TRANSFORMATION MODELS OF DIRECTED CYCLIC GRAPH ONTO ACYCLIC GRAPH AND MAPPING FOR TASK ASSIGNMENT PROBLEM

WAN NOR MUNIRAH ARIFFIN

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy

Faculty of Science
Universiti Teknologi Malaysia

APRIL 2021

ACKNOWLEDGEMENT

First, I would like to say my deepest gratitude to Allah the Almighty, the Merciful for guiding me all the way through completing this thesis with the title "Transformation Models of Directed Cyclic Graph onto Acyclic Graph and Mapping for Task Assignment Problem." Without His guidance and blessings, it would not be possible for me to complete the thesis due time.

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my best lecturer I ever met, Professor Dr. Shaharuddin Salleh, for encouragement, guidance, critics and friendship. I am also very thankful to Dr. Shazirawati Mohd Puzi and Professor Dr. Fadhilah Yusof for their guidance, advices, and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

Thank you so much to my husband, Dr. Emy Aizat Azimi, my charming prince Emir Abqari and my beautiful princess, Eiklil Aghnia. No words could well express my gratefulness of your patience and emotional supports, as well as your du'a have been providing me since the first year of my research journey in PhD. To my siblings Power Rangers Mu'izzuddin, Musliha, Mustika, and Musliyana, thank you for always cheering me up when I was down.

Last but not least it leaves me to thank two most important people that I love so much, my mother Shariffah Mas binti Syed Omar Al-Idrus and my father Ariffin bin Ibrahim. Without your love, support, and inspiration all of this would never been possible.

ABSTRACT

Task assignment is one of the most challenging problems in distributed computing environment. Several approaches and techniques of optimal task assignment have been proposed by various researchers, ranging from graph partitioning-based tools to heuristic graph matching. A good task assignment algorithm and mapping strategy ensure completion time minimization. Previous work put efforts on solving a directed acyclic task graph. However, if a cycle exists, it is difficult and sometimes impossible to complete the task according to the constraints. Thus, it is desirable to develop a method that can eliminate the cycle to obtain a directed acyclic graph for a task assignment problem. In this thesis, the techniques of transforming a directed cyclic graph formed by a plurality of nodes into a directed acyclic graph are studied. Three models have been developed, starting with graph matching which transforms unweighted directed graph with cycle into acyclic graph. The second model is an extension from the first model, but with weightage on the edges. The third model discussed edge reversing technique to solve the weighted directed cyclic graph. This technique comprises of three main stages which are cycle identification, decomposition and formation of acyclic graph, and task assignment. Preferably, the length of the edge is the computational time. The optimal solution gives the minimum time required to complete the proposed task. The proposed algorithms were coded and simulations are run for each model using respective program developed by JavaScript programming language with D3 library. Based on the results, the proposed models give better optimality index by 20.83% compared to previous works. This research provides fundamental solution and better understanding to avoid cycle on assignment problems thus helping organizations or companies to increase the efficiency in planning as well as reducing the routing cost. This will give a great impact and benefit the manufacturing industry as well as transportation business where model and algorithms can be adjusted in their software and applications.

ABSTRAK

Penetapan tugasan adalah salah satu masalah yang paling mencabar dalam persekitaran pengkomputeran teragih. Beberapa pendekatan dan teknik penetapan tugasan yang optimum telah dicadangkan oleh pelbagai penyelidik daripada kaedah pembahagian graf hingga padanan graf heuristik. Algoritma penetapan tugasan dan strategi pemetaan yang baik pasti meminimumkan masa penyelesaian. Kajian terdahulu menyelesaikan graf tugas tidak berkitaran berarah. Walau bagaimanapun, jika kitaran wujud, adalah sukar dan kadang-kala mustahil untuk menyelesaikan tugas mengikut kekangan. Justeru, adalah wajar untuk mengkaji kaedah untuk menghapuskan kitaran untuk mendapatkan graf tidak berkitaran yang diarahkan untuk masalah penetapan tugas. Dalam tesis ini, teknik-teknik mengubah graf kitaran berarah yang terdiri daripada sebilangan nod kepada graf kitaran tidak berarah akan dikaji. Tiga model telah dibangunkan, bermula dengan padanan yang mengubah graf arah tanpa pemberat dengan kitaran menjadi graf tak berkitaran. Model kedua adalah lanjutan dari model pertama, tetapi dengan pemberat pautan. Model ketiga membincangkan teknik membalikkan pautan untuk menyelesaikan graf berkitaran terarah berwajaran. Teknik ini merangkumi tiga peringkat utama iaitu mengenal pasti kitaran, menguraikan dan membentuk graf tak berkitaran, dan penetapan tugas. Sebaiknya, panjang pautan adalah masa pengiraan. Penyelesaian yang optimum memberikan masa minimum yang diperlukan untuk menyelesaikan tugas yang dicadangkan. Algoritma yang dicadangkan telah dikod dan simulasi dijalankan untuk setiap model menggunakan pengaturcaraan JavaScript dengan perpustakaan D3. Berdasarkan keputusan, model yang dicadangkan memberikan indeks optimum yang lebih baik sebanyak 20.83% berbanding kajian sebelumnya. Penyelidikan ini memberikan penyelesaian asas dan pemahaman yang lebih baik untuk mengelakkan masalah penugasan sehingga membantu organisasi atau syarikat untuk meningkatkan kecekapan dalam merancang dan juga mengurangkan kos peralihan. Ini akan memberi kesan yang besar dan menguntungkan industri pembuatan serta perniagaan pengangkutan yang mana model dan algoritma dapat disesuaikan dalam perisian dan aplikasi mereka.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DAG	-	Directed Acyclic Graph
DCG	-	Directed Cyclic Graph
HGM	-	Heuristic Graph Matching

