Integrating Genetic Algorithm with a Tabu Search (GTA) for Network Traffic Scheduling

R.Thamilselvan,
Kongu Engineering College/Computer Science and Engineering, Erode, India

P.Balasubramanie Ph.D
Kongu Engineering College/Computer Science and Engineering, Erode, India

ABSTRACT

Routing in networks is the major issue, because the packets may be sent through different topologies. This issue is addressed with the help of the metaheuristic algorithm. In the proposed work the Genetic Algorithm (GA) is integrated with Tabu Search to schedule the packets effectively in computer networks. The initial path is generated by Genetic Algorithm and then it is optimized by Tabu Search (TS) Algorithm. The importance of integration is that, to reduce the size of routing table.

Keywords:
Genetic Algorithm, Tabu Search, Routing.

1. INTRODUCTION

Routing is a process of transferring packets from source node to destination node with minimum cost. Hence the algorithm for routing has to organize and distribute the information about the network states. Routing algorithms generate feasible routes between source node and destination node to achieve the minimum cost. Routing in conjunction with congestion control and admission control defines the performance of the network. Routing algorithm should have generic objective of routing strategy to be both dynamically reconfigurable and based on locally available information.

The purpose of this proposed work is to generate solution using genetic and optimize the solution using tabu search algorithms. Genetic algorithm is one in which the population associated with each node co-evolve to solve the problem as whole.

Routing is a process of finding paths between source and destination node. There are mainly two types of routing policies viz., static routing and dynamic routing. In static routing, the routes between the nodes are precomputed based on certain factors and are stored in routing table [1]. In static routing, all the packets between any two nodes follow the same path. If topology of the network changes, then the path between two nodes may also change, and hence in dynamic routing policy, the routes are not stored but are generated when required. The new routes are generated based on the factors like traffic, link utilization etc which is aimed at having maximum performance.Routing policy may be centralized or distributed. In the case of centralized routing, only centralized node, generates routes between any pair of nodes. In distributed routing, each node generates routes independently between pair of nodes as and when required.
During the process of route generation, two types of packets are used viz., routing packets and data packets. Routing packets are given highest priority when compared to data packets. Each node in the network is of the type store and forward. The link performance may be measured in terms of bandwidth or link delay. The topology of the network may change due to growth in number of nodes, reconfiguration or failure of node. This change in topology should be reflected in the routing table, which in turn helps the routing protocol
to generate optimal route for the current state of network.

Some of the protocols are Resource Information Protocol (RIP), Interior gateway routing protocol (IGRP), Open source shortest path first (OSPF) and Border gateway protocol (BGP). RIP is a distance vector routing protocol, with hop count as metric for path selection. The change in
network topology is reflected by broadcasting route updates. IGRP is also distance vector routing protocol for autonomous system. It supports multiple metrics for each
route like bandwidth, delay, load and MTU (Maximum Transfer Unit). It also reflects the change in network by broadcasting route updates. OSPF is a link state routing protocol which uses shortest path first algorithm to compute low cost route to destination. Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced distance vector routing protocol with optimization to minimize the effect of change in topology and efficient use of bandwidth and processing power at the router. It uses unequal cost load balancing.

2. RELATED WORK

A. Genetic Algorithm (GA)

GA is an evolutionary technique for large space search. The general procedure of GA search is as follows:

1) Population generation: A population is a set of *chromosomes* and each represents a possible solution, which is a mapping sequence between tasks and machines. The initial population can be generated by other heuristic algorithms, such as Min-min (called seeding the population with a Min-min chromosome).
2) Chromosome evaluation: Each chromosome is associated with a fitness value, which is the makespan of the task-machine mapping this chromosome represents. The goal of GA search is to find the chromosome with optimal fitness value.
3) Crossover and Mutation operation: Crossover operation selects a random pair of chromosomes and chooses a random point in the first chromosome. For the sections of both chromosomes from that point to the end of each chromosome, crossover exchanges machine assignments between corresponding tasks. Mutation randomly selects a chromosome, then randomly selects a task within the chromosome, and randomly reassigns it to a new machine.
4) Finally, the chromosomes from this modified population are evaluated again. This completes one iteration of the GA. The GA stops when a predefined number of evolutions is reached or all chromosomes converge to the same mapping.

