International Journal Of Computer Science And Applications Vol. 1, No. 3, December 2008
ISSN 0974-1003

A Solution to combinatorial Optimization Problem using Memetic Algorithms

Aaquil Bunglowala
Department of Elex. & Telecommunication
Sanghvi Institute of Management & Science, Indore MP
aaquil786@yahoo.co.in

Dr. B. M. Singhi
Department of Elex. & Telecommunication
Medi-Caps Institute of Technology & Management, Indore
bmsinghi@gmail.com

ABSTRACT

The growing complexity in the hardware now necessitates in improving the performance of searching algorithms. The Genetic Algorithm and local heuristics have opened up a whole new paradigm of probability based approaches to complex NP-hard and NP-complete problems of real world. Genetic algorithms do not guarantee global optimum solution to a problem but are generally good at finding acceptable solution to problems. In complex combinatorial spaces, hybridization with other optimization techniques can greatly improve the efficiency of search. Memetic algorithms is an improvisation over genetic algorithms and combines global and local search by using evolutionary algorithms to perform exploration while the local search methods are used for exploitation. Here, exploitation is the process of visiting entirely new regions of a search space where the gain can be high. Recently the concept of grid computing has taken up the task of improving the computational abilities of systems. It is the combination of distributed, high throughput and collaborative systems for the effective sharing and distributed coordination of resources which belong to different control domains.

This paper discusses the advent of genetic algorithms (GAs) and memetic algorithms (MAs) as a solution to combinatorial optimization problems and procedures are laid down to strike a balance between genetic search and local search in MAs. The MAs for circuit partitioning in VLSI floor planning have been briefed. We have addressed the complexity issues in context of MAs as part of our research work. The problem of cell assignment to switches in cellular mobile networks is taken as a case.

Index Terms: *Memetic algorithm(MA), NP-complete, NP-hard, combinatorial optimization, VLSI floor planning.*

1 INTRODUCTION

With increasing power of computers, GAs in combination with neural networks are one obvious method for micro-analytic simulation of evolutionary systems [1] by modeling interactions between learning and evolution. GAs have been used in wide variety of optimization tasks including combinatorial optimization problems including circuit partitioning, placement and clustering [2].

Genetic algorithms generally do not guarantee global optimum solution to a problem but are good at finding acceptably good solution to problems. In complex combinatorial spaces hybridization with other optimization techniques can greatly improve the efficiency of search. The main positive effect of hybridization is the convergence speed to the Pareto front and the main negative is the increase in computation speed. So, if the available computation time is limited than the number of generations is decreased. Memetic Algorithms apply separate local search process to refine individuals i.e. improve their fitness by hill climbing. Under different contexts and situations MAs are also known as hybrid evolutionary Algorithms and genetic local searchers. Combining global and local search is a strategy used by many successful global optimization approaches [3,4,5] and MAs have, in fact, been recognized as a powerful algorithmic paradigm for evolutionary computing. In particular, the relative advantage of MAs over GAs is quite consistent on complex search spaces.

Memetic algorithms combines global and local search by using evolutionary algorithms to perform exploration while the local search methods are used for exploitation [6]. Exploration is the process of visiting entirely new regions of a search space where the gain can be high. Exploitation on the other hand concentrates on previously visited points to maximize the gain i.e. the determination of which places might be profitable to visit next. A purely random search is good at exploration whereas a purely hill-climbing method is good at exploitation. Combinations of these two strategies can be quite effective, but it is difficult to know where the best balance is set. So, one of the main objectives in implementing any MA is the means of achieving both techniques during the search. It is important to understand that injecting constructive initial solutions within a population is a form of local search. Also, the concept of clustering used to smooth the landscape being searched can be considered a different form of iterative improvement embedded within the Memetic Algorithm.

After a brief review of Gas we first discuss GA for VLSI partitioning. Secondly, MA for VLSI partitioning is elaborated. Thirdly, comparisons on the basis of results in the two cases based on standard test sets are presented. Fourthly, the paper discusses the balancing issue between genetic search and local search in MAs for multi-objective permutation flow shop scheduling. Lastly a case of MA for assigning cells to switches in cellular and mobile network is also discussed.

