

# An Effective Parallelism Topology in Ant Colony Optimization Algorithm for Medical Image Edge Detection with Critical Path Methodology (PACO-CPM)

<http://dx.doi.org/10.3991/ijes.v3i4.5139>

Chetan S<sup>1,2</sup>, H S Sheshadri<sup>3</sup> and V Loksha<sup>4</sup>

<sup>1</sup> Jain University, Bengaluru, India

<sup>2</sup>Dr.Ambedkar Institute of Technology, Bengaluru, India

<sup>3</sup>PES College of Engineering, Mandya, Karnataka, India

<sup>4</sup>Vijayanagar Sri Krishnadevaraya University, Bellary, Karnataka, India

**Abstract**—In the digital world of medical transcription involving various dimensions of processes, detecting the edge of a standard medical image for clinical research/diagnosis, telemedicine and other applicative purposes requires various efficient and effective methodologies to address the needs of the processes. Among these various meta-heuristics, as the size of the problem tends to increase along with time, the processes and their elemental techniques, proven to have been providing viable solutions appeals for reserve management and lesser computation times, with the efficiency of such algorithms and algorithmic operations to be enhanced at suitable levels of abstraction.

In this paper we propose an effective topological algorithm, which inhibits the characteristic features of high performance parallel enumeration in such heterogeneous computation environments. The proposed scheduler in the defined topological algorithm takes into consideration the metrics generated by As Built Critical Path (ABCP) - A hybrid methodological process. These metrics are re-initialized and processed to address the management of resources and the realization of search space. We also propose a methodology for shared memory access by the ants to perform parallel computation and as well implement the optimization factor in detecting the edge. An in-depth analysis with respect to the Speedup factor and the Execution time metrics are analyzed for various scenarios under consideration. The differentiations are evaluated and plotted for further futuristic analysis.

**Index Terms**—As Built Critical Path(ABCP), Ant Colony Optimization(ACO), Parallel Computation, Combinatorial Optimization, Speed-up, Efficiency.

## I. INTRODUCTION

The parallelization algorithms and their solutions, has broadened the exquisiteness of the parallel architectures involving a broad spectrum of parallel based algorithms supporting ACO techniques. This parallelization methodology is population-based meta-heuristic solution dealing with the mutual exchange of the information pertaining to an image of choice. This deals with large computational times, decrementing the quality of the solution rendered with a sublime run-time behavior. Such combinatorial

optimization problematic scenarios could be successfully addressed with satisfactory results by Ant Colony Optimization algorithm. The major hindrance in this methodology was that the algorithm deviated with the speedup metric of the algorithm with respect to the processor as the complexity of the problem increased. Though ACO by its natural behavior has the naive natural parallelism, suitable for its implementation on large parallel computation machines, was unsuccessful in converging to a satisfying solution. A series of associated developments and proposals justifying the natural behavior were introduced by Piriya Kumar[15], with the introduction of an asynchronous parallel Max-Min ACO strategy involving local search policies. With this many more advanced and informative approaches were introduced and proposed by M. Randall et.al[16], D. Merkle et.al[17] and M. Dorigo et.al[18] with the involvement of the suitable parallel strategies and their successful implementations on the reconfigurable hardware architectures. The efficiency and usefulness of the algorithm depends on its applicative environment and at large in its sequential form. Its ability/capability to resolve most of the combinatorial logic problems with an appreciated time in its execution is to be considered. In most of the situations, parallel computing topologies have centred itself as an instrument with proportions favoring it in domains of effective population-based methodologies, with the utmost efficiency.

In this paper we propose an effective and efficient algorithm, an extensible process involving parallel computing topology in detecting the edge of a medical image of DICOM standard, an extension to the proposed, As Built Critical Path (ABCP) - a hybrid methodology. The processes are parallelized with the shared memory master-slave concept and elaborated with the evolution of the algorithm with an increase in the frequency of the computations, addressing the key aspects of parallelization performance metrics such as;

- i. reduction in the execution time
- ii. easily address an increase in the size of the problem (in this case it is an image)
- iii. increase the efficiency
- iv. speedup metric evaluation

The heterogeneous computation algorithm such as Ant Colony Optimization (ACO), requires efficient scheduling of the underlying tasks in the algorithm. This helps in achieving a high performance of the algorithm. These underlying tasks are shown to be "Non-deterministic Polynomial" NP-complete with NP-hard requisites in various different requiring scenarios. One such problem was addressed using a novel framework in scheduling the tasks based on the meta-heuristic process of ACO. This was successfully demonstrated in MATLAB, which produced significant and effective schedules for a random set of tasks. This was proposed by Jun Mao[3]. On similar terms, Vetri Selvan et.al[4] proposed an inherent parallel implementation of this meta-heuristics, which was exploited to be implemented effectively and efficiently on a multi-core processor platform to successfully demonstrate the time taken in producing these effective schedules for the same set of random set of tasks with their graphs. Fang Liu[5], proposed an inference to the property of torpid and accomplishment of the concept of evolution in popular heuristic ACO algorithm, a dual population ACO involving parallelism (DPPACO). This algorithm was successfully tested and applied to the preeminent travelling salesman problem. The DPPACO algorithm basically divided the entire ant population into soldier and worker ant population which was characterized with evolution, parallelism and the timely exchange of the information among these population was maintained. This algorithm basically enlarges the searching ranges, thereby improving the performance of the convergence overcoming the imbalance of the pheromone. Chetan .S et.al[2] proposed a similar concept with the injection of the ants onto a medical image, with the ant population being segregated as real ants and virtual ants, in order to improvise the local convergence and highlight the critical parameters based on the Critical Path Methodology (CPM), in defining the edge within a standard medical image. With the ACO meta-heuristic process being one of the most effective and efficient combinatorial optimization logics, numerous traditional CPU-based and parallelization techniques in ACO have been proposed. Among these, the various approaches are classified broadly into two categories;

