

# **Studies in Computational Intelligence**

Volume 772

## **Series editor**

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland  
e-mail: [kacprzyk@ibspan.waw.pl](mailto:kacprzyk@ibspan.waw.pl)

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at <http://www.springer.com/series/7092>

Anis Koubaa · Hachemi Bennaceur  
Imen Chaari · Sahar Trigui  
Adel Ammar · Mohamed-Foued Sriti  
Maram Alajlan · Omar Cheikhrouhou  
Yasir Javed

# Robot Path Planning and Cooperation

Foundations, Algorithms  
and Experimentations



Springer

Anis Koubaa  
Prince Sultan University  
Riyadh  
Saudi Arabia

Hachemi Bennaceur  
College of Computer and Information Sciences  
Al Imam Mohammad Ibn Saud Islamic University  
Riyadh  
Saudi Arabia

Imen Chaari  
University Campus of Manouba  
Manouba  
Tunisia

Sahar Trigui  
University Campus of Manouba  
Manouba  
Tunisia

Adel Ammar  
College of Computer and Information Sciences  
Al Imam Mohammad Ibn Saud Islamic University  
Riyadh  
Saudi Arabia

Mohamed-Foued Sriti  
College of Computer and Information Sciences  
Al Imam Mohammad Ibn Saud Islamic University  
Riyadh  
Saudi Arabia

Maram Alajlan  
College of Computer and Information Sciences  
Al Imam Mohammad Ibn Saud Islamic University  
Riyadh  
Saudi Arabia

Omar Cheikhrouhou  
College of Computers and Information  
Technology  
Taif University  
Taif  
Saudi Arabia

Yasir Javed  
College of Computer and Information Sciences  
Prince Sultan University  
Riyadh  
Saudi Arabia

ISSN 1860-949X                      ISSN 1860-9503 (electronic)  
Studies in Computational Intelligence  
ISBN 978-3-319-77040-6            ISBN 978-3-319-77042-0 (eBook)  
<https://doi.org/10.1007/978-3-319-77042-0>

Library of Congress Control Number: 2018934389

© Springer International Publishing AG, part of Springer Nature 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

The objective of the book is to provide the reader with a comprehensive coverage of two important research problems in mobile robots, namely global path planning and cooperative multi-robots applications with a focus on multi-robot task allocation (MRTA) problem. As such, this book is organized in two major parts: Global Path Planning, and Multi-Robot Task Allocation. The objective of the first part of the book is to respond to a research question that we have been investigating along the two-year period of the iroboapp project: considering the vast array of AI techniques used to solve the robot path planning problem ranging from evolutionary computation techniques (e.g. GA, ACO) to meta-heuristic methods (e.g. A\*), which technique is the best? In this part, we first revisit the foundations and present a background of the global path planning problem, and the underlying intelligent techniques used to solve it. Then, we present our new intelligent algorithms to solve these problems, based on common artificial intelligence approaches, and we analyze their complexities. Different simulation models using C++, MATLAB and others have been devised. An extensive comparative performance evaluation study between the path planning algorithms is presented. In addition, we validate our results through real-world implementation of these algorithms on real robots using the Robot Operation System (ROS). The second part of the book deals with cooperative mobile robots. We focus on the multi-robot task allocation (MRTA) problem and we present a comprehensive overview on this problem. Then, we present a distributed market-based mechanism for solving the multiple depot, multiple travel salesman problem which is a typical problem for several robotics applications. A major contribution of this book is that it bridges the gap between theory and practice as it shows how to integrate the global path planning algorithms in the ROS environment and it proves their efficiency in real scenarios. We believe that this handbook will provide the readers with a comprehensive reference on the

global path planning and MRTA problems starting from foundations and modeling, going through simulations and real-world deployments. Links to videos and demonstrations will be included in the book.

