Mathematical Modelling and Efficient Algorithms for Autonomous Straddle Carriers Planning at Automated Container Terminals

by

Shuai Yuan

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

University of Technology, Sydney
Faculty of Engineering

October 2013

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of a requirement for a degree except as fully acknowledged within the text.

I certify that the thesis has been written by me. Any help that I have received in my research work and preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

(Shuai Yuan)

Sydney, October 2013

Abstract

In the past several decades, automation of handling equipment has been a worldwide trend in seaport container terminals. Increasing automation of yard handling vehicles not only reduces the cost of terminal operation, but also increases the efficiency of container transport. However, the primary loss of performance in the transhipment process is caused by the uncoordinated allocation and scheduling of quay cranes, yard vehicles and land-side operations. Hence, integrating transhipment processes is imperative for a fully automated container terminal. This thesis aims to study an integrated process and develop practically deployable strategies and algorithms, with the practical example of the Patrick AutoStrad container terminal, located in Brisbane, Australia.

The thesis first formulates two mathematical models: The *Comprehensive Model* is an analytical optimisation model which integrates the quay-side, yard and land-side operational sub-problems of the Patrick AutoStrad container terminal. Derived from the comprehensive model, the *Job Scheduling Model* is formulated to focus on the optimisation of job scheduling, as job scheduling plays a more important role than path planning, and resource utilisation and port operation are more dependent on job scheduling.

To solve the *Comprehensive Model*, a job grouping approach is proposed for solving the integrated problem, and experimental results show that the job grouping approach can effectively improve the time related performance of planning container transfers. Solving the *Job Scheduling Model* using a global optimisation approach is expected to provide higher productivity in automated container terminals. Hence, a modified genetic algorithm is proposed for solving the job scheduling problem derived from the integrated mathematical model of container transfers. Moreover, the live testing results show that the proposed algorithm can effectively reduce the overall time-related cost of container transfers at the automated container terminal.

Last but not least, a new crossover approach is proposed in order to further improve the solution quality based on the modified genetic algorithm, and it can also be directly

applied in solving the generic multiple travelling salesmen problem using the two-part chromosome genetic algorithm. The experimental results also show that the proposed crossover approach statistically outperforms the existing approaches when solving the job scheduling problem and the standard multiple travelling salesmen problem.

Acknowledgements

This thesis would never have been written without the support of the people who have enriched me through their wisdom, friendship and love.

I would like to express my deepest appreciation to my supervisors: Professor Dikai Liu and Associate Professor Shoudong Huang. I really appreciate their guidance, encouragement and constructive advice on modelling and algorithms. They have provided me with invaluable help throughout the course of my research.

I would like to thank my Patrick project mentors: Dr Bradley Skinner and Dr Haye Lau for all the hours of collaboration, insightful ideas and research inspiration.

I would like to thank all the other team members from UTS and PTS: Professor Gamini Dissanayake, Dr Binghuang Cai, Daniel Pagac, Timothy Pratley and Andrew Bott for their helpful insight along with their patience and passion for autonomous straddle carriers.

Last, but not the least, I am deeply indebted to my parents, Xijia and Shuhua, for the years of education, encouragement and care; my beloved wife Momo, for her unwavering support; and my dearest son Shengbo for imbuing my life with wonderful expectations.

To the people who helped

List of Abbreviations

ASC:	Autonomous Straddle Carrier
QC:	Quay Crane
TK:	Truck
TC:	Transfer Crane
TIA:	Truck Import Area
B2Y:	Buffer-to-Yard
Y2B:	Yard-to-Buffer
T2Y:	Truck-to-Yard
Y2T:	Yard-to-Truck
Y2Y:	Yard-to-Yard
VRP:	Vehicle Routing Problem
CVRP:	Capacitated Vehicle Routing Problem
PDP:	Pickup and Delivery Problem
QCSP:	Quay Crane Scheduling Problem
QCAP:	Quay Crane Allocation Problem
BAP:	Berth Allocation Problem

AGV:	Automated Guided Vehicle
ALV:	Automated Lifting Vehicle
ASCSP:	Autonomous Straddle Carrier Scheduling Problem
MTSP:	Multiple Travel Salesmen Problem
GA:	Genetic Algorithm
TCX:	Two-part Chromosome Crossover
ORX:	Ordered Crossover
PMX:	Partially Matched Crossover
CYX:	Cycle Crossover
ORX+A:	Combination of Ordered Crossover and Asexual Crossover
PMX+A:	Combination of Partially Matched Crossover and Asexual Crossover
CYX+A:	Combination of Cycle Crossover and Asexual Crossover

Nomenclature

Indices:

v	index of the ASC.
q	index of the QC.
g	index of the TK.
j	index of the job.
n	index of the positional node which indicates a position on the map.
l	index of the link between positional nodes on the map.
s	index of the schedule. Each schedule contains a list of jobs which need to be finished in order, and each schedule can be assigned to any ASC, but one schedule cannot be shared by multiple ASCs. It is not a complete solution, but it can be a possible sub-solution for some jobs.

Sets:

V	set of all ASCs, $\|V\|$ is the total number of ASCs in the fleet.
Q	set of all QCs, $\|Q\|$ is the total number of QCs.
G	set of all TKs, $\|G\|$ is the total number of TKs.
J	set of all jobs, $\|J\|$ is the total number of jobs.
J^H	set of jobs with high priority for scheduling and these jobs usually are time critical or considered as urgent tasks and $J^H \subset J$.
S	set of all schedules.
J^Y	set of yard-to-yard jobs.
δ_v	planned trajectory of ASC (v) including the travelling nodes and associated times. Please refer to Section 3.4.1 for more detail.
N	set of all positional nodes, $\|N\|$ is the total number of nodes in the map.
L	set of all links, $\|L\|$ is the total number of links on the map.