- Parallel Ants - defines an approach proposal initiated by Bullheimer et.al[6], wherein the ants work on the construction of their path all along on a certain number of elements in parallel.
- Multiple Ant Colonies - is an approach based on the distributed architecture of the memory and a message-passing architecture. The main aim of the approach introduced by Stützle[7].

The most important approach that mostly follows the inadvertent parallel and multiple ant colonial approaches, were specifically and significantly dedicated by the reactionary hardware-adapted parallel approaches. This was proposed by Scheuermann et.al[13], with the designs being implemented with parallel ACO algorithms or Field Programmable Gate Arrays (FPGA). This was later developed and a new solution to the adhering problem in orienting the solutions for the parallelisation was proposed Catala et.al[14]. Apart from these phenomena, a framework was justified and proposed by Talbi et.al[8], based on the principle of communication between the ants of a colony, in solving the various combinatorial and optimization problems. Their work initialized the Ant Colony Op-

timization algorithm used to solve a class of NP-hard combinatorial optimization problem such as Quadrature Assignment Problem (QAP), having a vibrant applicative features such as image synthesis, data analysis, etc.

## II. PARALLEL COMPUTATION TOPOLOGY

Majority of the optimization and combinatorial problems are efficiently addressed by parallel topology in ACO, which have been developed and effectively iterated over the due course. This topology mainly considers the master-slave paradigm with synchronous/asynchronous procedural aspects. The paradigm basically implements and monitors the memory architecture which can either be a centralized architecture with the global acquisition of the data captured during the run-time procedure and a hub for communication among the slaves with the master. The slave initiates procedures and handles various multiple necessary and required, search and update processes. While the master upon acquisition of series of pheromone matrix values and the best possible work-around solution for the combinatorial problem, manages and produces with first-rate viable solution from the algorithm. The task/s exercised by the master in this topology includes updating of pheromone matrix upon end of each iteration to the slave, which handles the process of transition matrix construction based on the specified criteria. The slave henceforth takes a leap and process the received pheromone matrix and constructs a best possible solution in finding the edge of an image. It also finds the local solution and transmits the same to master for decision making. With the phase of heterogeneity, the best possible solution from all the  $n$  saves are acquired at suitable time intervals and are synthesized. A suitable solution matrix is constructed and updated simultaneously among all  $n$  slave units.

Such algorithms consume more computation time when introduced as a parallel version. A time bound iteration has to be made in-order to obtain certain specific effective computation times with the production of best possible solutions in time. The *sequential* behaviour of ants, assumed to involve a high level of natural parallelism with behaviour of a single ant independent of the behaviour of other ants of the colony. This natural behaviour of ants was inspired and was successfully proposed by Bullheimer et.al[6]. These parallel topologies can be effectively and efficiently implemented. They can be evaluated by using certain rational techniques, which involves simulation of the behaviour of ants or with the implementation on varying real-time hardware architectures. In these analytical models, an algorithm or a program, an abstract detailed with the characteristics of the parallel program, is analysed with respect to distinct environmental scenarios. This requires various initial assumptions, such as the start-up time and execution time, to be made effectively so as to process the problem with an appropriate term and completely synthesize the discrete-event simulation. Figure 1 shows the architecture of a time synchronous analytical model involving parallel strategies.

The execution of such a model outputs a file which records the execution time along with several other metrics related to the performance measure, such as

1. Speedup,  $S(N)$
2. Efficiency,  $E(N)$

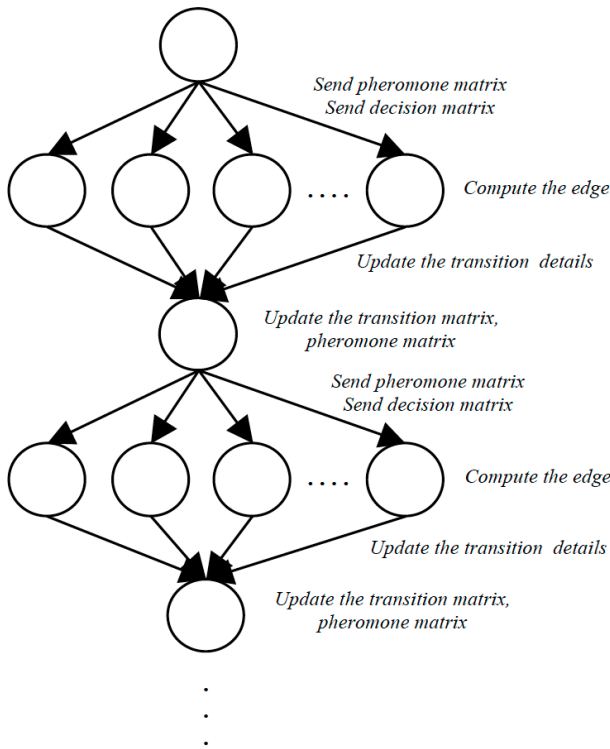


Figure 1. Master-Slave Asynchronous Model for Parallel Topology in PACO-CPM Analytical Model of the Parallel Computation Topology