Riyadh, Saudi Arabia  
Riyadh, Saudi Arabia  
Manouba, Tunisia  
Manouba, Tunisia  
Riyadh, Saudi Arabia  
Riyadh, Saudi Arabia  
Riyadh, Saudi Arabia  
Taif, Saudi Arabia  
Riyadh, Saudi Arabia

Anis Koubaa  
Hachemi Bennaceur  
Imen Chaari  
Sahar Trigui  
Adel Ammar  
Mohamed-Foued Sriti  
Maram Alajlan  
Omar Cheikhrouhou  
Yasir Javed

# Acknowledgements

The work is supported by the Robotics and Internet-of-Things (RIOTU) Lab and Research and Initiative Center (RIC) of Prince Sultan University, Saudi Arabia. It also supported by Gaitech Robotics in China. Part of this work was previously supported by the iroboapp project “Design and Analysis of Intelligent Algorithms for Robotic Problems and Applications” under the grant of the National Plan for Sciences, Technology and Innovation (NPSTI), managed by the Science and Technology Unit of Al-Imam Mohamed bin Saud University and by King Abdulaziz Center for Science and Technology (KACST).

# Contents

## Part I Global Robot Path Planning

<b>1 Introduction to Mobile Robot Path Planning</b>	3
1.1 Introduction	3
1.2 Overview of the Robot Path Planning Problem	4
1.2.1 Problem Formulation	5
1.3 Path Planning Categories	7
1.4 Spatial Representations Commonly Used in Path Planning	8
1.4.1 Environment Characterization	9
1.4.2 Path Planning Complexity	10
1.5 Conclusion	10
References	11
<b>2 Background on Artificial Intelligence Algorithms for Global Path Planning</b>	13
2.1 Introduction	13
2.2 Classical Approaches	14
2.3 Graph Search Approaches	15
2.3.1 The AStar (A*) Algorithm	15
2.4 Heuristic Approaches	19
2.4.1 Tabu Search	20
2.4.2 Genetic Algorithms	26
2.4.3 Neural Networks	31
2.4.4 Ant Colony Optimization	36
2.4.5 Hybrid Approaches	40
2.4.6 Comparative Study of Heuristic and Exact Approaches	42

2.4.7	Comparative Study of Heuristic Approaches . . . . .	42
2.4.8	Comparative Study of Exact Methods . . . . .	43
2.5	Conclusion . . . . .	45
	References . . . . .	45
<b>3</b>	<b>Design and Evaluation of Intelligent Global Path Planning</b>	
	<b>Algorithms . . . . .</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	System Model . . . . .	54
3.3	Design of Exact and Heuristic Algorithms . . . . .	55
3.3.1	A Relaxed Version of A* for Robot Path Planning . . . . .	55
3.3.2	The Tabu Search Algorithm for Robot Path Planning (TS-PATH) . . . . .	59
3.3.3	The Genetic Algorithm for Robot Path Planning . . . . .	63
3.3.4	The Ant Colony Optimization Algorithm for Robot Path Planning . . . . .	67
3.4	Performance Analysis of Global Path Planning Techniques . . . . .	69
3.4.1	Simulation Environment . . . . .	69
3.4.2	Simulation Results . . . . .	71
3.5	Hybrid Algorithms for Robot Path Planning . . . . .	76
3.5.1	Design of Hybrid Path Planners . . . . .	76
3.5.2	Performance Evaluation . . . . .	78
3.6	Conclusion . . . . .	80
	References . . . . .	81
<b>4</b>	<b>Integration of Global Path Planners in ROS . . . . .</b>	<b>83</b>
4.1	Introduction . . . . .	83
4.2	Navigation Stack . . . . .	85
4.2.1	Global Planner . . . . .	87
4.2.2	Local Planner . . . . .	88
4.3	How to Integrate a New Path Planner as Plugin? . . . . .	89
4.3.1	Writing the Path Planner Class . . . . .	89
4.3.2	Writing Your Plugin . . . . .	94
4.3.3	Running the Plugin . . . . .	97
4.4	ROS Environment Configuration . . . . .	98
4.5	Performance Evaluation . . . . .	99
4.6	Conclusion . . . . .	101
	References . . . . .	101
<b>5</b>	<b>Robot Path Planning Using Cloud Computing for Large Grid Maps . . . . .</b>	<b>103</b>
5.1	Introduction . . . . .	103
5.2	Cloud Computing and Robotics . . . . .	104
5.3	Literature Review . . . . .	104

