Adaptive Genetic Algorithm for Multiprocessor Scheduling

Mohamed M. Zahran
CairoUniversity,Giza, EGYPT
mzahran@ ieee.org
Tel : +2 02 3540125

Ashraf H. Abdel Wahab
Computers & Systems Dept.
Electronic Research Institute
El Tahrir str. Dokki, Giza, EGYPT
ashraf@eri.sci.eg
Tel : +2 02 3310502/3351631

Samir I. Shaheen
Computer Engineering Dept.
Faculty of Engineering
Cairo University, Giza, EGYPT
sshaheen@frcu.eun.eg
Tel : +2 02 3358577/5735208

This research presents a solution for the NP-hard problem of static multiprocessor scheduling using adaptive genetic algorithm (AGA), which represents a modification of the conventional genetic algorithm and gives optimal or near optimal solutions in a fewer number of generations. The main performance criteria to be optimized is the schedule makespan, defined as the finishing time of the last task in the task graph. The use of GA to schedule task graphs to multiprocessor system has been done before [Hou et al 1994]. The proposed AGA simple string of genes is used where the index indicates the task to be scheduled and the alleles represent the processors numbers. Remainder stochastic sampling without replacement was used as the selection scheme together with niching method to get multiple solutions. Two types of crossover were used: single-point crossover and uniform crossover. Also, two types of mutation were used: swapping mutation and a proposed mutation that causes high disruption. Depending upon the degree of diversity in the population, the selection between the types of crossover and mutation at each generation is performed. Choosing crossover mates is made adaptively at each generation depending on the degree of similarity between parents. This similarity is done using hamming or Euclidean distance. The degree of similarity used to choose mates changes adaptively at each generation. Probabilities of crossover and mutation are adapted depending on the current population and the chromosomes chosen as proposed in [Pal and Wang 96].

The proposed adaptive algorithm:

1. *Generate initial population at random*
2. *Assign best_so_far to best individual in the population*
3. *While number of generations have not reached the required maximum do:*
   - *a- Perform remainder stochastic without replacement*
   - *b- Adapt similarity metrics*
   - *c- Choose type of crossover and mutation*
   - *d- do until new population is formed:*
     - *d1) choose first mate using tournament selection*
     - *d2)chose second mate based on similarity metrics*
     - *d3)calculate P_c*
     - *d4)perform crossover with probability P_c and insert in new population the best two individuals*
   - *e-for each member of new pop. do:*
     - *e1)calculate P_m*
     - *e2)perform mutation with prob P_m and if the new individual is better then insert it in the pop.*
   - *f- if best individual in the current population is better than best_so_far then assign it to best_so_far*
   - *g- assign best_so_far to the worst Individual in the population*
   - *h- if after ([maximum number of generations]/6) best_so_far has not changed:*
     - *h1)generate 0.75*pop_size population at random*
     - *h2)the remaining population is assigned as best_so_far*

Experiments: The implemented system has been tested against conventional GAs as well as ranked weight [Samadzadeh and Henrick 92] method. The AGA gives better results and reaches a better solution in a fewer number of generations

References:

Hou,E.S., Ansari, N. and Ren,H.(1994) A GA for multiprocessor scheduling . IEEE Trans. Parallel and distributed systems, vol.5,No.2,pp.113-120

Pal,K.S. and Wang,P.(1996) Genetic algorithms for pattern recognition. CRC press

Samadzadeh,F.A and Henrick,G.E.(1992) A heuristic multiprocessor scheduling algorithm for creating near optimal schedules using task system graphs . Proc. ACM/SIGAPP symposium on applied computing (VOL II): technological challenges of the 1990s, pp.711-718