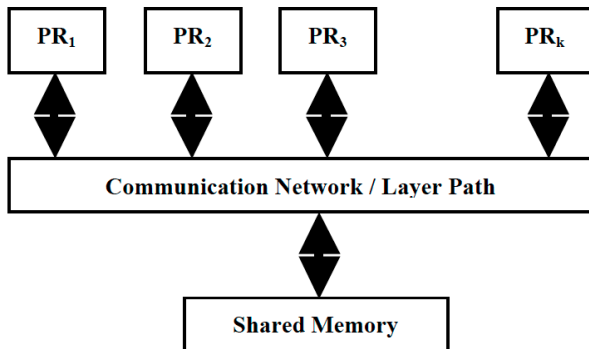


Figure 2. Shared Memory real implementation hardware architecture involving Parallel Computation Topology

With these two efficient approaches, parallel topologies involving the Ant Colony Optimization techniques in detecting the edge of a medical standard DICOM image are expected to gain momentum with execution time stamps as well speedup the process of their parallel logics execution. Such a competent methodology is proposed in this paper.

The main aspect in the combinatorial optimization problems is that the algorithm endeavours to detect discrete values that tends to optimize the solution with the best possible scenario. These combinatorial problems are easy to understand, while very critical with their solvability. The metric of performance, such as speedup of the algorithm, running parallel involving multiple processor scenarios will be limited by the time actually required by a sector of the algorithm or a fraction of the program/algorithm being executed. Under these conditions,  $n \in \mathbb{N}$ ,  $n$  is the number of threads/process of execution in an algorithm.

For a process to be parallelized, we consider the fact that a fraction of the algorithm needs to be strictly serial, according to Amdahl's Argument.

$$B \in [0,1]$$

The time,  $T(n)$ , is the actual execution time taken by the algorithm to execute  $n$  threads/processes, corresponding to the equation,

$$T(n)=T(1)(B+1/n(1-B))$$

As per the Amdahl's Argument, a model is considered to provide a speedup value. Also there exists a relationship between the parallelized process implementations and its serial implementations, assuming that the size of the problem remains the same throughout the execution of the algorithm. Under such circumstances speedup,  $S(n)$  can be obtained by executing the algorithm on a system capable of running the threads parallel during its execution, yielding

$$S(n) = T(1)/T(n) = T(1)/T(1)(B+1/n(1-B)) = 1/(B+1/n(1-B))$$

### III. METHODOLOGY

The initial proposal was proposed by Colorni et.al[20] and Dorigo et.al. In these, the communication happen between each other by the concentration of the *pheromone* deposition done by them along the path of their traversal. The natural behaviour of such population-based agents like ants is, the deposition of the *pheromone*, with a major concentration in a path of shortest distance between source to destination and the following members of the colony tends to follow the same path. This is reinforced to impact a preferred path for further processes. This concept was adopted by Dorigo and Gambardella, Dorigo and Di Caro, in finding the edge of an image and to address the combinatorial problem such as the travelling salesman problem, with the choice of suitable values as parameters and their details into an algorithm.

These agents also exhibit the natural behaviour of parallelism, adhering to the actual logic, improvising the execution form factors of the algorithm. The main aspect of this behaviour is that the solution can be obtained at a faster rate and the size of the problem can also be increased. Figure 3 shows the master-slave configuration which has got the synchronous model characteristics.

In this particular configuration, the master initiates the entire process of parallelization and manages the process until the best possible quality of the result is obtained. The master initialize the pheromone matrix  $\tau_{ij}$ , at the beginning of the algorithm run-time. This initialization factor is spawn over to  $n$  processors with the As-Built Critical Path (ABCP) methodology. Each processor handles the process with the acquisition of the broadcasted information by the master. The master system or the processor unit is provided with a global shared memory architecture, which acts as a database with the updated pheromone matrix values, constructed solutions at the end of each iterations and the final resultant matrix with the best possible solutions.

The initial point is that, the pheromone matrix  $\tau_{ij}$ , is broadcasted to all processors. The processor runs the algorithm parallel for generation and evaluation of solutions. The algorithm runs completely on each processor unit and

their probabilistic solutions are generated and broadcasted to the master unit. This unit evaluates all the solutions obtained, process and produce an updated pheromone matrix that will be initialized and stored in the global-shared memory unit. It can also be seen that in the Analytical message passing model the frequency of the hits that will be taken by the master unit in updating the pheromone matrix after each iteration, is reduced. The message passing model also requires suitable memory spaces allocated at different hierarchies in-order to accommodate the resulting solutions at different stages.

As the communication overhead is put aside and the implementation of parallelization strategies are involved, the scenario as explained Figure 4, shows the actual procedure involved in individual processors. As explained and proposed in the As-Built Critical Path (ABCP) methodology, both the  $k^{th}$  real ants and the  $m^{th}$  virtual ants are initialized into the processor. here we also assume that at least a single ant is inserted into each processor unit which contains both the real and virtual ants.