This genetic algorithm randomly selects chromosomes. Crossover is the process of swapping certain sub-sequences in the selected chromosomes. Mutation is the process of replacing certain sub-sequences with some task-mapping choices new to the current population. Both crossover and mutation are done randomly. After crossover and mutation, a new population is generated. Then this new population is evaluated, and the process starts over again until some stopping criteria are met. The stopping criteria can be, for example, 1) no improvement in recent evaluations; 2) all chromosomes converge to the same mapping; 3) a cost bound is met.

B. *Tabu Search (TS)*

TS is implemented beginning with a random mapping as the initial solution, generated from a uniform distribution. To manipulate the current solution and move through the solution space, a *short hop* is performed. The intuitive purpose of a short hop is to find the nearest local minimum solution within the solution space. The basic procedure to perform a short hop is to consider, for each possible pair of tasks, each possible pair of machine assignments, while the other assignments are unchanged. If the new makespan is an improvement, the new solution is saved, replacing the current solution. The short hop procedure ends when (1) every pair-wise remapping combination has been exhausted with no improvement, or (2) the limit on the total number of successful hops (*limithops*) is reached. When the short hop procedure ends, the final mapping from the local solution space search is added to the tabu list. The tabu list is a method of keeping track of the regions of the solution space that have already been searched. Next, a new random mapping is generated, and it must differ from each mapping in the tabu list by at least half of the machine assignments (a successful *long hop*). The intuitive purpose of a long hop is to move to a new region of the solution space that has not already been searched. After each successful long hop, the short hop procedure is repeated. The stopping criterion for the entire heuristic is when the sum of the total number of successful short hops and successful long hops equals *limithops*. Then, the best mapping from the tabu list is the final answer.

Problem Definition

The network under consideration is represented as G = (V, E), a connected graph with N nodes. The metric of optimization is cost of path between the nodes. The total
cost is the sum of cost of individual hops. The goal is to find the path with minimum total cost between source node Vs and destination Vd, where Vs and Vd belong to V. This paper presents the efficient on-demand, source initiated routing algorithm using Tabu search algorithm and genetic algorithm. It is implemented as two modules viz., Network generation or initialization of routing tables using Genetic algorithm and generation of optimal path using tabu search algorithm. Finally data is sent along the generated path.

3. PROPOSED ALGORITHM

A. *Combined Heuristics*

GA can be combined with TS to create combinational heuristics. For example, The Genetic Tabu Search Algorithm (GTA) heuristic is a combination of the GA and TS techniques. In general, GTA follows procedures similar to the GA outlined above. However, for the selection process, GTA uses the Tabu search process.

These Nature's heuristics were only relatively introduced into the scheduling area and more work needs to be done to fit them in a Grid context. There are a lot of interesting questions. First, the meaning of controlling measurements in such heuristics needs to be refined. For example, each possible pair of tasks, each possible pair of machine assignments, while the other assignments are unchanged. If the new makespan is an improvement, the new solution is saved, replacing the current solution. Second, there is a trade-off between the search cost and the degree of optimality of solutions found. For example, in a genetic algorithm, historical knowledge can be used to guide the chromosome selection, crossover, or mutation process so that the search process can converge quickly. But this adjustment seems to contradict the philosophy of an evolutionary algorithm: randomization and diversity generate better results, and it may not bring a better solution.

Experiments have shown that a good initial solution for TS improves both the quality of the solution as also execution time. We require *m* initial solutions for distributing among *m* nodes; we choose to combine TS with GA.

3.1.Crossover

The crossover operator involves the swapping of genetic material (bit-values) between the two parent strings. Two parents produce two offspring. There is a chance that the chromosomes of the two parents are copied unmodified as offspring. There is a chance that the chromosomes of the two parents are randomly recombined (crossover) to form offspring. Generally the chance of crossover is between 0.6 and 1.0[21]. The following sections propose the new crossover algorithms for job shop scheduling.

Unordered Subsequence exchange crossover (USXX)

We introduce one more cross over strategy named as Unordered Subsequence Exchange Crossover (USXX) that children inherit subsequences on each machine as far as possible from parents. Unordered Subsequence exchange crossover creates a new children's even the subsequence of parent1 is not in the same order in parent2. The algorithm for USXX is as follows.

Step 1. Generate two random parent individual namely P1 and P2 with a sequence of all operations.
Step 2. Generate two child individual namely C1 and C2.
Step 3. Select random subset of operations (genes) from P1 and copy it into C1.
Step 4. Starting from the first crossover point from P1, look for elements in P2 that have been copied as in the same order.
Step 5. The remaining operations of P2 that are not in the subset can be filled in C1 so as to maintain their relative ordering.
Step 6. If C1 is created then goto Step 3 to generate C2 analogously.

3.2 Mutation

Step 1. It may be possible that crossover operation may produce

Step 2. degenerate population. In order to undo this, mutation

Step 3. operation is performed. Mutation operation can be

Step 4. inversion, insertion, reciprocal exchange or others. In case

Step 5. of inversion two random points are selected and the string

Step 6. between them is reversed. In case of insertion a node is

Step 7. inserted at random position in the string. In reciprocal

Step 8. exchange, nodes at two random positions are exchanged

Step 9. This paper uses insertion.

This is because a node along the optimal path may be eliminated through crossover. By using insertion, it can be brought back. Once mutation is completed, the offspring generated by mutation have to be validated with the same technique used in crossover. Once the optimal solution is generated using GA, data is transmitted along that path. There may be change in topology of network as some nodes may join the network or some nodes may leave the network or some nodes may fail. The change in network may be detected by the algorithm at regular intervals, ΔT seconds. Under these circumstances the optimal path may no more be the shortest. Hence the network has to be refreshed at every ΔT seconds and new routes may be generated.

Table 3.1 Genetic Algorithm pseudo code

```
1. Choose initial population
2. Evaluate the fitness of each individual in the
   population
3. Repeat until termination: (time limit or sufficient
   fitness achieved)
   a. Select best-ranking individuals to reproduce
   b. Breed new generation through crossover and/or
      mutation (genetic operations) and give birth to
      offspring
   c. Evaluate the individual fitnesses of the offspring
   d. Replace worst ranked part of population with
      offspring
```

Table 3.2 Genetic Tabu Search Algorithm

```
1. Central node generates n initial solutions using GA. It
   runs GA for fixed number of iterations, t.
   a. Choose initial population of fixed size and set j=1
   b. While(j<=t)
      Begin
        i. Apply the operator on the two parent
           schedules chosen randomly to produce two
           offspring and replace the parents by the best
           two the four schedules.
       ii. J=j+1
           End
2. Central node sends m best solutions chosen to the m
   remote worker nodes
3. Each worker node runs the TS algorithm by using the
   initial state received.
4. Upon receiving a converged result from one of the
   worker nodes, the central node stops execution.
```

In GA, an initial population consisting of a set of solution is chosen and then the solutions are evaluated. Relatively more effective solutions are selected to have more off springs which are, in some way, related to the original solutions. If the genetic operator is chosen properly, the final population will have better solutions. GA improves the whole population. TS aim at producing one best solution. For the TS, we require several good initial solutions to ensure the required number of good initial solution. The pseudo-code for the GA and GTA is given in the Table 3.1 and Table 3.2 respectively.

4. EXPERIMENTS AND RESULTS

Both the above algorithms have been implemented by using Alchemi [12] running on a network of 5 workstations. It is built on a communication channel. The channel is possible to send task messages, result messages, configure messages and partial result messages, amount the various nodes of the network. The full implementation details of the algorithm are given in the following sections.

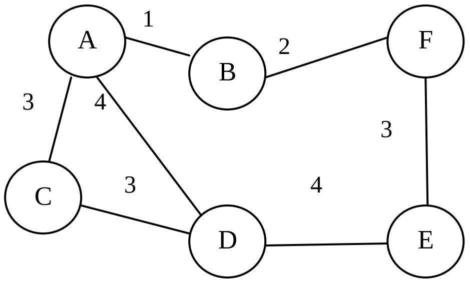

Fig 4.1: Sample Topology

Initially 10 random chromosomes are generated, out of which best 5 is considered for first generation. USXX Crossover and mutation functions are applied to generate next generation chromosomes. At each generation the chromosomes are validated and best fit chromosomes are sent to next generation. It is found that the fitness value increases at each generation. To optimize the result, the generated chromosome is optimized with tabu search.