5.4	Hadoop: Overview . . . . .	106
5.4.1	Hadoop Architecture Overview . . . . .	108
5.5	Giraph: Overview . . . . .	112
5.5.1	Giraph Architecture . . . . .	112
5.5.2	The Bulk Synchronous Parallel Model . . . . .	113
5.6	Implementation of RA* Using Giraph . . . . .	114
5.7	Performance Evaluation . . . . .	118
5.7.1	Cloud Framework . . . . .	119
5.7.2	Experimental Scenarios . . . . .	119
5.7.3	Impact of Number of Workers . . . . .	119
5.7.4	Execution Times . . . . .	120
5.7.5	Total Number of Messages Exchanged, Memory Footprint and CPU Usage of RA* . . . . .	122
5.8	Lessons Learned . . . . .	124
5.9	Conclusion . . . . .	125
	References . . . . .	125

## Part II Multi-robot Task Allocation

<b>6</b>	<b>General Background on Multi-robot Task Allocation . . . . .</b>	<b>129</b>
6.1	Introduction . . . . .	129
6.2	The Multi-robot Task Allocation . . . . .	130
6.2.1	Centralized Approaches . . . . .	131
6.2.2	Distributed Approaches . . . . .	131
6.2.3	Market-Based Approaches . . . . .	132
6.3	The Multiple Traveling Salesman Problem . . . . .	136
6.3.1	MTSP Overview . . . . .	136
6.3.2	Related Works on MTSP . . . . .	137
6.3.3	Multi-objective Optimization Problem (MOP) . . . . .	139
6.4	Conclusion . . . . .	141
	References . . . . .	142
<b>7</b>	<b>Different Approaches to Solve the MRTA Problem . . . . .</b>	<b>145</b>
7.1	Introduction . . . . .	145
7.2	Objective Functions . . . . .	146
7.3	Improved Distributed Market-Based Approach . . . . .	147
7.3.1	Distributed Market-Based (DMB) Algorithm . . . . .	148
7.3.2	Improvement Step . . . . .	150
7.4	Clustering Market-Based Coordination Approach . . . . .	151
7.4.1	CM-MTSP Algorithm Steps . . . . .	152
7.4.2	Illustrative Example . . . . .	153

7.5	Fuzzy Logic-Based Approach . . . . .	156
7.5.1	Fuzzy Logic Rules Design . . . . .	156
7.5.2	Algorithm Design . . . . .	159
7.6	Move-and-Improve: A Market-Based Multi-robot Approach for Solving the MD-MTSP . . . . .	161
7.7	Conclusion . . . . .	166
	References . . . . .	167
<b>8</b>	<b>Performance Analysis of the MRTA Approaches for Autonomous Mobile Robot . . . . .</b>	<b>169</b>
8.1	Introduction . . . . .	169
8.2	Performance Evaluation of the IDMB Approach . . . . .	170
8.2.1	Simulation Study . . . . .	170
8.2.2	Experimentation . . . . .	171
8.3	Performance Evaluation of the CM-MTSP Approach . . . . .	172
8.3.1	Comparison of the CM-MTSP with a Single-Objective Algorithm . . . . .	173
8.3.2	Comparison of the CM-MTSP with a Greedy Algorithm . . . . .	173
8.4	Performance Evaluation of the FL-MTSP . . . . .	177
8.4.1	Impact of the Number of Target Locations . . . . .	177
8.4.2	Impact of the Number of Robots . . . . .	178
8.4.3	Comparison with MDMTSP_GA . . . . .	178
8.4.4	Comparison with NSGA-II . . . . .	179
8.4.5	Comparison Between FL-MTSP, MDMTSP_GA, MTSP_TT, and MTSP_MT Algorithms . . . . .	183
8.4.6	Impact of the TSP Solver on the Execution Time . . . . .	183
8.5	Performance Evaluation of the Move-and-Improve Approach . . . . .	184
8.6	Conclusion . . . . .	187
	References . . . . .	188
	<b>Index . . . . .</b>	<b>189</b>

# Acronyms

2PPLS	Two-phase Pareto local search
A*	The Astar algorithm
ABC	Artificial bee colony
ACO	Ant Colony Optimization
AD*	Anytime Dynamic A*
AM	Application Master
ANA*	Anytime Nonparametric A*
APF	Artificial Potential Field
ARA*	Anytime Repairing A*
BLE	Broadcast of Local Eligibility
BSP	Bulk Synchronous Parallel
CACO	Conventional ACO
CFor	A set of forbidden configuration
CFree	A set of free configuration
CPD	Compressed path databases technique
CYX	Cycle crossover operator
DWA	Dynamic Window Approach
E*	The E Star algorithm
EDA	Estimation of distribution algorithm
FCE	Free configuration eigen-spaces
FIS	Fuzzy Inference System
FMM	Fast marching method
FOD	Front obstacle distance
GA	Genetic Algorithm
GGA	Grouping genetic algorithms
GGA-SS	Steady-state grouping genetic algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
HACO	Heterogeneous ACO
HDFS	Hadoop Distributed File System
IDPGA	Improved dual-population GA

ILS	Iterated Local Search
IWO	Invasive weed optimization
JPS	Jump point search
LOD	Left obstacle distance
MACO	Modified ACO
MD-MTSP	Multiple Depots MTSP
MLP	Multi-Layer Perceptron
MOKPs	Multi-objective Knapsack problems
MOP	Multi-Objective Optimization
MPCNN	Modified pulsecoupled neural network
MRS	Multi-Robot System
MRTA	Multi-Robot Task Allocation
MTD	Maximum Traveled Distance
MT	Maximum tour
MTSP	Multiple Traveling Salesmen Problem
NM	Node Manager
NN	Neural Networks
ORX	Ordered crossover operator
PFM	The Artificial potential field approach
PFM	Potential field method
PMX	Partially-matched crossover operator
PPaaS	Path Planning as a Service
PRM	The probabilistic roadmap method
PSO	Particle Swarm Optimization
QHS	Quad Harmony Search
RA*	Relaxed AStar
RM	Global Resource Manager
ROD	Right obstacle distance
ROS	Robot Operating System
RRT	Rapidly-exploring random tree
RTMA	Robot and Task Mean Allocation Algorithm
S+T	Services and Tasks
SA	Simulated Annealing
SOM	Self Organizing Maps
SP-CNN	Shortest path cellular neural network
SSSP	Single source shortest path algorithm
TSP	Traveling Salesmen Problem
TS	Tabu Search
TTD	Total Traveled Distance
TWD*	Two Way D*
UAV	Unmanned Air Vehicle
VNS	Variable Neighborhood Search
VRP	Vehicle routing problem
YARN	Yet Another Resource Negotiator

# List of Figures

Fig. 1.1	Different issues of path planning . . . . .	5
Fig. 1.2	Workspace and configuration space . . . . .	6
Fig. 1.3	Path Planning Categories . . . . .	7
Fig. 1.4	Spatial representations commonly used in path planning . . . . .	9
Fig. 2.1	Approaches used to solve the path planning problem . . . . .	14
Fig. 2.2	Application of classical and heuristic algorithms [31]. . . . .	19
Fig. 2.3	Simple illustrative example of the Tabu Search algorithm . . . . .	23
Fig. 2.4	5*5 grid map . . . . .	28
Fig. 2.5	Simple illustrative example of the NN basic algorithm in a static environment. Black cells represent obstacles. The maximum number of neighbours in this example is eight. And the transition function used is $g(x) = x/10$ . The shortest path is obtained, in step 7, by following the neighboring node with the largest activity, at each move . . . . .	34
Fig. 2.6	<b>a</b> Ants in a pheromone trail between nest and food; <b>b</b> an obstacle interrupts the trail; <b>c</b> ants find two paths and go around the obstacles; <b>d</b> a new pheromone trail is formed along the shortest path . . . . .	37
Fig. 3.1	A $10 \times 10$ grid environment . . . . .	54
Fig. 3.2	Example of several equivalent optimal paths between two nodes in a G4-grid. Obstacles are in gray . . . . .	59
Fig. 3.3	Insert, remove and exchange moves . . . . .	60
Fig. 3.4	Crossover operators . . . . .	67
Fig. 3.5	Examples of maps used for the simulation . . . . .	70
Fig. 3.6	Box plot of the average path costs and the average execution times (log scale) in $100 \times 100$ , $500 \times 500$ , and $1000 \times 1000$ random maps of heuristic approaches, Tabu Search, genetic algorithms, and neural network as compared to A* and RA* . . . . .	72