LIST OF SYMBOLS

$G(V', A')$	-	Co-comparability Graph
$G(V, A)$	-	Directed Acyclic Graph with node V and edge A
$G(V, E)$	-	Directed Cyclic Graph with node V and edge E
E	-	Edge
V	-	Node
P	-	Processor
w_{ij}	-	Weight

CHAPTER 1

INTRODUCTION

1.1 Overview

Graph theory is an important area of contemporary mathematics with many applications not only in other branches of mathematics, but also in scientific disciplines such as engineering, computer science, management sciences, operational research and the life sciences. In general, a graph is a connection of vertices (called nodes) and links (called edges) together with a rule on how the vertices are connected to one another with the edges. Graphs are excellent modeling tools. One can read all information from a graph and record the abstract definition from the graph in a unique way. Hence, the practical applications can be reduced into graphs.

To solve the problems of the applications, a graph which represents the problem can be transformed or reduced into a simpler form of graph rather than solving it directly. Directed graphs are employed for modelling various applications such as task assignment problem. Assignment problem refers to the analysis on how to assign *n* tasks onto *m* agents in the most optimal way. The problems associated with the assignment of tasks have long been recognised in the art. Assignment of jobs to workers, contract to contractors undergoing a bidding process, assigning nurses to duty

post, or time tabling for teachers in school and many more have become a growing concern to both management and sector leaders alike.

Several approaches of optimal task assignment have been proposed, ranging from graph partitioning-based tools to heuristic graph matching. In an attempt to solve the problem in the general case, a number of heuristics have been introduced. The task assignment problem is an open research area and still has room for improvement. Without accurate schedule information, a worker has no means of assessing and making realistic due date commitments for prospective customer orders. Consequently, workers are frequently reassigned or asked to work overtime to compensate for unanticipated bottlenecks. These disruptions are very costly.

Existing task assignment solution and scheduling techniques attempt to overcome these disruptions by mapping the task into a directed graph. However, most of the previous studies put attention on directed acyclic graph. Thus, this study will put effort on investigating the problem with directed cyclic task graph.

1.2 Problem Background

The classical problem of the task assignment of resources is a well-known issue, widely described in one of the branches of mathematics and computer science, namely in Graph Theory. The Assignment Problem (AP) is also known as the maximum weighted bipartite matching problem which is a special type of Linear

Programming Problem (LPP), in which the objective is to assign number of jobs to number of workers at a minimum cost (time).

Task assignment can be in the form of projects and processes performed as part of the workflow may utilize multiple human and physical resources to complete individual tasks. The management of the workflow can require coordination between each of the resources directed with completing individual tasks that are part of the process. Often, the resources assigned to complete tasks of the workflow may be located in geographically remote locations or otherwise unable to be present in the same location. Tasks may require resources to be coordinated remotely. Individual tasks of a workflow process may depend on the completion of a previous task before starting and completing a subsequent task. A delay in completing any previous tasks at any stage of the process may have a cascading effect which may in turn delay the completion of the subsequent tasks, therefore delaying the completion of the entire process.

Currently available computerized management tools may improperly assume that each of resources that may be assigned a task, as part of the workflow, are presently capable, ready and able to complete the task once it has been assigned and within the time frame allotted for completion. In reality however, resources requested for the completion of a process may often be conducting business away from their computer systems, or unable to fulfill the request. This may particularly true when the individuals being relied upon are management level individuals of an enterprise who have very busy schedules. Assigning tasks to a management level individual can be difficult when they are consistently away from their computing system. Often,

workflow management relies on the secretaries of the manager level individuals to schedule tasks for completion in a manner that fits into the schedules of the managers.

While relying on secretaries may assist in completing the tasks of the workflow processes, having to wait and schedule the performance of the tasks can cause delays resulting in an increased cycle time and incidentally limit the resources that may be available for completing subsequent tasks. Accordingly, a need exists in the art for a workflow management tool that automatically preschedules for completion each task of the workflow process and continuously optimizes the completion of the workflow in order to complete the process in the most efficient and reliable manner possible.

The mathematical formulation of the problem suggests that this is an integer programming problem and is highly degenerated. It can be stated as follows: given a bipartite graph made up of two partitions *V* and *U*, and a set of weighted edges *E* between the two partitions, the problem requires the selection of a subset of the edges with a maximum sum of weights such that each node $u_i \in V$ or $u_i \in U$ is connected to at most one edge. The problem may also be phrased as a minimization problem by considering, instead of edge weights w_{ij}, a set of non-negative edge costs, $c_{ij} = W - w_{ij}$, where W is at least as large as the maximum of all the edge weights. It can also be stated as: how to determine the best possible assignment of workers to jobs, such that the total ratings are maximized (Cimen, 2001). All the algorithms developed to find optimal solution of transportation problem are applicable to assignment problem. However, due to its highly degeneracy nature, a specially designed algorithm widely known as Hungarian method proposed by Bogomolnaia and Moulin (2001) is used for its solution.