#### A. Updating the $\tau_{ij}$ Matrix

The pheromone matrix has to be updated to be made available for the next consecutive stages of the parallelism operations. Hence at the end of each stage of iteration from among the cluster of parallel ABCP ACO processors, it has to be broadcasted to the master unit and the master has to update the matrix suitably. In the work proposed in this paper the algorithm will update the pheromone matrix concurrently and will never wait for all the processors to complete the iteration cycle. Thus there is no need for the synchronization to be included in the model and this can be avoided in the algorithm.

The overall procedure of parallelization is also impacted by the fact that the chosen number of ants is a metric for the ACO. This constraint will be limited by the number of processor units available for the synthesis of the algorithm. This majorly is categorized as the load balancing factor. In this proposed work we have assumed to have the number of ants not to exceed more than 2400 which 100 times the number of processor units available for the synthesis of the algorithm in the parallel topology.

The ABCP methodology involves various critical parameters which will be considered through which the parallel topology will be subjected to the process and the process acknowledges the updating of the pheromone matrix, transition probability matrix and the detection of an edge in an image. In these circumstances, the image we have considered for the evaluation of the parallel topology is a DICOM format medical images of suitable resolutions 256X256 and 512X512, grayscale images.

#### IV. ANALYTICAL RESULTS

The effectiveness and the implementation strategy of the proposed topology, was successfully experimented with the suitable development of a parallel algorithm. The performance metrics which includes speedup, execution time and efficiency related with the time, memory and the logical implementation space for an algorithm are evaluated and processed to differentiate between the applicative features and protocols to be followed in their implementation and application aspects. These analytical results are liable to prove that the proposed parallel topology involving the exclusive hybrid methodology such as As-Built

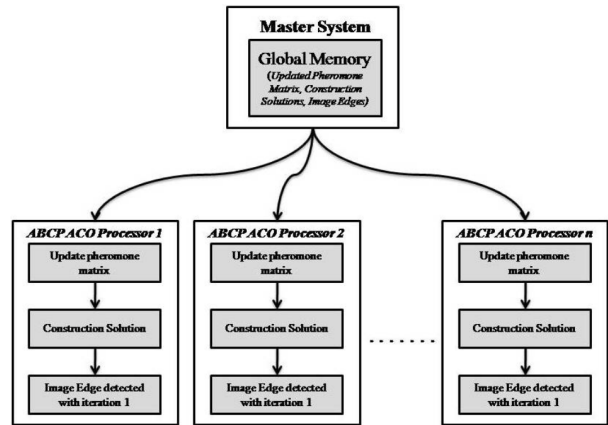


Figure 3. Master-Slave synchronous model for Parallel Topology in PACO-CPM

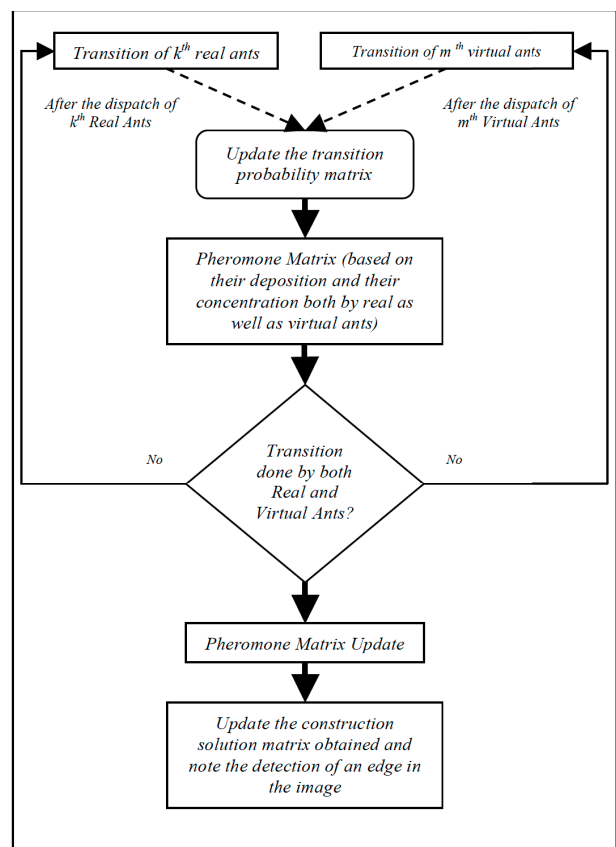


Figure 4. Proposed methodology workflow\*

\* The availability of at least one real and virtual ant can be disregarded in the case involving high number of communications being required.

critical Path (ABCP) [2] in their routine and the critical parameters to be considered along with, to prove their implementations are more reliable and efficient.

#### A. Speedup

In a parallel computation topological scenario time and memory yield major of the performance metrics. The metric that specifies the improvement in the performance, in terms of *Instructions Per Cycle (IPC)* or the execution time ( $T$ ), of an architectural topology defined for a computation machine. This metric is originated from a well known argument known as the *Amdahl's Argument*. In parallel computing, this argument specifies the conven-

tional improvement expected in the overall performance of the computation algorithm, wherein certain sectors of the algorithm are improvised, in multi-processor scenarios. The speedup parameter defines the throughput or the latency values for a given algorithm.

$$S = T_{old} / T_{new}$$

$$S = \frac{CPI_{old}}{CPI_{new}}$$

In the above expressions,

$S$  – the resultant Speedup

$T_{old}$  – the old execution time (without the improvement factor involved)

$T_{new}$  - the new execution time (with the improved factor/s derived from ABCP algorithm is/are included)

$CPI_{old}$  - the old performance metric (without the improvement factor involved)

$CPI_{new}$  - the new performance metric (with the improved factor/s derived from ABCP algorithm is/are included).

