Solving Traveling Salesman problem

Saumya Bhatnagar
16338296

Abstract: Traveling Salesman Problem is a routing problem which looks for the shortest possible route that a salesman would take to visit a list of cities. It is a BP-hard problem. Artificial Bee Colony (ABC) is a metaheuristic approach in which a colony of artificial bees cooperates in finding good solutions for numerical optimization problems. ABC is adopted widely for use in several domains of solution optimization. This report discusses the parallelization of this metaheuristic method for solving TSP.

1. Introduction:

In the Travelling Salesman Problem, the shortest route that visits all points/cities in the problem must be found.

It is computationally hard to solve with brute force methods as the time taken grows at a rate of n!. Increasing the number of cities from n to 2n will cause the program to take 2n! * (2n-1)! * (2n-2)! ….. * (n+1)! times longer. Even with a small number of cities the time taken quickly becomes large. Using greedy search solving algorithm, the order pf the problem becomes logarithmic. Further, we discuss on parallelizing the same problem

The remaining report has six more sections: SECTION II sets the background of the problem. SECTION III shows the related work that has been done in the field. Next Sections discuss the various methods implemented in detail: Brute force Method for solving TSP, Greedy Solver, parallel ABC.

2. Background:

ABC Algorithm

For more understanding on our distributed ABC algorithm, this section provides the knowledge background of the ABC algorithm and its sequential method for solving the optimization problems.

The ABC algorithm assumes the existence of a set of computational agents called honey bees. The honey bees in this algorithm are categorized into three groups: employed bees, onlooker bees and scout bees. The colony is equally separated into employed bees and onlooker bees. Each solution in the search space consists of a set of optimization parameters which represent a food source position. The number of employed bees is equal to the number of food sources. In other words, there is only one employed bee on each food source. The quality of food source is called its "fitness value" and is associated with its position. The process of bees seeking for good food sources is the

process used to find the optimal solution. The sequential method of the ABC algorithm can be shown as Fig. 1 below.

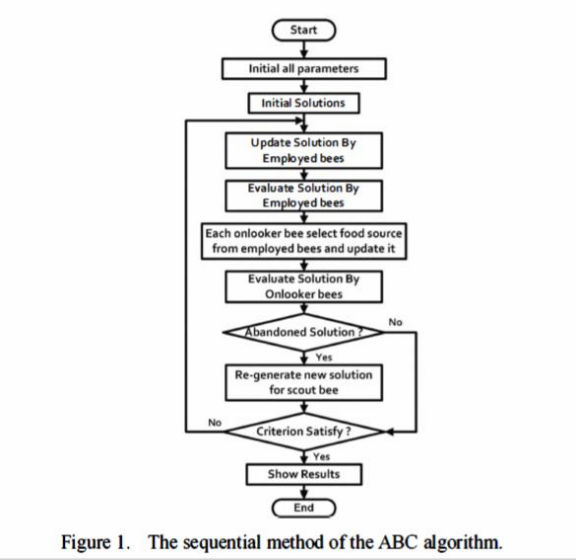

Figure 1. The sequential method of the ABC algorithm.

Traveling Salesman Problem

TSP can be modeling in the form of graphs. In this way, vertices can be modelled as cities to be visited by the salesman. The various edges of the graphs can be considered as the paths between cities, to be taken by the salesman. And the distance between two cities can be considered as the edge's weight.

On the basis of path weights, TSP can be modelled in two formats: Directed weighted graph & Undirected Weighted graph:

It is a minimization problem starting and finishing at a specified vertex after having visited each other vertex exactly once. Often, the model is a complete graph (i.e. each pair of vertices is connected by an edge). If no path exists between two cities, adding an arbitrarily long edge will complete the graph without affecting the optimal tour.

On the basis of direction, TSP can be modelled in two formats: Asymmetric model and symmetric model

In the symmetric TSP, the distance between two cities is the same in each opposite direction, forming an undirected graph. This symmetry halves the number of possible solutions. In the asymmetric TSP, paths may not exist in both directions or the distances might be different, forming a directed graph. Traffic collisions, one-way streets, and airfares for cities with different departure and arrival fees are examples of how this symmetry could break down. The below figure shows the symmetric TSP with four cities:

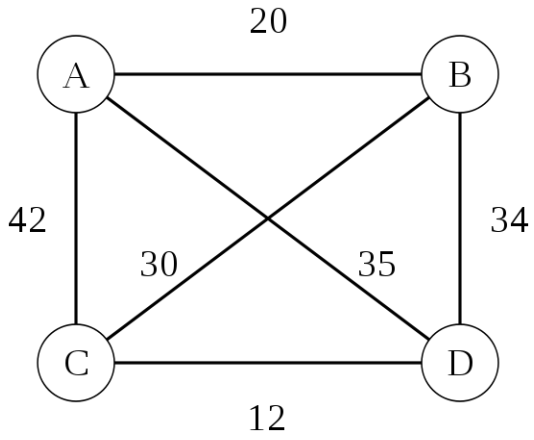

Figure 2. Symmetric TSP with four cities

3. Related work

Metaheuristic approaches, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO), are popular optimization methods that have been implemented on a parallel system. Many researchers have developed numerous parallel algorithms in order to increase the efficiency. To produce a faster GA based shortest part routing algorithm, a coarse-grained Parallel Genetic Algorithm was proposed by Yussof, et al.

