality of solution of parallel evolutionary algorithms. Parallel computational model for independent subpopulation evolutions with periodic circular migration on multicompurer platform is suggested in the paper. Several parallel strategies for variable mutation rate at subpopulation and individual levels are investigated. The impact of the parallel mutation strategies on the quality of the solution is evaluated and analyzed for the case study of the TSP.

2 Parallel Genetic Algorithm for TSP

2.1 Parallel Computational Model

The aim of parallel computing is to reduce the time needed for solving given problem by distribution of the computations among several concurrently working processing resources. Parallel computations for optimization problem solving by GA involve improvement of the speed of finding a promising solution as well as better quality of the solution found in terms of optimization function. Parallel GA (PGA) may be constructed on both parallel programming models of functional decomposition utilizing the parallel algorithmic paradigm master-slave and data decomposition utilizing the algorithmic paradigm SPMD (Single Program Multiple Data) [14].

TSP can be solved by genetic approach by representing each tour as a chromosome that is the sequence of visiting the towns by the salesman. The fitness to be optimized during the evolution process represents the length of the tour [15]. The goal is to find the best fitted individual among a population of possible solutions to the TSP using genetic operations.

The proposed parallel computational model for solving TSP by island PGA is shown in Fig.1. The model is based on parallel paradigm “client-servers” and utilizes bi-directional linear chromosome migration. The server process is responsible for initializing parallel computation by broadcasting a randomly generated initial population to the client processes and for terminating parallel computations gathering the final best individuals from each client process by implementing collective communication reduction. The island model is implied by running several concurrent client processes evolving different subpopulations. Each client process receives its initial subpopulation from the server process and then implements the genetic operations on its subpopulation evolving it towards local optimum solution.

In that way each subpopulation may reach different local extreme value of the optimization function due to involvement of different evolutions provided by the genetic operations. The diversification of the optimal solution search may be further deepened if different genetic parameters are applied first at subpopulation level and then at population’s individual level.

Best local chromosomes migration occurs between neighbor processes periodically related to the number of the iterations (generations) of the PGA in order to facilitate the convergence of the PGA. The topology of the interconnection between demes is an important factor in the performance of the PGA because it determines how fast or how slow a good solution disseminates to other demes. The migration topology constitutes a logical ring of processes and bi-directional circular migration is selected for the PGA model [16].

Mutation in GA is aimed at prevention of premature convergence towards local extremum. Mutation rate measures the chance of mutation in the individuals of given subpopulation during each generation of the evolution process.
Mutation rate is one of the control parameters of the GA. Its value can affect the overall performance of the GA. For the parallel PGA there are several strategies for controlling the mutation rate that can be broadly differentiated by the level at which they are applied:
- variable mutation rate at subpopulation level;
- variable mutation rate at individual level.

2.2 Variable Mutation Rate at Subpopulation Level

The mutation strategies at subpopulation level can be differentiated in regard to the parallel computational model of the genetic algorithm as follows:
1. variable mutation rate for each generation that either increases or decreases as the number of the generations grows;
2. different mutation rate for each subpopulation, fixed for all generations;
3. different variable mutation rate for each local subpopulation.

In the first case the mutation rate is identical for each subpopulation but changes for the subsequent generations. Two possible cases of this strategy are considered. The first one implies fixed increment and the second one fixed decrement to the mutation rate depending on the number of generations [3, 5].

The strategy of alteration of the mutation rate at each generation can be further differentiated by the rule for computation of the modifications:
- fixed modification of the mutation rate at each generation: mutation rate is increased or decreased by a fixed small constant, for example $\mu = 0.001$.
- variable modification at each generation: mutation rate modification at each generation is determined as follows:

$$\mu_t = \frac{1}{2 + \frac{n-2}{T-t}}$$  \hspace{1cm} (1)

where $\mu_t$ is the mutation rate of the current generation $t$, $n$ is the number of the chromosomes of each individual and $T$ is the total number of generations.

The second strategy utilizes the parallelism of the evolution process based on different mutation rates for each subpopulation being evolved by one of the client processes. The rate differs for each subpopulation but is fixed and does not change at different stages of the evolution.

