A Graph Partitioning Algorithm for the Berth Allocation Problem

David Ong Tat-Wee
B.Sc. (Computer & Information Sciences) (Honours)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
2000

Acknowledgements

Eternal thanks to my God for meeting all my needs according

to His glorious riches in Christ Jesus (Phil 4:19).

I wish to specially thank my mentor Assoc. Prof. Leong Hon Wai whose assiduous pedagogy all these years has certainly edified my mind for the challenges in this new millennium. I also thank Asst. Prof. Ang Chuan Heng for being in my M.Sc. advisory committee. His erudite counsel has been invaluable.

To my peers: Hee Meng for sharing his knowledge and references — the bibliography would not be complete otherwise; Chee Wai for supplying the Latex template for this thesis; Swee Seong for his water heater; Shen Tat and Seet Chong for MAME; and Aik Hui, Wee Yeh, et. al. for the fun times in the former DISCS.

Salut des RAS Garage members, and to Steven, Somnath and Daniel for their support and encouragement.

Contents

List of Figures

List of Tables

Summary

In this thesis, we studied the berth allocation problem (BAP) in a container transshipment port. We formulated the BAP as a generalisation of the multi-way graph partitioning problem and developed a graph partitioning algorithm to solve it. Our BAP-GP algorithm iteratively improves on an initial solution until it reaches a local optimal. We studied different methods of generating initial solutions and found that those which focused on transshipment volumes produced better solutions. We also looked into a subproblem dealing with the feasibility of assigning vessels in a busy time zone. This was modelled as an instance of the multiple knapsack problem and a simple pairwise exchange heuristic was developed to deal with it. We put our algorithms into a BAP partitioning framework which forms part of a larger BAP planning system (BAPS). The BAPS allows the user to generate BAP data sets, perform experimental runs on them and analyse the results obtained. Experimental results show that our BAP partitioning algorithm has a very fast running time and performs promisingly compared with the other algorithms in the BAPS.