Parallel Computing Approach to Solve Travelling Salesman Problem

Harshala C. Ingole
Department of Computer Engineering
St. Vincent Pallotti College of Engineering and Technology
Nagpur, India

Vivek B. Kute
Department of Computer Engineering
St. Vincent Pallotti College of Engineering and Technology
Nagpur,India

ABSTRACT
Travelling Salesman Problem (TSP) is eminent in combinatorial optimization problem. A typical problem in computational mathematics, scientific and business application such as VLSI chip design, social network analysis. TSP, combinatorial optimization problem belongs to the class of NP-Hard, and becomes significant method of verifying the correctness and feasibility of new algorithm. With the accuracy results and efficient cutting branch strategy of branch and bound algorithm, it used to solve TSP. However, branch and bound algorithm not suitable for large scale TSP with sequential execution. In this paper parallel branch and bound algorithm has been improved and proposed to solve the symmetric TSP. This paper uses parallel program code based on multithreading concept to verify TSP. The experimental result shows our algorithm is efficient, and solves the large scale TSP problem which cannot be solved by sequential branch and bound.

General Terms
Parallel Computing, Combinatorial Optimization, Travelling Salesman Problem

Keywords
Travelling Salesman Problem, Branch and Bound Algorithm, Parallel Computing

1. INTRODUCTION
Travelling Salesman Problem (TSP), first expressed as computational mathematics problem in 1930.TSP, extensively studied problem in combinatorial optimization [1].Solution to this problem cannot be find in polynomial time. On solving optimization problem, requires to get the best possible solution from all available solution spaces. The "best" solution inferred that more than one solution available. The travelling salesman problem results in more than one solution, but the aim is to find the best possible solution for large scale TSP amongst all available solution spaces in a polynomial time and the performance also increased [2].TSP widely used in VLSI chip design, network routing, robot control, gene sequencing, vehicle routing [3]. And also finds its application in the areas like logistics, transportation, and semiconductor industries.

Methods of solving TSP can be categorized in two directions such as exact algorithm and approximate algorithm. These two methods give the solution but with certain issues, as Exact algorithms search for the whole solution space tree and obtain the global optimal solution. The global optimal solution guarantees the exact solution to the problem but not with the higher performance. E.g. branch-and-bound method, linear programming method, and dynamic programming method. As approximate algorithm finds as nearer as to the optimal solution in a reasonable amount of time but the solution does not guarantees the exact optimal solution to the TSP problem. E.g. greedy algorithm, genetic algorithms, simulated annealing algorithm, neural network algorithm and ant colony algorithm [2].With the comparison of this two methods of solving the TSP problem, the former one has requires the exponential time to solve and difficult to acquire the large scale problem. As Exponential algorithms have some advantages as simple method, small amount of calculation requires and so on. So feasible for small scale problem but as size of nodes get increased it doesn't give the nearer optimal solution in polynomial time.

To solve TSP in an acceptable time the parallel computing mechanism taken into account .The parallel computing in which the computations carried out simultaneously ,the principle behind that the computational task divides into subtask and solves independently and after completion results get combined.

This paper is organized as follows: Section II describes the background of TSP and branch and bound algorithm and related work. Section III describes the design and implementation of parallel branch and bound algorithm. In Section IV the branch and bound performance evaluation results in parallel execution is presented. The last section is our conclusion and future work.

2. BACKGROUND AND LITERATURE SURVEY
This section starts with introduction to TSP, Branch and Bound algorithm and then discusses literature survey.

2.1 TSP Introduction
Operation research and theoretical computer science addresses the TSP as NP-Hard problem [4]. TSP used to find the shortest path to travel through the given number of cities. Travelling Salesman Problem states that given a number of cities and the distances between them, salesman has to visit all the given cities exactly once and return back to the city from where he started with the minimized cost.

In TSP [5], given a complete undirected graph G = (V, E) that has nonnegative integer cost c(u.v) associated with each edge (u ,v) ϵ E ,and to find a Hamiltonian cycle (a tour) of G with minimum cost path. As an extension of notation, let c(A) denote the total coat of in the subset $A \subseteq E$:

$$C(A) = \sum_{(u,v) \in A} c(u,v) \quad (1)$$

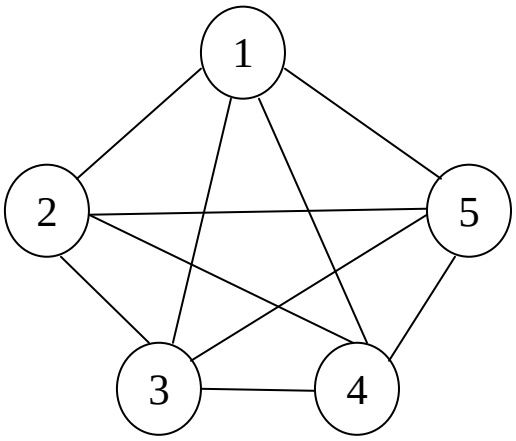

Fig.1. Complete Graph with five vertices

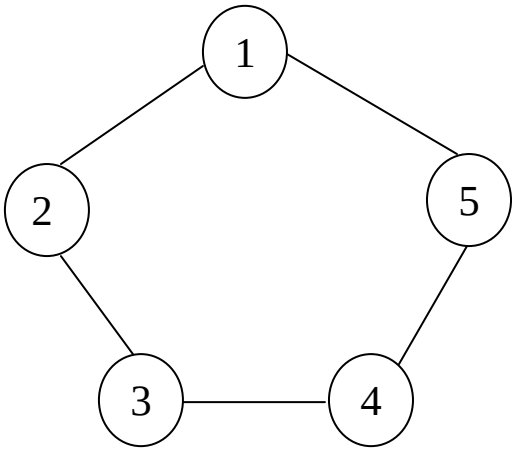

Fig.2. An Optimized Tour

2.2 Branch and Bound

In mathematical and combinatorial optimization, Branch and bound algorithm [6] used as a general algorithm for finding an optimal solution of various optimization problem. It makes a partition of the solution space of the given optimization problem and solve. The entire solution space represents as expansion tree, whose start point as root which is initially unsolved problem. The children at each node represent the subspaces obtained by branching, i.e. subdividing parent node; and the leaves of the tree represent nodes that cannot be subdivided any further, thus providing a final value of the cost function associated to a possible solution of the problem. It follows the non-exhaustive search for the solution space tree until an optimal solution to the initial problem found or to those that have possibility of being branched, becomes exhausted [7].

Cost Path: Used to calculate the cost of searched path. Cost path can be divided into two categories: one, cost of partial path when the search process is in progress and another one, the cost of total path when the search process is finished.

Objective Function Bound: Used to search the solution space tree. It can be divided into two categories: One is Objective Function Up Bound (OFUB), when cost of partial path of some partial path is greater than OFUB, needs to delete this path. Another one is Objective Function Low Bound (OFLB), when cost of total path of some total path is closest to OFLB, it is the best search path and that will be considered as global optimal solution to that optimization problem [3].

The branch and bound algorithm has two features: one is to estimate the bound value in advance. The other one is to search path while cutting branch and amending up bound value. It can improve search efficiency and find solution with higher performance.

2.3 Parallel Computing

Parallel computing implicitly coupled with the high speed and high performance computation. The main objective of implementing parallel computing is to reduce the time required to solve the combinatorial optimization problem and to improve the results performance. There are different methods of parallel computing as distributed computing, inter-process communication, message passing interface, p-threads, parallel algorithm, etc. Parallel computing achieved by the advanced hardware implementation, parallel implementation of the algorithm on single CPU system (multi-core system).

2.4 Literature Survey

Paper [2] proposed the survey of different approaches for TSP using Genetic Algorithm. With the literature survey found that parallel Genetic Algorithm can be implemented on Map/Reduce environment to solve the large scale problem and to improve the quality of the result.