The third strategy involves both variable rate for the subpopulation and modifications of the initial mutation rate of each subpopulation during the evolution process. In order to further differentiate the possibilities for alteration of the mutation rate, the change of the values depend on the rank of the client process.

Thus the mutation rate increases for the odd rank processes and decreases for the even rank processes.

2.3 Variable Mutation Rate at Individual Level

Although mutation can be fixed during the evolution process and is identical for all individuals in the population biologically inspired strategies suggest natural plausibility of mutation rate variation of each individual depending on its fitness in dynamically evolving context.

The mutation strategies described in the previous section are global for the subpopulation and do not take into account the dynamics of the evolution of the single individual in the population. In order to incorporate the state of the evolution of each individual the mutation rate can be adapted to the mutation rate independently for each individual on the basis of its fitness. A self-adaptive mutation strategy alteration is suggested in the paper that modifies the value of the mutation rate of each individual in the population depending on the difference of its fitness and the average fitness of the population at the current generation. The average fitness is used as an assessment of the stage of convergence of the genetic algorithm at the current generation. The self-adaptive variable mutation rate strategy employs the following rule for individual rate modification:

$$\mu_{ind} = \begin{cases} 
0.75 & \text{if } f_{ind} > f_{avg} \\
\frac{(f_{max} - f_{ind})}{(f_{max} - f_{avg})} & \text{if } f_{ind} \geq f_{avg}
\end{cases}$$  \hspace{1cm} (2)

where $\mu_{ind}$ is the mutation rate of given individual, $f_{ind}$ is the fitness of the individual, $f_{max}$ is the maximum fitness in the population at a given iteration and $f_{avg}$ is the average fitness of the population. If the fitness of the individual is lower than the average fitness of the population at the current generation, then its mutation level is set to 0.75, otherwise it is computed according to the above rule.

The adaptive modification of the mutation rate according to the above strategy avoids the GA to be trapped into local extremum: because of the self-adaptation the individuals with high fitness will not be allowed to mutate and lower mutation rate is set for them whereas the individuals with lower fitness are assigned higher mutation rate. Thus the best fitted individuals are preserved form being mutated while the solutions with fitness lower than the average population fitness are allowed to mutate and new solutions are added to the population instead of them.
3 Parallelism Profiling and Performance Analysis

The PGA based on the suggested parallel computation model is implemented for solving TSP. The programming language and the compiler used are C++ and Microsoft Visual Studio 2005. The message passing for the parallel computations is implemented using MPI standard implementation MPICH2 v.1.0.3 and thread level parallelism is introduced by OpenMP directives. The experiments are carried out on a multicomputer parallel platform comprising 10 workstations (Intel Pentium 4, 3.2GHz, 1G RAM, Hyper-Threading) connected via Fast Ethernet switch (100 Mbps). Parallelism profiling is made utilizing Jumpshot v.4.0.

The mutation strategies described in the previous section were implemented and compared in terms of the quality of solution obtained (Table 1). Experiments with fixed mutation rate for all local subpopulations have also been carried out for the purpose of comparison.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Description</th>
<th>Mutation rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed mutation rate</td>
<td>Fixed mutation rate $\mu$</td>
<td>$\mu = 0.01$, $\mu = 0.05$, $\mu = 0.10$, $\mu = 0.15$, $\mu = 0.20$</td>
</tr>
<tr>
<td>Variable mutation rate for each generation</td>
<td>Variable increasing mutation</td>
<td>$\mu^+ = 0.001$, $0.01 \leq \mu \leq 0.2$</td>
</tr>
<tr>
<td></td>
<td>Variable decreasing mutation</td>
<td>$\mu^- = 0.001$, $0.01 \leq \mu \leq 0.2$</td>
</tr>
<tr>
<td></td>
<td>Variable decreasing mutation with adaptable modifications</td>
<td>$\mu_f = \frac{1}{(2 + \frac{n-2}{T-1})}$</td>
</tr>
<tr>
<td>Parallel variable mutation rate different for each process</td>
<td>Fixed mutation rate for all generations</td>
<td>$0.01 \leq \mu \leq 0.2$</td>
</tr>
<tr>
<td></td>
<td>Variable mutation rate for each generation, increasing for even process rank and decreasing for odd process rank</td>
<td>$0.01 \leq \mu \leq 0.2$, $\mu^+ = 0.001$, $\mu^- = 0.001$</td>
</tr>
<tr>
<td>Variable mutation rate with self-adaptation</td>
<td>Variable mutation rate for each individual at each iteration utilizing self-adaptation</td>
<td>$0.01 \leq \mu \leq 0.75$</td>
</tr>
</tbody>
</table>