2 GENETIC ALGORITHM

Genetic Algorithm (GA) are a class of Evolutionary Algorithms (EA) that seek improved performance by sampling areas of the sample (parameter) space having a high probability of leading to good solutions. A chromosome represents a potential solution within the solution space [1]. These chromosomes undergo transformation by using a kind of "natural selection" together with genetics inspired operators of crossover, mutation and inversion. Each chromosome consist of "genes" (e.g. bits) each gene being an instance of particular "allele" (e.g. 0 or 1). The Selection operator chooses those chromosomes in the population that will be allowed to reproduce and on an average the fitter chromosome produce more offspring than the less fit ones. Crossover exchanges the subparts of two chromosomes. Crossover means the same as recombination. Mutation randomly changes the allele values of some locations in the chromosome and Inversion reverses the order of a contiguous section of chromosome, thus rearranging the order in which genes are arranged.

```
                 GENETIC ALGORITHM
   1. Solution space encoding
   2. (a)set size-of-popu, total_gen, generation=0;
      (b) set cross-over-rate, mutation_rate;
   3. Population Initialization
   4. While total_gen ≥ generation
Evaluate Fitness (number of cuts)
   For (k=1 to size-of-popu)
      Select (first_mate, second_mate)
      if (random(0,1) ≤ cross-over-rate)
            offspring = Crossover(first_mate,
                                  second_mate);
      if (random(0,1) ≤ mutatation_rate)
            offspring = Mutate();
      Repair offspring if required
   End For
   Add offsprings to the new generation.
   generation = generation + 1
End While
   5. Return back the best chromosomes
```

Fig. 1: The Genetic Algorithm for Circuit Partitioning

2.1 Genetic Algorithm of Circuit Partitioning

We have used a single point crossover technique, but a multipoint crossover (3-point and 4-point) works best for circuit partitioning problem. Also, roulette wheel parent selection method is used which is conceptually the simplest stochastic selection technique. The generation replacement technique is based on replacing the most inferior member in the population by new offspring.

3 MEMETIC ALGORITHM

The local search algorithms use greedy rather than steepest policy and work on principle of searching a neighborhood as a means of identifying a better solution. They continue until a local optima is found. This may take a long time. Many of the local search procedures embedded within the MAs are not standard, i.e. they usually perform a shorter truncated local search.

3.1 Memetic Algorithm of Circuit Partitioning

Simple local search techniques are embedded with GA to improve the performance. As GAs are not suitable for fine tuning solutions which are close to optimal, we apply a local improvement operator into recombination step of GA. After crossover, GA applies local optimization process on the offspring.

```
                 SDHC HEURISTIC
   Pass=0
   While (Stopping criteria is not met)
     Pass= Pass+1
     START DESCENT ROUTINE
       Mark all nodes as not yet moved
       While (Modules can be moved)
             Select node a_i with highest gain
             If balance criteria is OK
               Move a_i to destination block
               Mark a_i as locked
             End if
       End While
       Choose k nodes which maximize G
       Perform move on nodes a_1 to a_k
     END DESCENT ROUTINE
     START ASCENT ROUTINE
       Mark all nodes as not yet moved
       While (Modules can be moved)
             Select node a_i with lowest gain
             If balance criteria is OK
               Move a_i to destination block
               Mark a_i as locked
             End if
       End While
       Choose k nodes which minimize G
       Perform move on nodes a_1 to a_k
     END ASCENT ROUTINE
   End While

Record Best Solution
```

Fig.2: Simple Dynamic Hill Climbing Heuristic

3.1.1 *SIMPLE DYNAMIC HILL CLIMBING (SDHC):*

This technique is a refinement over local search heuristic as it overcomes the main drawback of local heuristic i.e. the immediate area around the current initial solution is the main focus and thus all regions of solution space are not explored and so these heuristics do not converge to optimal or near optimal solution unless they begin from good optimal solution.

As shown in the algorithm the heuristic continues to explore new regions until either cycling occurs or a certain number of passes have elapsed. A comparison of performance of the Sanchis heuristic to that of SDHC is shown in the figure. It is indicated that once the Sanchis interchange technique stops at a local minima, SDHC focuses the search on other parts of the solution space in order to ensure that other regions are explored. The complexity of SDHC is similar to Sanchis algorithm. The main objective of SDHC is to explore small regions effectively in relatively short duration of time.