The paper [3] proposed the multi-core based parallel computing approach to get the solution of Travelling Salesman Problem, used Beehive as multi-core platform to implement the problem. As Beehive architecture involved 16 cores with this only small scale TSP has been solved. In that, due to the limitation of timer in Beehive the experiment cannot the search time accurately when the nodes more than 15.Only test nodes is confined to 13.And again due to small memory space of Beehive, only solve up to 13 node city TSP problem and cannot solve the large scale TSP.

[7] Proposed the solution for travelling Salesman Problem with the use of CPAN Branch and Bound algorithm .CPAN or high level parallel composition is a set of parallel object of three type one object manager, the stages and the collector objects. In this paper, the real life example (TSP problem) GoodMan and Hedetniemi 1977 solved with CPAN Branch and Bound mechanism and compared, found the same result.

The proposed paper [8] uses branch and bound approach in which code matrix uses to calculate the low bound value. In this method, each branch needs a matrix and the child node uses code matrix of its parents. These methods can improve the traditional algorithm and improve the speed of cutting branch. But these types of algorithms are more complexity and get affected by the performance of the hardware. If the hardware configuration is not high, algorithm speed up effect not evident or the results cannot be obtained. On single core, it can improve the performance but as the number of nodes increased, the improved results are not obvious.

[10] In this paper the combination of genetic algorithm with dynamic programming uses for solving travelling salesman problem. In *CGADP*, the solutions obtained by genetic algorithm selected for applying a local search based on *DP*. The convergence rate of the solution found by *CGADP* is faster than that of *GA*. But with larger size problem, the running time is also increasing.

In [11], a simulated annealing algorithm performs better than the metropolis algorithm for any fixed temperature. Although the use of multi core technology can improve the efficiency of algorithm on dealing with large-scale TSP problem, these paper both refers to local optimal solution algorithm, e.g. genetic algorithms, simulated annealing algorithms. They can calculate the results quickly, but the results obtain for local search space thus doesn't get globally optimal solution.

In order to solve the above problems,try to use parallel method to realize branch and bound algorithm. Branch and bound algorithm is very suitable for distributed and parallel computing as it can be divided into independent sub-problems. The independent sub-problem or subtask can run parallel and afterward combined results will give the global optimal solution to the given TSP.

3. DESIGN AND IMPLEMENTATION

With the sequential execution of Branch and Bound algorithm it is possible to solve the small scale TSP. But with the increase number of nodes, the performance of the algorithm is not meet requirement and improvement is not obvious. So the realization of the algorithm environment should be changed to parallel branch and bound algorithm to improve the results and to solve for large scale problem.

The realization of parallel branch and bound algorithm involves many factors as it must have hardware environment to execute parallel programs, it can solve task allocation, load balancing, it can write multithreaded program which can be executed in parallel.

3.1 Parallel Implementation

This proposed work used the parallel branch and bound algorithm as parallel computing method. Parallel algorithm involves identifying the coding methods, making the code scalable in multiprocessing environment. The challenge is to design or redesign the code to run in parallel without making the CPU of one part wait for data from another, while keeping the resultant answer identical to the original coding. We use the concept of multithreading of Java programming language for the implementation of parallel branch and bound. As multi-threading support for the parallel execution of the code. Multithreading is the concept where the number of threads run parallel and obtain the solution.

While implementation of branch and bound algorithm in parallel B&B, it should be required that no data dependency. The branch and bound algorithm is suitable for parallel computing as it can be divided into independent sub-problems. And that independent sub-problem can act as independent thread and can run parallel and after the execution of all threads at each branch level comparing with bound value, the last global optimal solution obtained.