Table 1. Mutation strategies

The experimental study of the impact of the mutation strategy on the solution quality of the parallel genetic algorithm for solving TSP with circular migration topology and various types of mutation rates is based on 50 runs on the multicomputer platform. The genetic parameters used in our experiments are summarized in Table 2.

Gantt’s chart for the parallel genetic computation on a multicomputer of ten workstations, showing the behavior of the circular migration model implementation is presented in Fig.2.

Fig.2. Gantt’s chart of the communication transactions during PGA

In Fig.3 communication transaction between the processes for one migration period are shown. Histograms of the communication transactions for each process demonstrate the little communication overhead implied by the migration (Fig.4). The quality difference statistics of the fitness function [17] obtained for 50 runs of solving the TSP for 200 cities implementing PGA with circular migration topology and the strategy with parallel fixed mutation rates for local evolutions are shown in Fig.5. Fig.6 presents the quality difference for variable decreasing mutation rate with adaptive modifications.
The quality difference for the self-adaptive mutation strategy is given in Fig.7. The comparison of the average quality difference of the different experimental mutation strategies is shown in Fig.8. The distribution of the deviation of the quality difference for some of the experimental mutation strategies is given in Fig.9. Obviously the best results are obtained for the self-adaptive mutation strategy with independent modification of each individual’s mutation rate savmr. The mutation strategy with parallel fixed mutation rates pvmr0.05 has also shown good quality of the obtained solutions.
The mutation strategy with increasing parallel mutation rates for generations \(v\text{mrginc}0.001\) and the mutation strategy with parallel variable mutation rates adapted at each generation \(v\text{mrga}\) have shown similar results for the solution quality. A little bit worse are the values for the deviation of the quality difference for the mutation strategy with parallel mutation rates for the generations \(v\text{mrg} \pm 0.001\) that also lead to satisfactory quality solutions. Moreover results show that the self-adaptive strategy provides better quality of solution compared to the fixed rate strategy \(f\text{mrg}0.2\) and the mutation strategy with parallel variable mutation rates adapted at each generation \(v\text{mrga}\). More than 75% of the experimentally obtained solutions deviate with up to 10% of the optimal solution for the elf-adaptive strategy. The best solution quality results obtained for the self-adaptive strategy are achieved implying small computational overhead due to the required adjustments of the mutation rate for each individual.

4 Conclusion

The paper is aimed at investigating the influence of the mutation strategy on the quality of solution of parallel evolutionary algorithms. Parallel computational model based on independent subpopulation evolutions on multicore platform is suggested. Several parallel strategies for variable mutation rate at subpopulation and individual levels are investigated and their impact on the quality of the solution is evaluated and analyzed for the case study of the TSP. Hybrid programming model utilizing both message passing (MPI) and multithreading (OpenMP) is applied. Results show that the adaptive mutation strategy with parallel variable mutation rates adapted at each generation \(v\text{mrga}\) have shown similar results for the quality solutions. Moreover results show that the self-adaptive strategy provides better quality of solution compared to the fixed rate strategy \(f\text{mrg}0.2\) and the mutation strategy with parallel variable mutation rates adapted at each generation \(v\text{mrga}\). More than 75% of the experimentally obtained solutions deviate with up to 10% of the optimal solution for the elf-adaptive strategy. The best solution quality results obtained for the self-adaptive strategy are achieved implying small computational overhead due to the required adjustments of the mutation rate for each individual.

References:


