

# Multi Colony Ant System based Solution to Travelling Salesman Problem using OpenCL

Shobhit N Sharma  
Gyan Ganga College of Technology

Vikram Garg  
Gyan Ganga College of Technology

## ABSTRACT

Travelling salesman problem (TSP) finds applications in wide domains. It is a well known NP Hard problem. In this paper we have proposed GPU based implementation for TSP using OpenCL based on Multi colony Ant System. A comparative analysis is done among the standard travelling salesman problem, multi colony based implementation of travelling salesman problem and GPU based implementation. It is found that GPU based implementation is most efficient in terms of execution time and average tour length.

## Keywords

Travelling salesman problem , OpenCL , Graphical processing unit(GPU), Ant colony optimization.

## 1. INTRODUCTION

With these recent trends in the technologies and performance enhancements, it is more crucial to develop algorithms which are time efficient in addition to accuracy and other measures. Capabilities of hardware present in general computers are increasing day by day. With the recent trends in High definition video streaming and gaming demand of graphics card in systems has been a common thing. So with the availability of high processing graphics card in the systems, it's more important to use those graphics processing unit (GPU) to do some other tasks in addition to graphics processing. OpenCL [9] is one such parallel processing framework which uses heterogeneous environment (both CPU and GPU) for program execution. To make efficient utilization of hardware including CPU and GPU we have proposed our GPU based algorithm for TSP which will execute in parallel on GPU and will be controlled by host program on CPU.

## 2. TRAVELLING SALESMAN PROBLEM

In Travelling salesman problem [4][2] , a salesman has to construct a tour from his hometown to different cities such that he should visit every city exactly once and total tour length should be minimum. Such type of problem can be suitably represented using graphs. Given a Graph G in which vertices will represent cities and edges will represent distance between the respective cities. We have to find a tour starting from a vertex such that all the vertices of the graph should be visited exactly once before returning to initial vertex.

So travelling salesman problem is actually a problem to find Hamiltonian circuit within the graph G which is a NP complete problem.

### Constraints in travelling salesman problem :

- Tour length should be minimum
- And each vertex should be visited exactly once.

## 3. ANT SYSTEM

Ant colony optimization[1][2][6] was proposed in early 90's. The first algorithm proposed was Ant system (AS) by Dorigo subsequent to which many variants were proposed. All the variants use a general concept in which stochastic state transition rules which are function of pheromone value and heuristic information and two other parameters  $\alpha$  and  $\beta$  are used.  $\alpha$  and  $\beta$  are used to control the effects of pheromone and heuristic information so that in different scenarios we can adjust these two values accordingly and obtain the desired optimal result.

There are two main phases of every Ant inspired algorithm. These are :-

- Solution construction
- Pheromone update

**a) Solution construction :** Ant select next node to be cited using a stochastic rule. Ant 'k' at node 'i' has the following probability to select node j as next node at time 't'

$$p_{i,j}^k = \frac{[\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta}{\sum_{l \in N_i^k} [\tau_{i,l}]^\alpha [\eta_{i,l}]^\beta}, \quad \text{if } j \in N_i^k$$

Where,

$p_{ij}^k$  = Probability with which ant k moves from i node to j.

$\eta_{ij}$  = heuristic information.

$\tau_{ij}(t)$  = pheromone trail update.

And this heuristic information depends on the edge length.  $\alpha$  and  $\beta$  are adjustable parameters to adjust relative influence of probability and heuristic information.

**b) Pheromone update :** After all ants have constructed their tours pheromone trails are updated.

This is done by first lowering the pheromone value on all edges by a constant factor called evaporation rate :

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}, \text{ for all } (i,j) \in L$$

After evaporation all ants deposits pheromones on the edges they have visited in their tours.

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k, \text{ for all } (i,j) \in L$$

Where,

$$\Delta \tau_{ij}^k = \begin{cases} 1/C_k, & \text{if } (i, j) \in T_k(t) \\ 0, & \text{otherwise} \end{cases}$$

Where  $C_k$  is the length of the tour  $T^k$  built by ant  $k$ .