3.2 Pseudo Code for Parallel Branch and Bound

The pseudo code for parallel branch and bound algorithm for solving Travelling Salesman Problem is as follows:

Input: City list file

```
1.  Start {
2.   Vector v1<- declare and initialization of vector
3.   PriorityQueue<Tour> () work <-declare PriorityQueue to get tour
4.   For (k=0 to n)//first permutation vector
5.   v1 [k]<- k;
6.   Start<-Arrays.binarySearch( city, "city name") ;
      //starts from selected city
7.   PriorityQueue<-work.add(tour) // get the new
      generated tour and add into Queue
8.   While (!work.isEmpty()) // branch and bound
      loop ; do
9.   Tour current <- get the tour ;
10.   index <- get the current index ;
11.   v1 <- get the current solution ;
12.   If (index = n) //full permutation vector
13.   { If ( ( wt[ v1 [n-1]][ v1 [0]]>0) && (
        current.dist < bestTour) ) // is it return
        edge and better than earlier ?
14.     bestTour <- get the current distance ;
       //save the state in the list
15.     If (DEBUG) then accepted bound
16.   } else
17.   Path too long and rejected bound ;} else
18.   Thread thread1 <- new Thread( Runnable ()
19.   { Run()
20.    { if( lstIndicesWorked.contains(index)) return ;
21.     for (k = index ; k < n ; k++ )
22.     {swap(v1, index, k);
23.       If (wt[v1 [index-1]] [v1[index]]<0)
          continue ;
24.       work.add( tour (v1,index+1,wt) ;}
          //restore original permutation
25.       hold<-v1[index] ;
26.       for ( k = index+1 to n )
27.       v1 [k-1] <- v1[k];
28.       v1 [n-1] <- hold; });
29.    thread1.run(); }
30.  }
```

Fig.3. Pseudo code for parallel branch and bound

The pseudo code for solving TSP using parallel branch and bound algorithm uses the thread mechanism. Multiple threads

run parallel at the same branch level and bound value whichever obtained compare and proceed for next branch level until all nodes of the problem get visited and obtained the global optimized value i.e. the solution for that TSP.

4. EXPERIMENTAL RESULTS

The proposed parallel branch and bound algorithm uses parallel program code of multi-threading concept to achieve the parallel computing mechanism to solve combinatorial optimization problem as TSP. It is implemented in java, all experiments are conducted on Intel Core 2 Duo 2.10GHz processor with 3.00GB RAM in windows 7 operating system.

Table I shows the execution time required for files consisting of different number of cities of TSP problem. It contains number of nodes and the time required for execution using both sequential branch and bound and parallel branch and bound algorithm. In case of sequential branch and bound for TSP after 11 city problem it doesn't give the solution.

Table I The Snapshot Execution Time

Number of nodes	Time Required(millisecond)	
	Sequential branch and bound	Parallel branch and bound
5	94	40
7	2058	51
9	115581	66
11	1334010	85
15	-	136
20	-	222
24	-	308

Fig.4 is the comparison of sequential branch and bound and parallel branch and bound algorithm. X-axis indicates the number of cities and y-axis indicates time (in millisec.) required to get the result. Fig.4 proves the advantages of parallel computing. So in the experimental analysis, it used the same input file to be executed under sequential branch and bound and parallel branch and bound to analyze their execution time.

5. CONCLUSION AND FUTURE WORK

This paper proposed parallel computing approach for Symmetric Travelling Salesman Problem as parallel program code for branch and bound algorithm which uses a multithreading concept. With the experimental results, we demonstrate that our algorithm can solve large scale TSP (more than 20 city) with the optimum cost path in minimum time. This work has provided an approach for solving TSP using branch and bound algorithm using parallel computing mechanism. However, there are still some works not well studied. In future, this work can be extended to higher configuration systems. It can further evaluate for higher number of the city's problem. The work can be extended and evaluated for parallel and distributed environment.