Table 4.1: Generation of 10 chromosomes

chromosome						Delay	Fitness
A	B	E	C	F	D	1	8
A	B	C	D	E	F	1	7
A	C	B	D	E	F	3	10
A	C	D	E	F	B	15	17
A	B	D	C	E	F	1	8
A	B	D	C	E	F	1	9
A	B	C	D	E	F	1	9
A	F	C	B	E	D	0	7
A	D	E	F	C	B	11	15
A	B	D	C	E	F	1	9

Table 4.2 : Generation 1

chromosome						Nodes Visited	Delay	Fitness
A	C	B	D	E	F	2	3	10
A	C	D	E	F	B	2	15	17
A	B	D	C	E	F	6	1	9
A	B	C	D	E	F	2	1	9
A	D	E	F	C	B	4	11	15

Table 4.3 : Generation 2

chromosome						Nodes Visited	Delay	Fitness
A	B	F	E	D	C	6	13	15
A	D	E	F	B	C	5	13	17
A	C	D	E	F	B	6	15	9
A	B	C	D	E	F	1	1	9
A	D	E	B	F	C	3	7	12

After the path to all nodes from source node A is computed. The set of paths to a specific node will be displayed. Let the destination node is node D. Following is the set of paths from node A to node D.

Table 4.4 : Routes to the Destination D from A

Source	Destination	Delay	Route
A	D	4	Direct
A	D	6	C
A	D	10	B,F,E

Implementation of Genetic Tabu Search Algorithm

Table 4.1 Clustering Algorithm for Central Node

```
1. Initialize()
2. Generate m random initial states and assign to the m
   nodes of the network
3. Wait for results
4. Output_Results
```

Table 4.2 Clustering Algorithm for the Worker node

```
1. Get the subtask from the central node and $p$, the
   exchange parameter.
2. while ($p$ . 0)
   begin
           Simulated–annealing ($m$).
           Send the best solution obtained to the
   central node.
           $P$=p-2 (loop–iteration–value)$^{8}r$.
   end.
3. Run SA.
4. Send the converged value to the central node.
```

5. CONCLUSION AND FUTURE WORK

In this paper we have presented a Genetic Tabu Search Algorithm for the dynamic routing of packets from different network topologies. Here the genetic algorithms are used to generate a initial route from any source to destination and the Tabu search algorithm is used to find the optimal path between any source to destination node. Hence, the integration of these two algorithms, which makes the packets to explore the network independently and it helps to find the path between any two pair of nodes effectively. It can be improved with better crossover and mutation strategies of Genetic Algorithms.

6. REFERENCES

[1] K. Aggarwal and R. D. Kent, *An Adaptive Generalized Scheduler for Grid Applications*, in Proc. of the 19th Annual International Symposium on High Performance Computing Systems and Applications (HPCS'05), pp.15-18, Guelph, Ontario Canada, May 2005.

[2] M, Arora, S.K. Das, R. Biswas, *A Decentralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments*, in Proc. of International Conference on Parallel Processing Workshops (ICPPW'02), pp.:499 – 505, Vancouver, British Columbia Canada, August 2002.

[3] R. Bajaj and D. P. Agrawal, *Improving Scheduling of Tasks in A Heterogeneous Environment*, in IEEE Transactions on Parallel and Distributed Systems, Vol.15, no. 2, pp.107 – 118, February 2004.

[4] J Blythe, S Jain, E Deelman, Y Gil, K Vahi and A Mandal,K Kennedy, *Task Scheduling Strategies for Workflow-based Applications in Grids*, in Proc. Of International Symposium on Cluster Computing and Grid (CCGrid'05), pp.759-767, Cardiff, UK, May 2005.

[5] Greening, Daniel R., "Parallel Simulated Annealing Techniques", Physica D, Vol.42, pp. 293-306,1990

[6] Van Laarhoven, P.J.M., E.H.L., AArts, and Jan Karel Lenstra, "Job Shop Scheduling by Simulated Annealing", Operation Research, Vol. 40, pp. 113-125,1992.

[7] Syswerda, Gilbert, "Schedule Optimization Using Genetic Algorithms", L. Davis(ed.), Handbook of Genetic Algorithmsm pp.332-349,1991.

[8] Nowicki, E. and C.Smutnicki, "A Fast Tabu Search Algorithm for Job Shop Problem", Report 8/93, Institute of Engineering Cybernetics, Technical University of Wroclaw, 1993.

[9] Whitley, Darrel, Timothy, Starkweather and Daniel, Shaner, "Schedule Optimization Using Genetic Algorithms", Lawrence Davis, (ed.), pp.351-357

[10] Abraham, R. Buyya and B. Nath, Nature's Heuristics for Scheduling Jobs on Computational Grids, in Proc. of 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), pp. 45-52, Cochin, India, December 2000.

[11] Andresen, S. Kota, M. Tera, and T. Bower. An ip-level network monitor and scheduing system for clusters. In *Proceeding of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02)*, LasVegas.

[12] Most popular Grid computing web site www.gridbus.org.

[13] Most related articles web page www.buyya.com.