In the above equations, the values are substantially substituted for the variables mentioned as for the old execution time, the base algorithm is run on the 12 logical processors, with 6 cores and 24 processor units yielding an execution time of 365.96 seconds, for a 256X256 resolution DICOM standard medical image. Then later when the critical parameters are considered and are critically included into the algorithm and run with the same 24 processor units and 12 logical processors, produced an execution time of 97 seconds. Substituting the resulting magnitudes obtained in the above equations for Speedup metric, we obtain a magnitude value of 3.77. Hence we can conclude with the proclamation that, the new algorithm has provided an effective result, with 3.77x speedup over the actual algorithm without the ABCP critical parameters included in the parallel topological implementation of the algorithm.

Figure 5, Figure 6, Figure 7, Figure 8 show the relation between the performance metrics speedup and the efficiency obtained from the parallel implementation when processor units up to 24 are used to process a medical standard DICOM format grayscale image of resolutions 256X256 and 512X512. This also involves the assumption of two set parameters to be considered, as the number of ants involved and the computation cycles the parallel algorithm would run in providing the best possible solutions

### B. Execution Time Analysis

With the results obtained from the successful run of the algorithm based on the parallel topology, we can see from the Figure 9, Figure 10, Figure 11, Figure 12, that there appears to be a lot of deviation in the execution time when there is only single lone processor involved in the parallel process.

The synchronization factor involved in this proposed work has an impact over efficiency. This factor was addressed by inserting variations in the number of computation cycles involved in generating the pheromone matrix and the best possible solution for the considered resolution of the image. In spite of these factors, we can also note that there appears to be a degradation in the efficiency, with increase in the number of cycles as well as in the

number of processors. This also implies with the fact that there will be as significant drop in the efficiency with the increase in the number of ants beyond the permissible level of 2400.

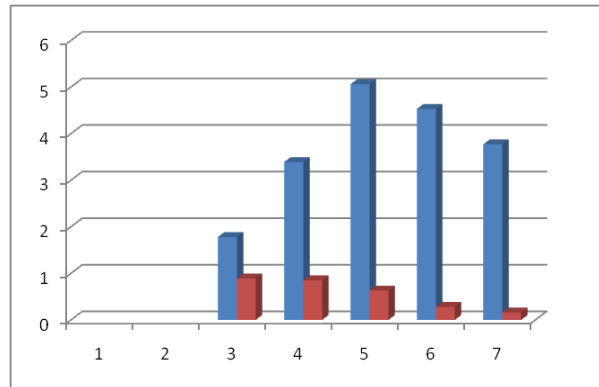


Figure 5. Spedup vs Efficiency for 256\*256, 100 cycles, 1500 ants

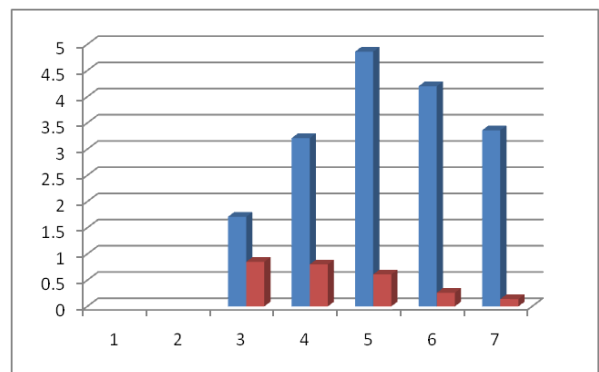


Figure 6. Speedup vs Efficiency for 256\*256, 200 cycles, 1500 ants

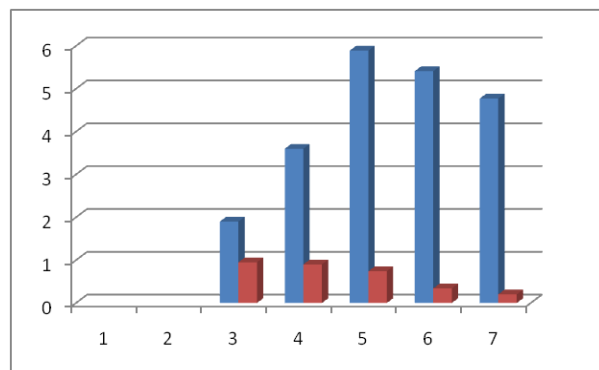


Figure 7. Speedup vs Efficiency for 512\*512, 100 cycles, 1500 ants

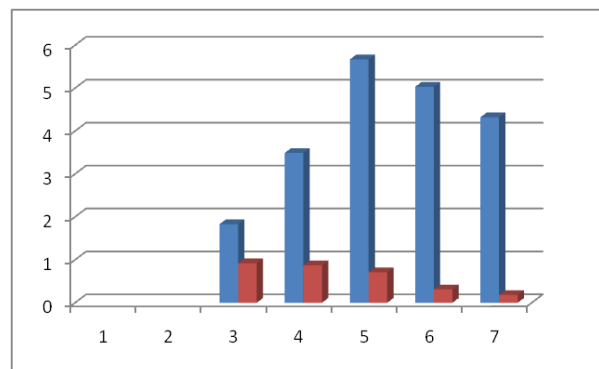


Figure 8. Speedup vs Efficiency for 512\*512, 200 cycles, 1500 ants

TABLE I.  
RESULTS OF PARALLEL EXECUTION WITH 256X256 RESOLUTION IMAGE,  
100 COMPUTATION CYCLES AND 1500 ANTS INITIALIZED