In Ant System the above process which comprises of solution construction and pheromone update is for a colony of ants. If there are more than one colonies of ants then both the colonies will have one local pheromone update matrix and will share one global pheromone update matrix.

#### 4. INTRODUCTION TO OPENCL - A GPU COMPUTING FRAMEWORK

OpenCL is an open standard framework for parallel programming composed of several computational resources (CPU, GPU and other processors). Thus one can achieve considerable acceleration in parallel processing if it utilises all the computational resources. The main advantage with OpenCL is its portability as it provides cross vendor software portability[11]. It consists of different models :

- a) Platform model.
- b) Execution model.
- c) Memory model.
- d) Programming model.

**Work Groups and Work items :** In OpenCL host program is the one which controls the flow of execution and it is responsible to call kernel[9] that is the code which will be executed in parallel for a number of work items. Group of work items for which kernel is executed in parallel is called work group[9]. In openCL workgroups are defined as :

```
global_work_size = {16,16}  
local_work_size = {8,8}
```

The above steps will form a global work group of 16x16 and there will be 4 local work groups in that work group each of size 8x8.

Here each local work group will act as colony of ant and each will find appropriate solution and best solution among all will be selected.

#### 5. RELATED WORK

Many solutions to Travelling salesman problem has been proposed in past. The most efficient among them were GPU implementations and when metaheuristics are applied to TSP.

In [2] serial and parallel implementation of TSP has been compared in terms of speedup based on ant colony optimization. In [3] also parallel implementation of TSP is proposed. In [7] local optimization has been applied to solve travelling salesman problem. In [8] travelling salesman problem is divided into a multi objective problem and it is solved by combinatorial algorithm like aco.

In all above papers single colony based approach to solve TSP is proposed. Which may fail due to stagnation. SO in this paper a multi colony ant based approach using proper stagnation techniques is proposed.

#### 6. IDENTIFIED PARALLELISM IN TSP

In Travelling salesman problem given a graph  $G$  and we have to find tour such that every city should be visited exactly once and we should return to starting node. There is inherent

parallelism in this problem as all the tours can be explored in parallel and then all these tours can be compared in terms of tour length and the tour having minimum length can be selected. So in this paper a parallel GPU based implementation for TSP is proposed. But as graphs can be large and may have large number of nodes and resulting in large number of tours to explore so it may be time consuming to explore all these tours so its more efficient to apply some metaheuristic algorithm to reduce the search space and to minimize the number of tours explored.

#### 7. APPLYING ACO ON TSP

In ACO, colony of ants move from nest to food source through different paths initially. While moving they lay a special chemical substance called pheromone. This Pheromone is used for ants indirect interaction i.e. stigmergy. Using this all ants select an optimal path from nest to food source after some time. This time period is called convergence time. Thus ant colony are capable of finding shortest path for their purpose which is to find optimal solution. In fact Ant Colony Optimization is a multi agent system as here multiple colonies of ant are used to come across a solution for the given problem[6]. The role of agents is depicted by artificial ants. These artificial ants explore for the solution space probabilistically to uncover different solutions and finally converge to find optimal solution out of these candidate solution.

So here in TSP we can use multiple colony based ant system approach is used where each colony of ants explore a path and then resulting tour length of all the tours returned by different colonies are compared and the tour with minimum tour length is selected.

#### Pheromone trails and heuristic information :

Here pheromone trail at any time  $t$   $\tau_{ij}(t)$  will represent willingness to visit city  $j$  after visiting city  $i$ . and heuristic information  $\eta_{ij}$  will be inversely proportional to the path length between city  $i$  and  $j$ .

#### 8. ALGORITHM FOR ANT SYSTEM BASED TSP ON GPU

1. initialize all edges to initial pheromone content.
2. Select Source city 'S'
3. For all the colonies of ants from 'S' in parallel do  
do  
    select next city to visit based on probability rule  
    add edge weight to tour length and city to visited city list  
until tour completed and all cities visited.
4. Host will compare tour lengths returned by all the colonies of ants and will return the minimum tour length.