Vectors:

J_s — vector of jobs in schedule s. That is, a list of jobs that will be done by an ASC in the defined order.

J_q^D — vector of jobs associated with discharging QC q.

J_q^U — vector of jobs associated with uploading QC q.

J_g^E — vector of jobs associated with exporting TK g.

J_g^I — vector of jobs associated with importing TK g.

J_v^G — vector of potential jobs grouped by (v).

$N^{lock}(n)$ — vector of locked nodes associated with the node (n) of an ASC to prevent all vehicle collisions.

Parameters:

m_q — the total number of discharging containers for QC q.

n_q — the total number of uploading containers for QC q.

e_g — the total number of containers for exporting TK g.

r_g — the total number of containers for importing TK g.

p_v^I — the initial position of ASC v.

α_{ab} — parameter for defining pick-up job sequence. $\alpha_{ab} = \{0,1\} : \forall(a,b)$. Let $\alpha_{ab} = 1$ if and only if job a needs to be picked up before job b is picked up, else $\alpha_{ab} = 0$.

β_{ab} — parameter for defining set-down job sequence. $\beta_{ab} = \{0,1\} : \forall(a,b)$. Let $\beta_{ab} = 1$ if and only if job a needs to be set-down before job b is set-down, else $\beta_{ab} = 0$.

u_j — the pick-up position of job j. The position is generated by the yard management system.

d_j — the set-down position of job j. The position is generated by the yard management system.

y_{js} a binary parameter. $y_{js} = 1$ if and only if the candidate schedule *s* has job *j*, otherwise $y_{js} = 0$.

t_0 plan starting time.

Δt_{pickup} time required for an ASC to pick up a container.

$\Delta t_{setdown}$ time required for an ASC to set down a container.

Δt_{acc} time associated with acceleration of an ASC moving from zero velocity to maximum (constant) velocity.

Δt_{dec} time associated with deceleration of an ASC from its maximum (constant) velocity to zero velocity.

Δt^{QC} the turnaround time of each QC unloading/uploading a single container.

Δt^{TK} the turnaround time of an ASC performing TK importing/exporting for a single container within the TK area.

t_q^D the starting time of QC *q* operations (discharging/unloading) which is predefined a priori as part of the QC schedule. For each discharging QC we assume that no container from QC *q* can be picked up by an ASC at the buffer before the time ($t_q^D + \Delta t^{QC}$).

t_q^U the starting time of QC *q* operations (uploading) which is predefined a priori as part of the QC schedule. For each uploading QC we assume that no container can be set-down at buffer by an ASC before the time ($t_q^U + \Delta t^{QC}$).

t_g^E the starting time of TK *g* operations (exporting) which is predefined a priori as part of the TK schedule. For each exporting TK we assume that no container from TK *g* can be picked up by an ASC at the TK gate before the time ($t_g^E + \Delta t^{TK}$).

t_g^I the starting time of TK *g* operations (importing) which is predefined a priori as part of the TK schedule. For each importing TK we assume that no container from TK *g* can be set-down by an ASC at the TK gate before the time ($t_g^I + \Delta t^{TK}$).

w_{oz} the minimum theoretical travel time between node o and node z. A look-up table built based on Dijkstra's algorithm is used for all positional nodes.

σ_j^R the predefined parameter which specifies the container door alignment at the setdown node (d_j) for job (j). Let σ_j^R=1 if the container door is aligned, otherwise σ_j^R=0. Here, $\sigma_j^R \in \{0,1\}$, depending on the actual job type.

σ_j^S the predefined parameter which specifies the container door's initial alignment at the pickup node (u_j) for job (j).Let σ_j^S=1 if the container door is aligned, otherwise σ_j^S=0. Here, $\sigma_j^S \in \{0,1\}$, depending on the actual job type.

f_{ij} a flip flag of each physical link which relates the container alignment during traversal of the link. Let $f_{ij}=1$ if container alignment is changed when an ASC carrying a container from node ($i \in N$) to node ($j \in N$), otherwise $f_{ij}=0$. The flip flag of each link is encoded within the map.

ψ_{abc} Patrick yard predefined traversal information between ($a \rightarrow b \rightarrow c$). Let $\psi_{abc}=1$ if an ASC needs to decelerate and adjust direction and then accelerate for traversing from node a to node c via node b, otherwise, $\psi_{abc}=0$. This information is encoded within the map.

ε_j the minimum theoretical processing time based on Dijkstra's algorithm (from pickup to setdown) for job (j).

Decision variables:

X_{vs} binary decision variable. $X_{vs}=1$ if and only if the candidate schedule s is selected for ASC v, otherwise $X_{vs}=0$.

T_j^S the planned start time of job j.

T_j^F the planned finish time of job j.

T_q^{DS} the planned time to pick-up the final container for discharging QC q.

T_q^{UF} the planned time to set-down the final container for uploading QC q.

T_g^{ES} the planned time to pick-up the final container for exporting TK g.

T_g^{IF} the planned time to set-down the final container for importing TK g.

Dependent variables

$P_v(t)$	position of ASC (v) at time ($t \geq t_0$), according to the trajectory planning.
σ_j^F	dependent variable which is the final box alignment associated with the trajectory planning of job (j), and $\sigma_j^F \in \{0,1\}$.

Table of Contents

List of Figures

List of Tables