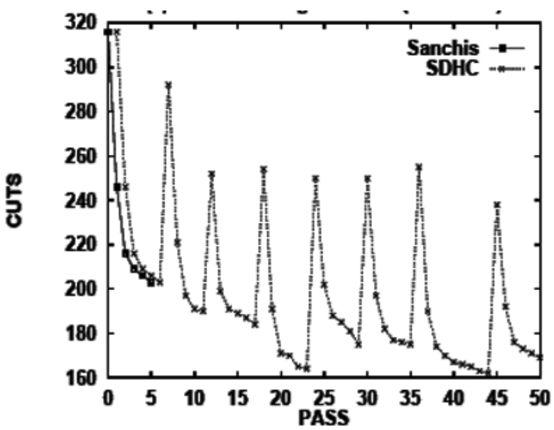

Fig. 3: Comparison of Sanchis Algorithm to SDHC Heuristic

3.1.2 MEMETIC ALGORITHM:

As depicted in figure 6 it starts with the GA techniques followed by SDHC with and without relaxation of size constraints. It is basically the combination of both Genetic Algorithm and SDHC Heuristic. If the relaxation of size constrains are used then algorithm either applies FM or DHC on the population for limited number of passes.

MEMETIC ALGORITHM

```
While (Gen ≤ Gen_size)
  Apply Generic GA
  If Relax Size Constraint
    Relax Size of Required Partitions
    If Apply Local Search
      Apply FM Local Search to Population
      Else
        Apply SDHC to Population
        Enforce Size constraint Greedy
    Else
  Apply SDHC or FM to Population
End While
```

Fig. 4: Memetic Algorithm for Circuit Partitioning

3.2 Memetic Algorithm for Assigning Cells to switches in Cellular Mobile Networks:

Here a simple notation is introduced to represent cells and switches, and for encoding chromosomes and genes. A non-binary representation of chromosomes was taken. In this representation, the genes represent the cells in the network, and the integers they contain represent the switch to which each cell is assigned.

The first element of the initial population is the one obtained when all cells are assigned to the nearest switch. The first chromosome is created therefore in a deterministic way. The creation of other chromosomes of the population is probabilistic. All chromosomes of the population verify the unique assignment constraint, but not necessarily the constraint of switches' capacity.

The MA is controlled by many parameters that affect their efficiency and accuracy. The population size=100, number of generations=800, number of cycles=10, crossover probability=0.9, and mutation probability=0.08.

Tabu search was executed by supposing that the cells are arranged on a hexagonal grid of almost equal length and width. The antennas are located at the center of cells and distributed evenly on the grid. However, when two or several antennas are too close to each other, the antenna arrangement is rejected and a new arrangement is chosen. The cost of linking a cell to a switch is proportional to the distance separating both. Without loss of generality, a proportionality coefficient was taken equal to the unit. The call rate of a cell *i* follows a gamma law of average and variance equal to the unit. The call duration inside the cells are distributed according to an exponential law of parameter equal to 1. This is justified by the fact that we are considering a simple model. If a cell *j* has *k* neighbors, the [0,1] interval is divided into *k+1* sub-intervals by choosing *k* random numbers distributed evenly between 0 and 1. At the end of the service period in cell *j*, the call could be either transferred to the i^{th} neighbor $(i=1\ldots\ldots k)$ with a handoff probability r_{ij} equal to the length of i^{th} interval, or ended with a probability equal to the length of the $k+1^{th}$ interval. To find the call volumes and the rates of coherent handoff, the cells are considered as *M/M/1* queues forming a Jackson network.

3.3 Memetic Algorithm for VLSI Floorplanning

Initially, the MA randomly generates a population of individuals. Then, the MA starts evolving the population generation by generation. In each generation, the MA uses the genetic operators probabilistically on the individuals in the population to create new promising search points (admissible floorplans) and uses the local search method to optimize them if the fitness of the admissible floorplans is greater than or equal to *v*. The process is repeated until a preset runtime is up. An outline of the MA is as follows:

1) $t := 0$;

2) generate an initial population $P(t)$ of size *PopS ize*;

3) evaluate all individuals in $P(t)$ and find the best individual *best*;