Fig. 3.7	Box plot of the average path costs and the average execution times (log scale) in $512 \times 512$ random, $512 \times 512$ rooms, $512 \times 512$ video games, and $512 \times 512$ mazes maps of heuristic approaches Tabu Search, genetic algorithms, and neural network as compared to A* and RA* . . . . .	72
Fig. 3.8	Box plot of the average path costs and the average execution times (log scale) in the different maps (randomly generated and those of benchmark) of heuristic approaches Tabu Search, genetic algorithms, and neural network as compared to A* and RA* . . . . .	73
Fig. 3.9	Average Percentage of extra length compared to optimal path, calculated for non-optimal paths . . . . .	74
Fig. 3.10	Flowchart diagram of the RA* + GA hybrid algorithm . . . . .	78
Fig. 3.11	Average path lengths and average execution times (log scale) of hybrid approach RA*+GA and RA* + TS as compared to A* and RA* . . . . .	79
Fig. 4.1	Example of a ROS computation graph . . . . .	84
Fig. 4.2	Recovery behaviors. . . . .	87
Fig. 4.3	Willow Garage map . . . . .	100
Fig. 4.4	Average execution time (microseconds) of RA* and navfn . . . . .	101
Fig. 5.1	The Hadoop distributed file system architecture . . . . .	109
Fig. 5.2	Parts of a MapReduce job . . . . .	111
Fig. 5.3	The Giraph Architecture . . . . .	113
Fig. 5.4	The BSP Model . . . . .	114
Fig. 5.5	Average execution times of RA* implemented using Giraph/Hadoop for the different grid maps . . . . .	120
Fig. 5.6	Average execution times of the different implementation of RA* and Hadoop initialisation time for $500 \times 500$ , $1000 \times 1000$ and $2000 \times 2000$ grid maps . . . . .	121
Fig. 5.7	Average execution times of RA* implemented using Giraph/Hadoop for $1000 \times 1000$ grid map tested for different RAM sizes . . . . .	122
Fig. 5.8	Number of messages (local and remote) exchanged of RA* for different grid maps . . . . .	122
Fig. 5.9	Memory consumption of RA* implemented using Giraph/Hadoop and RA* implemented using C++ for different grid maps . . . . .	123
Fig. 5.10	CPU Time of RA* implemented using Giraph/Hadoop and RA* implemented using C++ for different grid maps . . . . .	124
Fig. 7.1	<b>a</b> Initial position of the robots and the targets to be allocated. <b>b</b> Messages interchanged between the robots with the appearance of an infinite loop. <b>c</b> Messages interchanged between the robots for the DMB algorithm . . . . .	149