Number of Processors	Execution Time (sec)	Speedup	Efficiency
1	568		
2	298	1.78	0.88
4	172	3.39	0.85
8	109	5.06	0.63
16	118	4.53	0.28
24	97	3.77	0.16

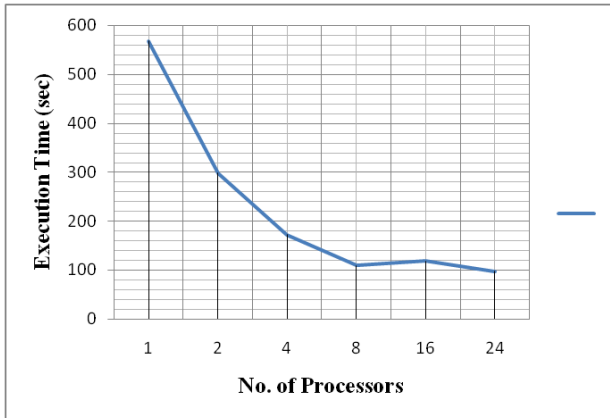


Figure 9. Execution time with 256\*256 resolution DICOM format medical image, 100 computation cycles, 1500 ants

TABLE III.  
RESULTS OF PARALLEL EXECUTION WITH 512X512 RESOLUTION IMAGE,  
100 COMPUTATION CYCLES AND 1500 ANTS INITIALIZED

Number of Processors	Execution Time (sec)	Speedup	Efficiency
1	1102		
2	558	1.9	0.95
4	300	3.6	0.90
8	181	5.89	0.74
16	196	5.41	0.34
24	134	4.77	0.20

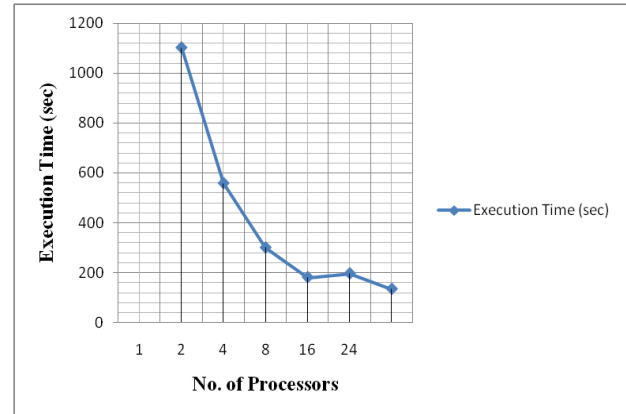


Figure 11. Execution time with 512\*512 resolution DICOM format medical image, 100 computation cycles, 1500 ants

TABLE II.  
RESULTS OF PARALLEL EXECUTION WITH 256X256 RESOLUTION IMAGE,  
200 COMPUTATION CYCLES AND 1500 ANTS INITIALIZED

Number of Processors	Execution Time (sec)	Speedup	Efficiency
1	1127		
2	628	1.72	0.85
4	343	3.21	0.80
8	225	4.86	0.61
16	259	4.2	0.26
24	217	3.36	0.14

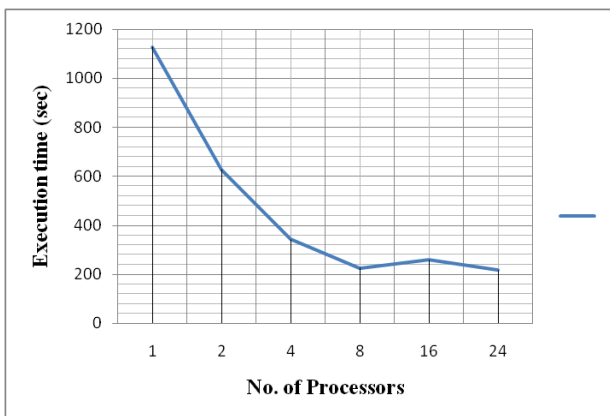


Figure 10. Execution time with 256\*256 resolution DICOM format medical image, 200 computation cycles, 1500 ants

TABLE IV.  
RESULTS OF PARALLEL EXECUTION WITH 512X512 RESOLUTION IMAGE,  
200 COMPUTATION CYCLES AND 1500 ANTS INITIALIZED

Number of Processors	Execution Time (sec)	Speedup	Efficiency
1	2172		
2	1146	1.83	0.9
4	605	3.49	0.82
8	375	5.67	0.69
16	421	5.03	0.31
24	396	4.32	0.34

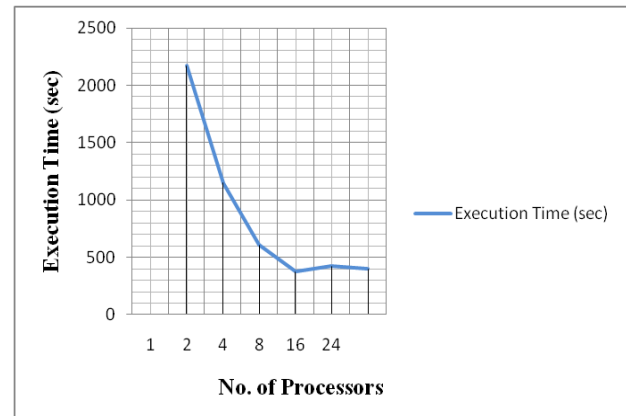


Figure 12. Execution time with 512\*512 resolution DICOM format medical image, 200 computation cycles, 1500 ants

### C. Efficiency

This metric gives a measure of the performance related to the speedup metric closely. This metric is actually defined as the ratio of the performance metrics speedup and the number of processors. Speedup is factor which is linearly related to the efficiency. Speedup is also closely related to the run-time, as we can say it is dependent on run-time, as it favours processors with slow run-time and code being run on those processors, efficiency also.