4) while the preset runtime is not up:

a) $t := t + 1$;

b) for each individual in $P(t)$:

i) this individual becomes the first parent $p1$;

ii) select a second parent using roulette wheel selection $p2$;

iii) probabilistically apply crossover to produce a child $c1$;

iv) if $fitness(c1) \geq v$, then optimize $c1$ using the local search method;

v) if $fitness(c1) \geq fitness(p1)$, then $p1 := c1$;

vi) if $fitness(c1) \geq fitness(best)$, then $best := c1$;

vii) probabilistically apply the two mutators (picked up randomly) on $c1$ to

produce a new individual f;

viii) if $fitness(f) \geq v$, then optimize f using the local search method;

ix) if $fitness(f) \geq fitness(p1)$, then $p1 := f$;

x) if $fitness(f) \geq fitness(best)$, then $best := f$.

5) output *best*.

4 BALANCE BETWEEN GENETIC AND LOCAL SEARCH

In this section we examine the effect of the balance between genetic search and local search on the search ability of our algorithm. The problem is how to allocate the available computation time wisely between genetic search and local search. This problem has been studied in the field of single objective hybrid i.e. memetic algorithm [8]. Goldberg and Voessner [9] presented a theoretical framework for discussing the balance between genetic search and local search. Hart [10] investigated the following for questions for designing efficient memetic algorithms for continuous optimization.

a. How often should local search be applied?
b. On which solution should local search be used?
c. How long should local search be run?
d. How efficient does local search need be?

Hart's study was extended to the case of combinatorial optimization by Land [11] where the balance between genetic search and local search was referred to as the local/global ratio. The balance can also be adjusted by the use of different neighborhood structures. Krasnoger [12] investigated how to change the size and the type of neighborhood structures dynamically in the framework of multimeme memetic algorithms where each meme had a different neighborhood structure, a different acceptance rule, and a different number of iterations of local search.

5 CONCLUSIONS

In this paper we have discussed the advent of genetic algorithms and the memetic algorithms as a solution to combinatorial optimization problems. Also, we discussed very briefly the key aspects involved in striking a balance between GA and local search in the MA. A problem of combinatorial optimization using GA and MA for VLSI circuit partitioning has been discussed. Looking at this the improvement in the speed of search process finds a great scope in the IT applications where the resources are fast approaching a dead end.

6 REFERENCES

[1] Melanie Mitchell, *An Introduction to Genetic Algorithms*, Prentice-Hall of India, EEE, 2004

[2] Maolin Tang, Xin Yao, "A Memetic Algorithm for VLSI Floorplanning", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 37, No. 1, February 2007

[3] "First workshop on memetic algorithms (WOMA I)," *Proc. 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pp. 95–130, July 8, 2000.

[4] "Second workshop on memetic algorithms (WOMA II)," *Proc. 2001Genetic and Evolutionary Computation Conf. Workshop Program*, pp.137–179, July 7, 2001.

[5] *Proc. 3rd Workshop on Memetic Algorithms (WOMA III)*, Sept. 7, 2002.

[6] Shwaki Arebi, "An Integrated Genetic Algorithm With Dynamic Hill Climbing for VLSI Circuit Partitioning",*Proc. 2000 Genetic and Evolutionary Computation Conf. Workshop Program*, July 2000.

[7] Hisao Ishibuchi, Tadashi Yoshida, Tadahiko Muruta, "Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling", *IEEE Transactions on Evolotionary Computation,* Vol. 7, No. 2, April 2003.

[8] D. E. Goldberg and S. Voessner, "Optimizing global-local search hybrids," in *Proc. 1999 Genetic and Evolutionary Computation Conf.*, Orlando, FL, July 13–17, 1999, pp. 220–228.

[9] D. Orvosh and L. David, "Shall we repair? Genetic algorithms, combinatorial optimization, and feasibility constraints," in *Proc. 5th Int. Conf. Genetic Algorithms*, Univ. of Illinois at Urbana-Champaign, July 17–21, 1993, p. 650.

[10] W. E. Hart, "Adaptive global optimization with local search," Ph.D. dissertation, Univ. of California, San Diego, 1994.

[11] M.W. S. Land, "Evolutionary algorithms with local search for combinatorial optimization," Ph.D. dissertation, Univ. of California, San Diego, 1998.

[12] N. Krasnogor, "Studies on the theory and design space of memetic algorithms," Ph.D. dissertation, Univ. of the West of England, Bristol, U.K., June 2002.