Fig. 7.2	Difference in cost between the solutions obtained with <b>a</b> the Hungarian algorithm, <b>b</b> the DMB algorithm, and <b>c</b> the IDMB algorithm. Blue squares represent the robots and red circles represent the target locations to be visited . . . . .	151
Fig. 7.3	Illustrative example. 2 robots (blue squares) and 5 target locations (red circles) . . . . .	155
Fig. 7.4	Definition of membership functions of the inputs fuzzy sets . . . . .	157
Fig. 7.5	Simulation example with 5 robots and 15 target locations. <b>a</b> Initial position of the robots and the targets to be allocated. The blue squares represent the robots and the red circles represent the target locations. <b>b</b> Tour of each robot after applying the fuzzy logic approach. <b>c</b> Final assignment after redistributing the targets. <b>d</b> Final tour of each robot after applying the TSP GA solver [10] . . . . .	162
Fig. 7.6	Move-and-Improve . . . . .	163
Fig. 8.1	Error in percentage in comparison with the optimal solution for the DMB, the IDMB, and the RTMA algorithms . . . . .	171
Fig. 8.2	Results of the estimated cost of the Hungarian, DMB, IDMB, and RTMA algorithms over 30 simulations per case . . . . .	171
Fig. 8.3	ROS map used for experiments in Prince Sultan University . . . . .	172
Fig. 8.4	<i>TTD</i> of CM_MTSP and CSM_MTSP solutions . . . . .	174
Fig. 8.5	<i>MTD</i> of CM_MTSP and CSM_MTSP solutions . . . . .	174
Fig. 8.6	Mission time of CM_MTSP and CSM_MTSP . . . . .	174
Fig. 8.7	Distribution of targets in the case of 3 and 6 robots . . . . .	175
Fig. 8.8	Comparison results of the CM-MTSP with a greedy algorithm . . . . .	176
Fig. 8.9	Simulation example of the CM-MTSP and the greedy algorithm . . . . .	177
Fig. 8.10	Impact of the number of targets on the total traveled distance and max tour cost (number of robots is fixed) . . . . .	178
Fig. 8.11	Impact of the number of robots on the total traveled distance and max tour cost (number of targets is fixed) . . . . .	179
Fig. 8.12	Comparison between FL-MTSP and the MDMTSP_GA in terms of total traveled distance. The results are shown for a different number of targets with a fixed number of robots . . . . .	180
Fig. 8.13	Comparison between FL-MTSP and the MDMTSP_GA in terms of max tour cost. The results are shown for a different number of targets with a fixed number of robots. The number of robots is 10 in <b>a</b> , 20 in <b>b</b> , and 30 in <b>c</b> . . . . .	181

Fig. 8.14	Time comparison between FL-MTSP and the MDMTSP_GA . . . . .	181
Fig. 8.15	Solutions example obtained for FL-MTSP (blue star) and NSGA-II (red stars) . . . . .	182
Fig. 8.16	Comparison between FL-MTSP, MDMTSP_GA, MTSP_TT, and MTSP_MT. . . . .	184
Fig. 8.17	Time comparison between FL-MTSP using TSP_GA solver and FL-MTSP using TSP_LKH solver. . . . .	184
Fig. 8.18	Total traveled distance versus communication range. . . . .	186
Fig. 8.19	Communication overhead versus communication range . . . . .	186
Fig. 8.20	Ratio of overlapped targets versus communication range . . . . .	187

# List of Tables

Table 1.1	Global and local path planning. . . . .	8
Table 2.1	Different ACO Approaches . . . . .	38
Table 3.1	Average path cost (grid units) for the different algorithms, per environments size. . . . .	70
Table 3.2	Average execution times (microseconds) for the different algorithms, per environment size . . . . .	73
Table 3.3	Percentage of extra length compared to optimal paths, calculated for non-optimal paths. . . . .	74
Table 3.4	Percentage of optimal paths, per environment size. . . . .	74
Table 3.5	Average path costs (grid units) for A*, RA*, RA* + GA, and RA* + TS algorithms, per environment size . . . . .	80
Table 3.6	Average execution times (microseconds) for A*, RA*, RA* + GA, and RA* + TS, per environment size . . . . .	80
Table 4.1	Execution time in (microseconds) and path length in (meters) of RA* and navfn . . . . .	101
Table 5.1	Comparison with some related works. . . . .	107
Table 5.2	Grid Maps Characteristics . . . . .	119
Table 7.1	Bids on clusters $c_1$ and $c_2$ in terms of time . . . . .	155
Table 7.2	Fuzzy rules base. . . . .	158