Different efficiency formulations were proposed by Carmona et.al [19], wherein the efficiency metric was scribed as ratio of the accomplished work by the parallel algorithm and the time of the work by the same algorithm for a process. For this, they suitably considered the work accomplished by the algorithm as the process serviced in providing the best possible scenarios, with the speedup of the individual processor and the number of processors being expressed as  $P$ , then,

$$we = (\text{parallel time}) * S * P$$

$$wa = (\text{best possible time by ABCP}) * S$$

$$wa/we = ((\text{best possible time by ABCP}) / (P * (\text{parallel time})))$$

### V. CONCLUSION

In this paper we have proposed an effective parallel topology for As-Built Critical Path (ABCP). This involves parallel implementation of ABCP algorithm inclusive of the famous meta-heuristic ACO. This algorithm was modelled to address the performance metrics such as execution time, speedup and provide an efficient topology to enhance edge detected from a medical image of DICOM format. The implementation of the algorithm was successfully done resulting in the development of an effective parallel topology with shared memory architecture, thereby increasing the probability of the search paths and also improvising the convergence to the best possible solution. An increase in execution time for the same magnitude of the combinatorial problem was also observed. The algorithm was tested for the effect of synchronicity, parallelism and concurrency on the performance of the original shared memory ACO parallel algorithm. A constitutional tolerance was also exhibited in terms of the process order and arriving at the affirmative probabilistic solutions.

In our work, we have tested for two images of resolution 256X256 and 512X512 grayscale DICOM format medical images. The number of ants independent of the number of processor units were assumed in processing the algorithm for obtaining an effective solution probability and also the number of computation cycles rendered to the processor unit was also scribed for an effective approach. This algorithm was also successful in show casing its prosperities against the serial run-time of the algorithm/program.

### ACKNOWLEDGMENT

This research paper is made possible through the help and support from everyone, including: parents, teachers, family, friends, and in essence, all sentient beings.

I want to express my sincere gratitude to my institution Dr. Ambedkar Institute of Technology for giving me an opportunity and made resources available to fulfill my research work. Especially, I want to thank Dr. Murali. R,

Professor, Dr. Ambedkar Institute of Technology for his encouragement, invaluable advice in bringing up this research paper.

I sincerely thank my parents, family, and friends, who provide the advice and moral support. The product of this research paper would not be possible without all of them.

### REFERENCES

- [1] S. Chetan, H.S. Sheshadri, V. Loksha, "A Descriptive Study on Improvisation of Parallel Ant Colony optimization with the Implementation of CPM for Genetic Algorithm with Evolution and Fast tracking", Proceedings of Second International Conference on "Emerging Research in Computing, information, Communication and Applications", ERCICA 2014, Elsevier Publications, pp. 303 - 307, ISBN: 9789351072621.
- [2] S. Chetan, H.S. Sheshadri, V. Loksha, "A Hybrid Critical path methodology - ABCP (As Built Critical Path); its Implementation and Analysis with ACO for Medical Image Edge Detection", International Journal of Computer Science, Information technology and Control Engineering (IJCSITCE), Vol.2, No.1/2, April 2015, pp.27 - 42, ISSN: 2394-7527.
- [3] Jun Mao, "Task Scheduling of Parallel Programming Systems using Ant Colony Optimization", Proceedings of the third International Symposium on Computer Science and Computational Technology (ISCST '10), August 2010, pp. 179-182, ISBN: 978-952-5726-10-7.
- [4] T. Vetri Selvan, P. chitra, Dr. P. Venkatesh, "Parallel Implementation of Task Scheduling using Ant Colony optimization", International Journal of Recent Trends in Engineering, Vol. 1, No. 1, May 2009, ACEEE.
- [5] Fang Liu, "A Dual Population Parallel Ant Colony Optimization algorithm for solving the Travelling Salesman Problem", Journal of Convergence Information Technology (JCIT), Vol. 7, No. 5, March 2012.
- [6] Bullheimer Bernd, Kotsis Gabriele, Strauss Christine, "Parallelization Strategies for the Ant System", in "High Performance Algorithms and Software in Nonlinear Optimization", Vol. 24, *Applied Optimization*, pp. 87-100, Kulwer, Dordrecht, 1997.
- [7] Max Manfrin, Mauro Birattari, Thomas Stützle, Marco Dorigo, "Parallel Ant Colony Optimization for the Traveling Salesman Problem", Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V), Vol. 1498, pp. 722-731, Springer-Verlag, New York, 1998.
- [8] El-ghazali Talbi, Olivier Roux, Cyril Fonlupt, Denis Robillard, "Parallel Ant Colonies for Combinatorial Optimization Problems", *Future Generation Computer Systems*, 17(4): 441-449, 2002.
- [9] M. Dorigo, Thomas Stützle, "Ant Colony Optimization", MIT Press/Bradford Books, 2004.
- [10] D. Chu, M. Till, A. Zomaya, "Parallel Ant Colony Optimization for 3D Protein Structure Prediction using the HP Lattice Model", Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), ISBN: 1530-2075/05.
- [11] Ling Chen, Chunfang Zhang, "Adaptive Parallel Ant Colony Algorithm", First International Conference on Advances in Natural Computation (ICNC'05), Springer 2005, LNCS 3611, Springer-Verlag Berlin Heidelberg 2005, pp. 1239-1249, ISBN: 3540-28320-X. [http://dx.doi.org/10.1007/11539117\\_165](http://dx.doi.org/10.1007/11539117_165)
- [12] Pierre Delisle, Michaël Krajecki, Marc Gravel, Caroline Gagné, "Parallel Implementation of an Ant colony Optimization Metaheuristic with OpenMP", *Journal of Parallel and Distributed Computing*, Vol. 73, Issue. 1, January 2013, pp.52-61.
- [13] Bernd Scheuermann, Stefan Janson, Martin Middendorf, "Hardware-oriented Ant Colony Optimization", *Journal of Systems Architecture* (2006),. <http://dx.doi.org/10.1016/j.sysarc.2006.12.002>
- [14] Alejandro catala, Javier Jaen, Jose A. Mocholi, "Startegies for Accelerating Ant Colony Optimizaion Algorithms on Graphical Processing Units", IEEE Congress on Evolutionary Computation (CEC'07), ISBN: 978-1-4244-1339-3, pp.492-500.
- [15] Douglas Antony Louis Piriyaikumar, P.L (2002), "A new approach to exploiting Parallelism in Ant Colony Optimization", Proceedings of 2002 International Symposium on Micromechatronics and