In this work, all computing nodes randomly created their own sub-population and each executed sequential GA independently. The migration scheme was then employed to increase the sub-population diversity by exchanging solutions between computing nodes.

The parallel Ant Colony Optimization algorithm was presented by Jie, et al. In this work, the computational time was reduced and the significant speedup was obtained through coarse-granularity and partially asynchronous parallel technique. Based on this method, ant colony was divided into sub-colonies and each sub-colony was deployed on a computing node. Master node gathered the local best solution from all slave nodes and identified the node id with the best solution. Finally, the best pheromone matrix and solution were broadcasted to the rest of the participating nodes. Li and Wada introduced a parallel version of the Particle Swarm Optimization algorithm

for solving complex large-scale optimization problems. A communication method called delayed exchange was presented in the work. In this approach, each processor used its current local best solution and local best solutions computed previously by the other computing nodes to calculate the global best solution.

The Artificial Bee Colony (ABC) algorithm introduced by D. Karaboga was one of the metaheuristic approaches that has been used to find an optimal solution in several optimization problems. This algorithm is inspired by the behavior of honey bees when seeking a quality food source. It is simple in concept and easy to implement because it only uses one parameter called a "limit" for controlling the abandoned solution in its algorithm.

4. Brute Force method

Brute Force method examines all possible permutations of cities, keeping the one that is shortest.
Benefits of solving the TSP by brute force:

- Guaranteed to find a shortest tour.
- Since it looks at all possible tours and finds the shortest one, it is conceptually easy.

As n get bigger, n! grows worse than exponentially. If you want to use brute-force to solve a 20-city problem, then you'll need to generate 121645100408832000 different permutations, i.e. over 120 quintillion permutations. But that's nothing compared to the 99! different permutations you would need to search through for a 100-city problem.
Pseudocode of the brute force algorithm is as below:

- get an initial tour; call it T
- best_tour ⇦ T
- best_score ⇦ score(T)
- while there are more permutations of T do the following
    - generate a new permutation of T
    - if score(T) < best_score then
        - best_tour ⇦ T
        - best_score ⇦ score(T)
- print best_tour and best_score

The code for the same has been attached in the folder.
The corresponding files are:

main.c
brute.c
brute.h
Makefile

The graph is plotted and saved as: Task2.png

5. Greedy – Nearest Neighbor Algorithm

I have used Greedy - Nearest neighbor Algorithm to find an upper bound on the solution. The greedy nearest neighbour algorithm simply goes from the starting point, finds the nearest city and travels there, then finds the nearest unvisited city and travels there, and so on until all cities have been visited.
Benefits of solving TSP:

- Can be applied to very large TSP instances

Pseudocode is as below:

- Start with city 0 in the tour
- Look at the last city in the tour, and then add to the end of the tour the closest city not already in it.
- Repeat the previous step until all cities have been added.

The corresponding files are:

main.c
greedy.c
greedy.h
Makefile

6. ABC Algorithm

While applying greedy search algorithm, we fix the starting point. Thus, for very large problem, the analysis increases as for each city we are getting the min distance and comparing wit the previous route length computed. This shows multiple communications. So, for using parallelization the problem size should be large enough to show any improved speedup.

In figure one we noticed, the foraging behavior of bees is a distributed process in nature. Each forage bee seeks food sources concurrently around their hive. The collective decision is gradually made during the selection of food sources. This foraging behavior can thus be modeled straightforwardly for distributed environments.

The bees independently evaluate the quality of different new candidate food sources on their own by using information advertised by other bees. This concept is mapped into our proposed method. We divide bees into subgroups. Each subgroup then seeks a quality food source and exchanges this information with other subgroups.

The algorithm design has been shown in figure 3.