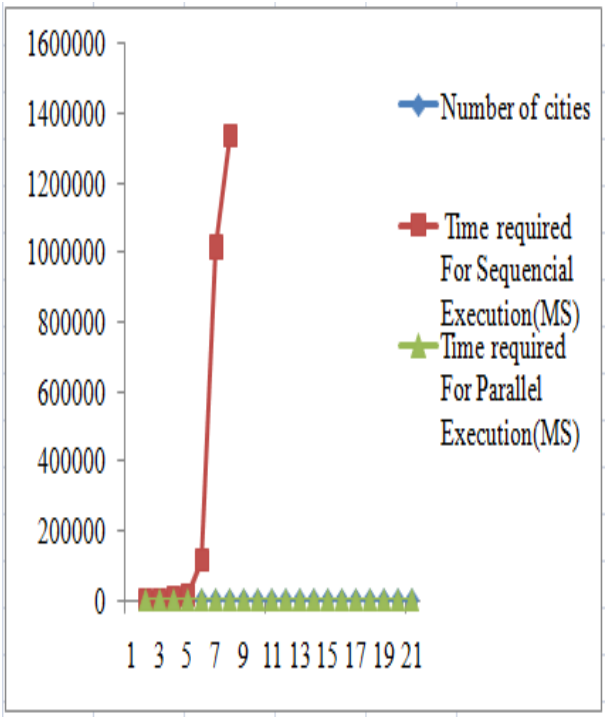

Fig.4. Comparison of sequential and parallel branch and bound

6. REFERENCES

[1] Anshul Singh, Devesh narayan "A Survey Paper on Solving Travelling Salesman problem Using Bee colony optimization" International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 5, May 2012). Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[2] Anitha Rao, Sandeep Kumar Hegde "Literature Survey On Travelling Salesman Problem" International Journal of Advance Research in Eduation Technology (IJARET) 42 Vol. 2, Issue 1 (Jan. - Mar. 2015) Using Genetic Algorithms.

[3] Kai Ma, Jiong Zhang "An Efficient Multicore based Parallel ComputingApproach for TSP Problems" 978-1-4799-3012-8/14 $31.00 © 2014 IEEEDOI 10.1109/SKG.2013.41.

[4] Klaus Meer, "Simulated Annealing versus Metropolis for a TSP instance", in Information Processing Letters, vol.104, 2007, pp. 216- 219.

[5] Thomas H.Cormen,Charles E. Leiserson,Ronald L. Rivest,Clifford SteinIntroduction To Algorithms (3rd Edition).

[6] M MostaFizur Rahman,Muhammad Foizul Islam Chowdhury, "Examining Branch and Bound Strategy on Multiprocessor Task Scheduling" 12th International Conference on Computers and Information Technology 2009

[7] Mario Rossainz Lopez Manuel , Capel Tunon "Design and Use of the CPAN Branch and Bound For The Solution Of Travelling Salesman Problem (TSP)" Proceedings 19th European conference On Modelling and Simulation Yuri Merkuryev,Richard Zobel,Eugene Kerckhoffs 2005 ISBN 1-84233-112-4.

[8] Laleh Haerian Ardekani, Tiru S. Arthanari, Matthias Ehrgott, "Performance of the Branch and Bound Algorithm on the Multistage Insertion Formulation of the Traveling Salesman Problem", Proceedings of the 45th Annual Conference of the ORSNZ, November 2010, pp. 326-335.

[9] Paulo Henrique Siqueira, Maria Teresinha Arns Steiner, Sergio Scheer, "Anew approach to solve the traveling salesman problem" in Neurocomputing, vol, 70, 2007, pp. 1013-1021.

[10] PHAM Dinh Thanh, HUYNH Thi Thanh Binh, BUI Thu Lam "A Surveon Hybridizing Genetic Algorithm with Dynamic Programming forSolving the Traveling Salesman Problem" International Conference ofSoft Computing and Pattern Recognition (SoCPaR), 2013.

[11] Yongsheng Pan, Yong Xia* "Solving TSP by Dismantling Cross Paths" 2014 IEEE International Conference on Orange Technologies (ICOT).