- Human Science, pp. 237-243. <http://dx.doi.org/10.1109/MHS.2002.1058041>
- [16] Marcus Randall, A Lewis, "A Parallel Implementation of Ant Colony optimization", Journal of Parallel and Distributed Computing, 62(9), Elsevier Science, 2002, ISBN: 0743-7315/02, pp. 1421-1432.
- [17] S Janson, D Merkle, M Middendorf, Alba E, "Parallel Ant Colony Algorithms", Parallel Metaheuristics, Wiley Book Series on Parallel and Distributed Computing, pp. 171-201, Wiley, 2005.
- [18] M Dorigo, L M Gambardella, m Birattari, A Marinoli, R Poli, T Stützle, "Parallel Ant Colony Optimization for Travelling Salesman Problem", Ant Colony Optimization and Swarm intelligence, 5th International Workshop, ANTS 2006, Vol. 4150, Lecture notes in Computer Sciences, Sep 2006.
- [19] E A Carmona, M D Rice, "A Model of Parallel Performance", Technical Report AFWL-TR-89-01, Air Force Weapons Laboratory, 1989.
- [20] A Colomi, M Dorigo, V Maniezzo, "Genetic Algorithms and Highly Constrained Problems: The Time-Table Case", Parallel Problem Solving from nature, Springer Berlin Heidelberg, 1991/1/1, pp. 55-59, 1991.
- [21] Vipin Kumar, Anshul Gupta, "Analyzing Scalability of Parallel Algorithms and Architectures", Journal of Parallel and Distributed Computing, 1984, pp. 1-21.

## AUTHORS

**Chetan S** is a PhD Scholar majoring in Electronics Engineering at the Jain University, Bengaluru. He received the B.E. Degree in Electronics and Communication Engineering from Visvesvaraya Technological University in 2005 and post graduation in VLSI Design and Embedded System from Visvesvaraya Technological University in 2009. His current research interests include Image Processing, medical image segmentation and analysis and digital communication. He is a life member of ISTE, SIE, IAENG, IASCIT and has participated in various national

and international conferences and seminars in India and abroad.

**Dr. H S Sheshadri** obtained his B.E. from University of Mysore during 1979 in E & C Engg, M.E (Applied Electronics) from Bharathiar University, Coimbatore during 1989 and PhD from Anna University Madras during 2008. He is serving presently as a Professor and Dean (Research) in PES College of Engineering, Mandya. His field of interest is Medical Image processing and embedded systems. He has published 12 papers in national journals and 21 papers in International journals. Also guiding 6 research candidates for PhD programme under university of Mysore, Two candidates under VTU, Belgaum. He is a life member of IE (India), IETE, ISTE, SSI, and has participated in various national and international conferences and seminars in India and abroad.

**Dr. V Lokesh** obtained his PhD from Mysore University, Mysore and is currently Special Officer and Deputy Registrar in Vijayanagara Sri Krishnadevaraya University, Bellary. He is presently Chairman of Department of Computer Science and Mathematics. He was invited in Turkey, France, South Korea, and Iran for International conferences. He is a reviewer of American Mathematical reviews, USA and several Journal of international repute. He delivered several series of lectures under the V.T. U EDUSAT - LIVE telecast lecture series programme. He has published more than 100 research papers in the journals of international repute. Under his Guidance 8 PhD and 28 MPhil degrees are awarded and working as editorial member of several international and National Journals.

Submitted 05 November 2015. Published as resubmitted by the authors 30 November 2015.