#### 9. COMPARATIVE ANALYSIS AND RESULTS

All the results are computed on AMD Radeon HD 6450(GPU): 2 Compute units, 625 MHz clock, 2048MB Global Memory, 32KB Local Memory on a system having Intel Core i5 CPU 650 @ 3.2GHz and 2GB RAM

Here proposed algorithm is compared with standard implementation of travelling salesman problem in terms of accuracy and execution time.

Accuracy is and important aspect as we are dealing with metaheuristic algorithm.

Here in each ant colony there are 64 ants as local work group size taken is {8,8}

**Table 1. comparison of execution time for different number of ant colonies**

NO OF VERTICES IN GRAPH	NO OF ANT COLONIES	EXECUTION TIME (msec)	
		Seq_TSP	GPU_TSP
64	8	15	1
	16	21	2
	32	29	3
128	8	49	4
	16	51	6
	32	71	7
256	8	102	9
	16	132	12
	32	151	13
512	8	266	20
	16	311	21
	32	343	24
1024	8	417	36
	16	452	39
	32	491	44

**Table 2. comparison of accuracy for different number of ant colonies**

NO OF VERTICES IN GRAPH	NO OF ANT COLONIES	ACCURACY (%)
		GPU_TSP
64	8	95
	16	96.2
	32	99.1
128	8	94.1
	16	98.1
	32	99.5
256	8	95.7
	16	96.2
	32	99.7
512	8	96.8
	16	97.2
	32	99.9
1024	8	94.1
	16	98.1
	32	99.7

## 10. CONCLUSION

In this paper a multi colony ant system based GPU implementation of travelling salesman problem is proposed. Here analysis is done for graphs having different number of nodes generated by a random graph generator. And these graphs are solved for TSP using ant colonies of different sizes. Comparison of execution time and accuracy is done. It is found that GPU based implementation is faster and having speedup of 11x and accuracy % is 98.9.

## 11. REFERENCES

- [1] Laurence Dawson, Iain Stewart "Improving Ant Colony Optimization performance on the GPU using CUDA" in 2013 IEEE Congress on Evolutionary Computation June 20-23, Cancún, México
- [2] Elmedina Fejzagić, Adna Oputić "Performance Comparison of Sequential and Parallel Execution of the Ant Colony Optimization Algorithm for Solving the Traveling Salesman Problem" in MIPRO 2013.
- [3] Yuet Ming Lam, Shuang Ying Yu, Peng Wang "A Parallel Threads Coordination Scheme for Solving Combinatorial Optimization Problems" in IEEE 2013.
- [4] YingLi , Kai Ma, Jiong Zhang "An Efficient Multicore based Parallel Computing Approach for TSP Problems" in 2013 Ninth International Conference on Semantics, Knowledge and Grids
- [5] Wang Jiening, Dong Jiankang, Zhang Chunfeng "Implementation of Ant Colony Algorithm based on GPU" in 2009 Sixth International Conference on Computer Graphics, Imaging and Visualization.
- [6] Akihiro Uchida, Yasuaki Ito, Koji Nakano "An Efficient GPU Implementation of Ant Colony Optimization for the Traveling Salesman Problem" in 2012 Third International Conference on Networking and Computing.
- [7] Kamil Rocki, Reiji Suda "High Performance GPU Accelerated Local Optimization in TSP" in 2013 IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum.
- [8] Murilo Zangari de Souza, Aurora Trinidad Ramirez Pozo "A GPU Implementation of MOEA/D-ACO for the Multiobjective Traveling Salesman Problem" in 2014 Brazilian Conference on Intelligent Systems.
- [9] A. Munshi, B. R. Gaster, T.G. Mattson, J. Fung, D. Ginsburg, "OpenCL Programming Guide", Addison Wesley pub., 2011.