Several approaches and techniques of optimal task assignment have been proposed by various researchers ranging from graph partitioning-based tools to heuristic graph matching. A good task assignment algorithm and mapping strategy ensure turnaround time minimisation. Thus, many researchers have attempted to produce good algorithm to solve the task assignment problem. Previous work put efforts on solving a directed acyclic task graph. Liao, Yin and Goa (2012) presented a work on designing routing schemes in the DPillar network. The idea of how servers in DPillar are addressed and connected is interesting. However, they only discussed on the arrangement of the nodes without solving the assignment problems. Gupta (2013) solved the weighted acyclic graph using matching technique and later mapped onto processors. Mohan and Gopalan (2013) extended the idea of Gupta (2013) and discussed the Shen Tsai's Algorithm on task assignment problem. They came out with a new 'Parallel Heuristic Graph Matching Algorithm (HGM)' which improvised their previous techniques written in the previous work. Both qualitative and quantitative, pertaining the algorithm which signifies how efficient the proposed algorithm is, compared to the existing ones.

The previous study highlighted on task assignment problem from the directed-acyclic graph and undirected graph. Numerous techniques have been produced from the previous work. What is lacking is that if a directed cycle exists, it would be impossible to complete the task according to the constraints. To address these challenges in industrial work, transportation costs can be a significant part of a company's overall logistics budget. With the increases in the price of fuel, company should provide good assignment of workers to the transports together with good routing to avoid repetitions of travelling to complete the given tasks. To address this issue, businesses have incorporated better strategies that will help identify transportation issues and develop the appropriate solutions. Therefore, it would be

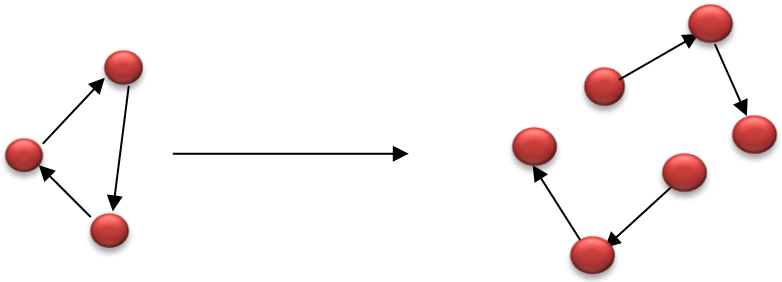
useful to find a solution if the cycle exists. The cycle in a task graph (job) should be eliminated, so that the job can be assigned to the right people and the task can be completed or done in appropriate time.

1.3 Problem Statement

Despite the fact that there are many studies on solving the task assignment problem ranging from graph partitioning to matching and heuristics, there is one thing need to be considered if cycles exists on the task graph. Based on the issue, it takes the attention on the cyclic task graph. The transformation problem can be stated as follows:

How can a given directed cyclic graph be transformed into a form of directed acyclic graph so as to optimally assigned onto n number of processors, P ?

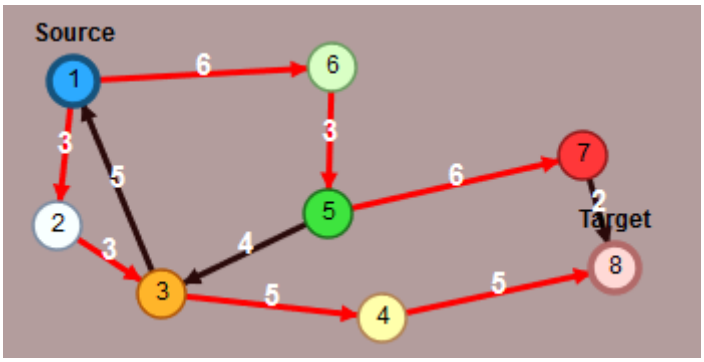
The problem can be illustrated as in Figure 1.1. The initial task graph consists of cycle. It is undesirable and impossible to complete the task according to the constraints if the cycle exists. Therefore, an effort should be done in order to eliminate the cycle to obtain a directed acyclic graph (DAG). Figure 1.1 shows the initial idea of transformation cyclic graph (a) into acyclic graph (b).



(a) (b)

Figure 1.1 The initial idea of transformation of cyclic graph onto acyclic.

Figure 1.2 shows the initial idea on the mapping of the task graph onto *n* number of processors. In this example, the number of processors is two, which is P_1 and P_2.



P_1	7	5				
P_2	8	3	1	2	4	6

Figure 1.2 Mapping of task graph onto the two processors.

In the mapping of the nodes (tasks) onto processors, the chosen nodes when reduced into the form of directed acyclic graph has significantly affected the results obtained in the schedule. Thus, the best formation of directed-acyclic graph which can lead to the optimal result with minimum computation time (distance on edge) is the core of this research. The proposed techniques with algorithm were designed and tested to various number of nodes in the task graph.

Thus, the following issues need to be addressed, as shown below:

i. Could the proposed technique be capable of optimizing optimal selection of cyclic nodes and edges to become acyclic, where the unweighted directed cyclic graph and weighted directed cyclic graph are taken into consideration as the proposed task graph?

ii. Would the mapping of tasks onto two and more than two number of processors gives better optimality index (efficiency) when hierarchy arrangement of nodes and reversing the cyclic edges are implemented?

iii. Could the proposed technique improve the task assignment problem when dealing with larger number of nodes and would it be able to give better approach on transforming the cyclic graph onto acyclic graph?

1.4 Research Objectives

As discussed in Section 1.2 and 1.3, there are three main issues (unweighted directed-cyclic graph and weighted directed-cyclic graph, the technique of breaking

the cycles as well as the mapping of task graph onto processors) encountered by the task assignment problem. These issues tend to restrict the efficiency of the algorithm and optimality of the results. Therefore, this study aims to develop and propose a better algorithm, which can increase the accuracy of mapping the task graph. The objectives of the study are listed as follows:

i. To develop an algorithm for solving the cyclic graph to become acyclic based on hierarchy of task (node) for unweighted graph and then mapped onto two processors.

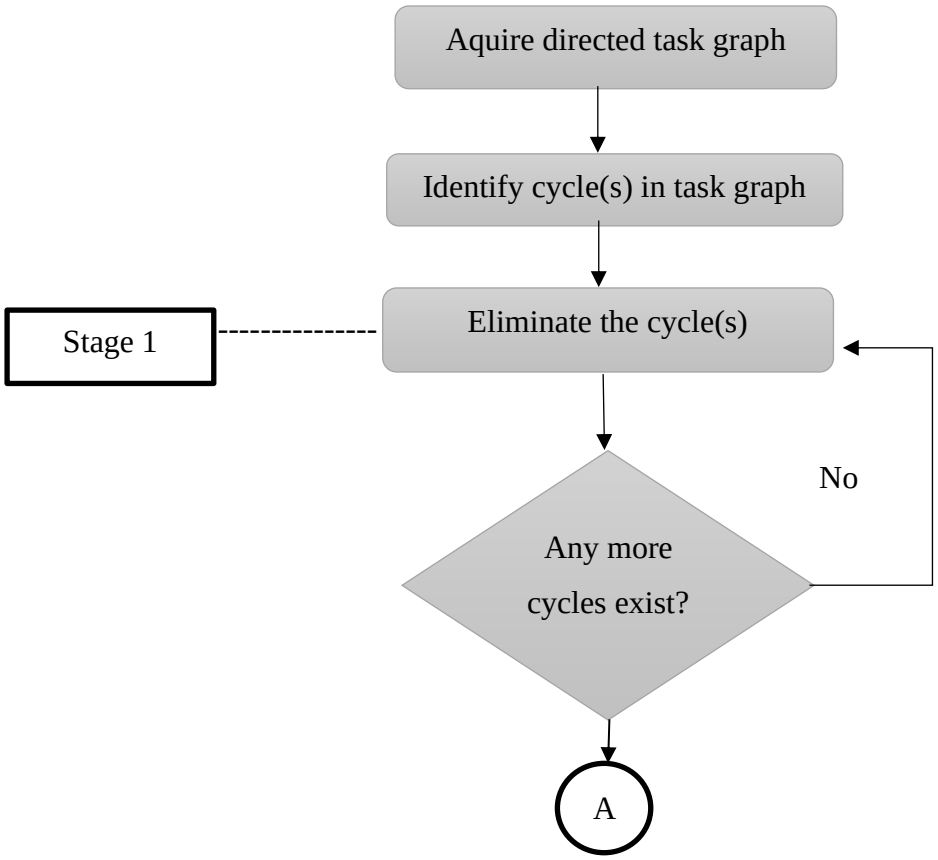
ii. To implement the algorithm from (i) for weighted directed graph in which it can optimally reduce the cyclic graph into an acyclic graph and then mapped onto n number of processors.

iii. To develop another heuristic technique by reversing the edges to obtain acyclic graph and assigned the task graph onto n number of processors.

1.5 Research Methodology

This research requires a set of nodes and edges with cycles as a cyclic task graph. There are three models proposed in this study involving the unweighted and weighted directed cyclic graph. Figure 1.3 shows the flow of the algorithm to eliminate the cycles and assign the tasks onto n number of processors in general. The methodology of solving the proposed models can be divided into three main stages. The first stage involves cyclic task graph and identification of the cycle. The second

stage involves the formation of directed acyclic graph by proposed techniques. The last stage is the assignment the nodes (tasks) onto *n* number of processors and obtain the schedule. Finally, the proposed algorithms, along with numerous numbers of nodes, were coded and tested in the Javascript programming language with D3 library with Intel Core i7-3632 QM CPU @ 2.20GHz and 4GB RAM environment.



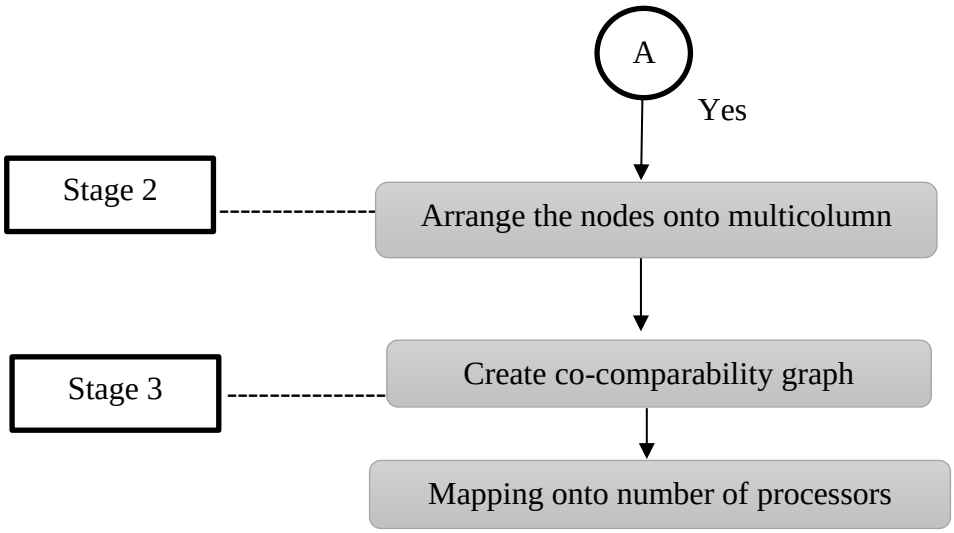


Figure 1.3 The general flow of algorithm to eliminate the cycles and assign the tasks onto processors.

1.6 Scope of Research

The scope of this study is on the development of the algorithm to break the cycle(s) and map the task graph onto two processors and more, $P \geq 2$. Specifically, the algorithms are confined to the proposed task graphs. This study considered only finite simple directed-cyclic graph $G(V, E)$ where $|V| = m$ and $|E| = n$. The cycle involves in a task graph must be three nodes. This study is only based on theoretical graph applications and the results obtained is based on simulation. There are four numerical experiments involving eight, twenty, forty and one hundred number of nodes for each of the proposed model. This research does not involve load balancing issue. Note that Parallel Virtual Machine (PVM) and Message Passing Interface (MPI) are not covered in this study.

1.7 Significance of Research

This research develops algorithms to eliminate the cycles in directed cyclic graph and assign the tasks into several processors. From the viewpoint of the task assignment problem with cyclic task graph, the algorithm in this study is the first implementation of acyclic task graph and mapping onto processors. In the scope of application, the real problem in optimising the reduction of cycle is solved. Theoretically, the proposed algorithms are novel, provide an easy, reliable, and systematic way to solve a cyclic task graph.

A set of nodes with no weightage imposed on the edges is proposed. A model called *DCGSimplify* is developed as a fundamental model to eliminate cycles to become acyclic. This model is believed can provide a reliable technique which can be extended to the weighted directed cyclic graph in the next model.

The integration of good algorithm in eliminating the cycles and good mapping strategy can provide a better understanding of task assignment problem to find the most optimal way of assigning the tasks onto processors (workers). Other models called *Weighted-DCGSimplify* and *Edge Splitting* are proposed with several numerical experiments tested on different number of nodes is proposed to solve more complex problem, so that it can enhance decision making. The schedule produced from the model provides suggestion for the company to take appropriate action improving the assignment of the tasks to all available resources.

This research in addition to partially filing the research gap, provides fundamental solution and better understanding to avoid cycle on assignment problems thus helping enterprises to increase the efficiency in planning as well as reducing the routing cost. It is hoped that all the proposed models be off benefit to manufacturing industry and transportation business in the effort to become more effective where model and algorithms can be adjusted in their software and applications.

The proposed developed algorithm is believed can help the problem which investigates drone-based delivery at a system level. The entities of the system are goods, customers, vehicles, and depots. Customers request goods that are stored in depots and delivered by vehicles. Service requests, also denoted as jobs or customer requests, are not known in advance and arrive over time at certain locations according to a space-time stochastic process. The approach proposed in this study can answer the questions on how many vehicles and depots are needed for a certain area and analyze policies for job assignment such as ‘how to assign customer requests to vehicles to minimize the expected delivery time?’. Nowadays, finding the minimum and cost-effective routing is paramount importance in many businesses. Without a good and efficient routing, it will increase the total cost and would become a major burden to companies. Therefore, even small improvements in routing efficiency can result in large cost reductions and affected especially on industrials involving transportation, manufacturing, and others. This research highly impactful where the new model and algorithms will be transformed into software based and use in graph network problem. If successfully, it will become one of the pillars decision support system where until now this country still dependence on foreign technology for simulation software development.

Finally, four intellectual properties, i.e. two patents and two copyrights could be produced through three proposed models in this research.

1.8 Thesis Organization

The layout of this thesis can be divided into six chapters. The thesis starts from the introduction followed by the literature review, developed models and discussion and end with conclusion and suggestions.

Chapter 1 is mainly about the overview of the thesis. It consists of the introduction, problem background, problem statement, research objectives, research methodology, scope of the study, and the significance of the study as well as thesis organization.

Chapter 2 is the literature review which consists of all the important information gathered throughout the entire course of the study. All related works of task assignment problem and its algorithms with their methodology used are compiled in this chapter.

Chapter 3 discusses on how a directed-cyclic graph can be reduced into the form of directed-acyclic graph for unweighted directed graph. The algorithm of technique is introduced and expressed through a schematic diagram. The algorithm is

coded using JAVA programming language with D3 library. It is tested to different number of nodes.

Chapter 4 presents a multicolumn matching technique to assign the task onto processors. It considers the directed-acyclic graph as in Chapter 3. The algorithm of method is introduced and expressed through a flowchart. The algorithm is coded using JAVA programming language with D3 library. It is tested to different number of nodes.

Chapter 5 presents an edge splitting technique to transform the directed-cyclic graph to a directed-acyclic graph and to assign the task onto processors. It considers the directed-acyclic graph as in Chapter 3. The algorithm of method is introduced and expressed through a flowchart. The algorithm is coded using JAVA programming language with D3 library. It is tested to different number of nodes.

Finally, chapter 6 provides a summary and conclusion of the research. Apart from that, some suggestions for future research are also presented in this chapter.

REFERENCES

Amponsah, S. , Otoo, D., Salhi, S., and Quayson, E. (2016). Proposed Heuristic Method for Solving Assignment Problems. *American Journal of Operations Research*. 6, 436-441. doi: 10.4236/ajor.2016.66040.

Bender, M.A., Fineman, J.T., Gilbert, S., Tarjan, R.E. (2015). A new approach to incremental cycle detection and related problems. *ACM Trans. Algorithms.* 12(2), 1549-6325. https://doi.org/10.1145/2756553.

Błażewicz, J., Domschke, W., and Pesch, E. (1996). The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research*, *93*(1), 1–33.

Bogomolnaia, A. and Moulin, H. (2001). New Solution to the Random Assignment Problem. *Journal of Economic Theory*. 100, 295-328.

Bozejko, W., Pempera, J., & Wodecki, M. (2017). A fine-grained parallel algorithm for the cyclic flexible job shop problem. *Archives of Control Sciences, 27*(2), 169-181. doi:10.1515/acsc-2017-0010.

Bozejko, W., Gnatowski, A., Klempous, R., Affenzeller, M., & Beham, A. (2017). Cyclic scheduling of a robotic cell. Paper presented at the *7th IEEE International Conference on Cognitive Infocommunications, CogInfoCom 2016 - Proceedings*, 379-384. doi:10.1109/CogInfoCom.2016.7804579.

Bui, T. N., Chaudhuri, S., Leighton, F. T., & Sipser, M. (1987). Graph bisection algorithms with good average case behavior. *Combinatorica*, 7(2), 171-191. https://doi.org/10.1007/BF02579448.

B.W.Kernighan and S.Lin. (1970). An efficient Heuristic procedure for partitioning graphs. *Bell Systems Tech, J*. 49, 291-308.

Carlier, J., & Chrétienne, P. (1988). *Problèmes d'ordonnancement: modélisation, complexité, algorithmes*. Paris: Masson.

Chien-chung Shen and Wen-Hsiang Tsai. (1985). A Graph Matching Approach to Optimal task assignment in Distributed computing systems using a Minimax Criterion. *IEEE Transactions on Computers*, 34(3), 197-203.

Cimen, Z. (2001). A Multi-Objective Decision Support Model for the Turkish Armed Forces Personnel Assignment System. Department of Operational Sciences. Air Force Institute of Technology, Ohio.

Dell'Amico, M., & Trubian, M. (1993). Applying tabu search to the job-shop scheduling problem. *Annals of Operations Research*, 41(3), 231–252.

Draper, D. L., Jonsson, A. K., Clements, D. P., Joslin, D. E. (1999). Cyclic scheduling. In *IJCAI*, Citeseer, 1016–1021.

Elmi, A., & Topaloglu, S. (2017). Cyclic job shop robotic cell scheduling problem: Ant colony optimization. *Computers and Industrial Engineering*, 111, 417-432. doi:10.1016/j.cie.2017.08.005.

Garey, M., & Johnson, D. (1979). Computers and intractability. New York: WII Freeman and Company.

Gopalan, N. P., & Suresh, S. (2015). Modified delay scheduling: A heuristic approach for hadoop scheduling to improve fairness and response time. *Parallel Processing Letters*, 25(4) doi:10.1142/S0129626415500097.

Hafnaoui I, Rabeh A, gabriela N, Giovanni B. (2017). Scheduling real time systems with cyclic independence using data criticality. *Des Autom Embed Syst.* 21:117-136.

Hanen, C. (1994). Study of a NP-hard cyclic scheduling problem: The recurrent job-shop. *European Journal of Operational Research*, 72(1), 82–101.

Haeupler, B.; Kavitha, T.; Mathew, R.; Sen, S.; Tarjan, R.E. (2017). Incremental Cycle Detection, Topological Ordering,and Strong Component Maintenance. Available online: https://arxiv.org/abs/1105.2397.

H. El-Rewini and T. G. Lewis. (1994). Scheduling parallel program tasks onto arbitray target machines. *Journal of Parallel and Distributed Computing.* 9(2):138–153.

H. Tong, C. Faloutsos. (2006). Center-piece Subgraphs: Problem Definition and Fast Solutions. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and data Mining, ACM.* New York, NY, USA, pp. 404-413.

I.Gutman, S. Wegner. (2012). The Matching Energy of a Graph. *Discrete Appl. Math.* 160, 2177-2187.

Janardhanan, M. N., Li, Z., & Nielsen, P. (2019). Model and migrating birds optimization algorithm for two-sided assembly line worker assignment and balancing problem. *Soft Computing*, 23(21), 11263-11276. doi:10.1007/s00500-018-03684-8.

J Sun, D Ajwani, PK Nicholson, A Sala, S Parthasarathy. (2017). Breaking cycles in noisy hierarchies. *Proceedings of the 2017 ACM on Web Science Conference*. 151-160

J.R. Ullmann. (1976). *The Journal of the ACM* 23 (1), 31-42.

Kim, H. J., Lee, J. H. (2018). Cyclic robot scheduling for 3D printer-based flexible assembly systems. *Annals of Operations Research*, 1–21.

Korbaa, O., Camus, H., & Gentina, J. C. (2002). A new cyclic scheduling algorithm for flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, *14*(2), 173–187.

Kühn, D.; Osthus, D. (2012). A survey on hamilton cycles in directed graphs. *Eur. J. Combin.*, 33, 750–766.

L.P. Cordella, P. Foggia, C. Samsone, M. Vento. (2004). *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(10), 1367-1372.

L. Shuli, Y. Weigen. (2013).The Matching Energy of Graphs with Given Parameters, *Discrete Applied Mathematics*, 162, 415-420.

Li, B., Zhang, S. (2016). Heavy subgraph conditions for longest cycles to be heavy in graphs. *Discuss. Math.Gr. Theory*, 36, 383–392.

Li, B.; Xiong, L.; Yin, J. (2016). Large degree vertices in longest cycles of graphs, I. *Discuss. Math. Gr. Theory*, 36, 363–382.

Lv, X.; Zhu, D. (2010). An approximation algorithm for the shortest cycle in an undirected unweighted graph. *In Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering*, Changchun, China. IEEE: New York, NY, USA, 297–300.

M. Fuji, T. Kasami, K. Ninomiya. (1969). Optimal Sequencing of Two Equivalent Processors. *SIAM Journal of Appl. Math.*, 17(4).

Mangesh Gupte, Pravin Shankar, Jing Li, S. Muthukrishnan, and Liviu Iftode. (2011). Finding Hierarchy in Directed Online Social Networks. *In Proceedings of the 20th International Conference on World Wide Web (WWW '11)*. New York, NY, USA, 557--566.

Paulusma, D.; Yoshimoto, K. (2007). Cycles through specified vertices in triangle-free graphs. *Discuss. Math. Gr. Theory.* 27, 179–191.

Quinton, F., Hamaz, I., & Houssin, L. (2020). A mixed integer linear programming modelling for the flexible cyclic jobshop problem. *Annals of Operations Research, 285*(1-2), 335-352. doi:10.1007/s10479-019-03387-9.

Schutten, J. M. (1998). Practical job shop scheduling. *Annals of Operations Research*, *83*, 161–178.

Reif, J.H. (1985). Depth-first search is inherently sequential. *Inform. Process. Lett*. 20, 229–234.

R. Mohan, A. Gupta. (2013). Graph Matching Algorithm for Task assignment Problem, *International Journal of Computer Science, Engineering and Applications (IJCSEA*). Vol. 1, No. 6.

R.Mohan and N.P.Gopalan. (2013). A Modified Parallel Heuristic Graph Matching Approach for Solving Task Assignment Problem in Distributed Processor System . *International Journal of Information Technology and Computer Science*, 5(10), 78-84.

Silva, J.L.C.; Rocha, L.; Silva, B.C.H. (2016). A new algorithm for finding all tours and hamiltonian circuits in graphs. *IEEE Lat. Am. Trans.* 14, 831–836.

Spieker, H., Gotlieb, A., & Mossige, M. (2018). Different cycle, different assignment: Diversity in assignment problems with multiple cycles. Paper presented at the *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 8161-8162.

S.Zhang, S. Li, J. Yang, Gaddi. (2009). Distance Index Based Subgraph Matching in Biological Net- works. *Proceedings of the 12th International Conference on Extending Database Technology ACM.* New York, NY, USA, 192-203.

WH Tsai, KS Fu .(1979). Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Transactions on systems, man, and cybernetics*. 9 (12), 757-768.

Yilmaz, H., & Yilmaz, M. (2020). A mathematical model and tabu search algorithm for multi-manned assembly line balancing problems with assignment restrictions. *Engineering Optimization, 52*(5), 856-874. doi:10.1080/0305215X.2019.1618288.

Yuster, R. (2011). A shortest cycle for each vertex of a graph. *Inform. Process. Lett.,* 111, 1057–1061.

Y. Liao, J. Yin, D. Yin, L. Goa. (2012). DPillar: Dual-port Server Interconnection Network for Large Scale Data Centers, *Computer Networks,* 56, 2132-2147.

Y. Tian, J.M. Patel (2008). Tale: A Tool for Approximate Large Graph Matching. *Proceedings of the 24th International Conference on Data Engineering, IEEE Computer Society.* Washington, DC, USA, 963-972.

Z.Linhong, Wee K.N., J. Cheng. (2012). Structure and attribute for approximate graph matching in large graphs, *Information Systems.* Vol. 36, Pages 958-972.

LIST OF PUBLICATIONS

1. **W.N.M Ariffin**, S. Mat Zin, S. Salleh (2012). Shortest Path Technique for Switching in a Mesh Network. *International Journal of Modern Physics: Conference Series, World Scientific*. Vol. 9 (2012) 488–494.

2. **W.N.M Ariffin**, S. Salleh. (2014). The matching technique of directed cyclic graph for task assignment problem. *AIP Conference Proceedings*. 1635, 387. **(Indexed by Scopus).**

3. **W.N.M Ariffin**, S. Salleh. (2015). Task Scheduling for Directed Cyclic Graph Using Matching Technique. *Contemporary Engineering Sciences*. 8 (17), 773-788. **(Indexed by Scopus).**

4. **W.N.M Ariffin**, S. Salleh. (2016). The partitioning technique of directed cyclic graph for task assignment problem. *AIP Conference Proceedings*. 1750, 020010**. (Indexed by Scopus).**

5. **W.N.M Ariffin**, S. Salleh. (2016). Bi-partition approach of directed cyclic task graph onto multicolumn processors for total completion time minimization task assignment problem. *AIP Conference Proceedings.* 1775 (1), 030072. **(Indexed by Scopus).**

6. **W.N.M Ariffin**, S. Salleh. (2018). The Reduction of Directed Cyclic Graph for Task Assignment Problem. *MATEC Web of Conferences, EDP Sciences*. 150, 06031**. (Indexed by Scopus).**

INTELLECTUAL PROPERTIES

1. Patent File: PI2017705114. The New Technique to Reduce Directed Cyclic Graph to Acyclic.

2. Patent File: PI2020004798. Ranking And Edge Splitting Technique For Cyclic Graph In Task Assignment Problem.

3. Copyright: LY2020006328. *DCGSimplify* Programming Codes On Transforming Directed Cyclic Graph Onto Acyclic.

4. Copyright: LY2020006329. Task Assignment Of Two Resources Program.

RESEARCH EXHIBITIONS AND AWARDS

1. **Gold Award.** Malaysia Technology Expo 2021. Product: π-Assistant - Promotion Item Assistant. Tool To Help Retail Industries To Set Optimal Promotion Items.

2. **Gold Award.** Ekspo Penyelidikan Dan Rekacipta UniMAP 2021 (eREKA2021) . Product: π-Assistant - Promotion Item Assistant. Tool To Help Retail Industries To Set Optimal Promotion Items. (Representing Universiti Malaysia Perlis to Malaysia Technology Expo 2021).

3. **Gold Award.** International Invention, Innovation And Technology Exhibition Malaysia 2020 (ITEX2020). Product: T-Deck: Task & Schedule Generator.
4. **Silver Award.** Ekspo Penyelidikan Dan Rekacipta UniMAP 2020 (eREKA2020). Product: Taskdeck: Find Your Optimal Solution.

5. **Silver Award.** International Invention, Innovation And Technology Exhibition Malaysia 2019 (ITEX2019). Product: Route Solver: Generate Optimal Schedule for Task Assignment. Save Cost, More Profit.

6. **Gold Award.** Ekspo Penyelidikan Dan Rekacipta UniMAP 2018. (eREKA2018). Product: ROUTE SOLVER: Avoiding Cycles. Generate Good Schedule For Your Task Assignment Problems.

7. **Gold Award.** International Engineering Invention and Exhibition 2018 (i-ENVEX 2018). Product: Cyclic Solver: Generate Good Task Scheduler.

8. **Silver Award.** Ekspo Penyelidikan Dan Rekacipta UniMAP 2017 (eREKA2017). Product: *DCGSIMPLIFY*: A New Technique For Minimizing Task Scheduling System Based On Theoretical Graph.

CONFERENCE PRESENTATION

1. Shortest Path Technique for Switching in a Mesh Network. International Conference on Mathematics and Computational Biology, Melaka. (2012).

2. The Matching Technique Of Directed Cyclic Graph For Task Assignment Problem. International Conference on Quantitative Science and Industrial Applications, Langkawi. (2014).

3. The Partitioning Technique Of Directed Cyclic Graph For Task Assignment Problem. Simposium Kebangsaan Sains dan Matematik. Johor Bahru. (2015).

4. Bi-Partition Approach Of Directed Cyclic Task Graph Onto Multicolumn Processors For Total Completion Time Minimization Task Assignment Problem. International Conference on Mathematics, Engineering and Industrial Applications. Thailand. (2016).

5. The Reduction of Directed Cyclic Graph for Task Assignment Problem. Malaysian University Conference on Engineering and Technology. (2017).