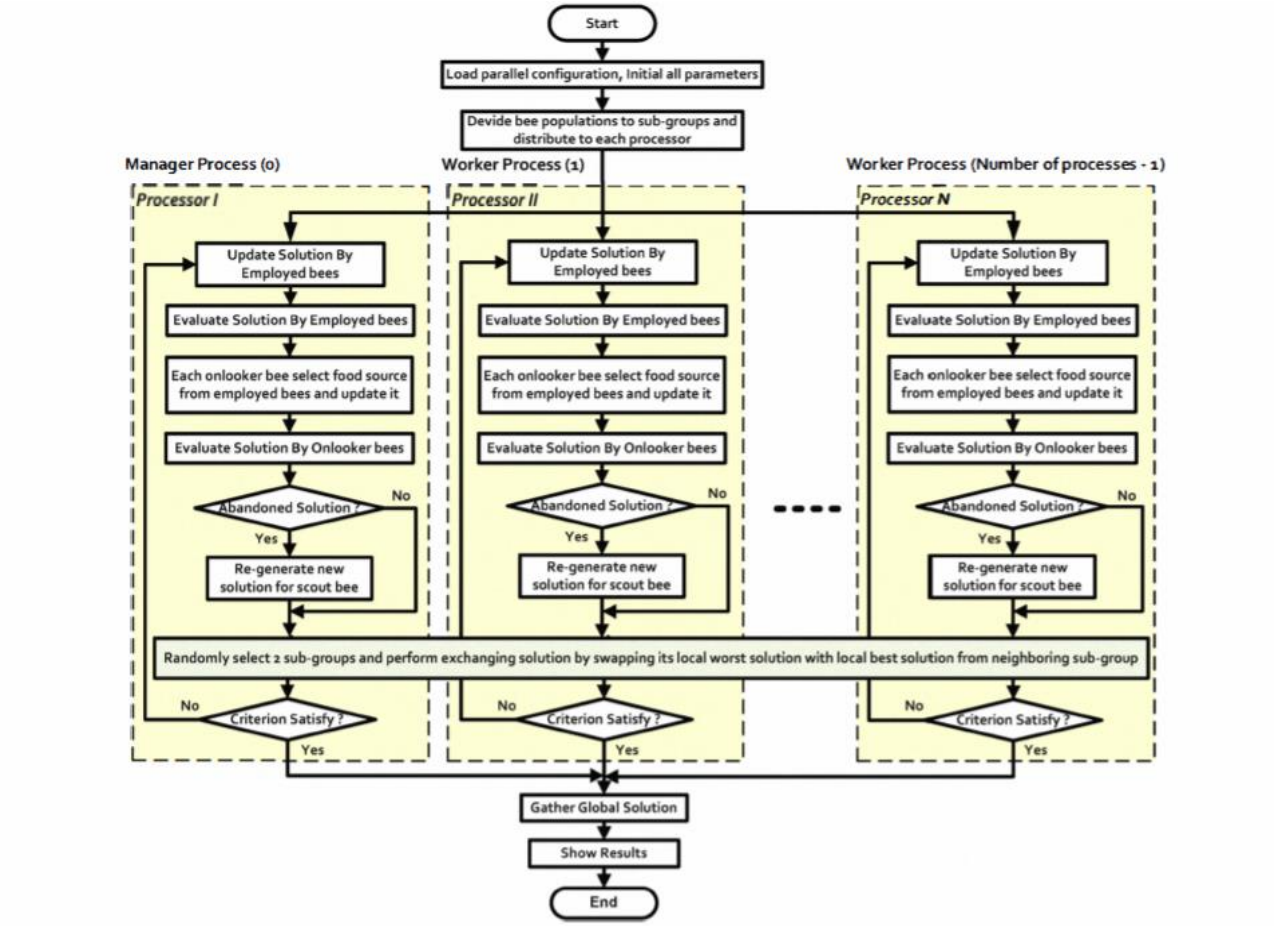

Figure 3: Design of ABC for distributed environment and implementation on TSP

While the sequential method uses only one random seed at the initial step, the distributed method employs a number of random seeds equal to the number of subgroups. This mechanism will increase the diversity of candidate solutions. In other words, the exploration process on search space will be improved and chance to [md the optimal solution is thus higher than the sequential method. Moreover, the solution can converge more quickly.

7. References:

1. Shin Siang Choong, Li-Pei Wong, Chee Peng Lim, "*An artificial bee colony algorithm with a modified choice function for the Traveling Salesman Problem"* [2017].
2. A Grama, A Gupta, G. Karypis, V. Kumar, *"Introduction to parallel computing"*. New York: Addison-Wesley, [2003].
3. S. Yussof, R.A Razali, O.H. See, AA Ghapar, M.M. Din, "*A Coarse-Grained Parallel Genetic Algorithm with Migration for Shortest Path Routing Problem*," IEEE International Conference on High Performance Computing and Communications, 2009, pp. 615-621.
4. D. Karaboga, "*An idea based on honey bee swarm for numerical optimization*," Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, Technical Report-TR06, [2005].