HYBRID ANT COLONY SYSTEM ALGORITHM FOR STATIC AND DYNAMIC JOB SCHEDULING IN GRID COMPUTING

MUSTAFA MUWAFAK THEAB ALOBAEDY

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2015
Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok
Abstrak


Kata Kunci: Algoritma metaheuristik, Ant colony system, Genetic algorithm, Tabu search, Penjadualan kerja dalam pengkomputeran grid.
Abstract

Grid computing is a distributed system with heterogeneous infrastructures. Resource management system (RMS) is one of the most important components which has great influence on the grid computing performance. The main part of RMS is the scheduler algorithm which has the responsibility to map submitted tasks to available resources. The complexity of scheduling problem is considered as a nondeterministic polynomial complete (NP-complete) problem and therefore, an intelligent algorithm is required to achieve better scheduling solution. One of the prominent intelligent algorithms is ant colony system (ACS) which is implemented widely to solve various types of scheduling problems. However, ACS suffers from stagnation problem in medium and large size grid computing system. ACS is based on exploitation and exploration mechanisms where the exploitation is sufficient but the exploration has a deficiency. The exploration in ACS is based on a random approach without any strategy. This study proposed four hybrid algorithms between ACS, Genetic Algorithm (GA), and Tabu Search (TS) algorithms to enhance the ACS performance. The algorithms are ACS(GA), ACS+GA, ACS(TS), and ACS+TS. These proposed hybrid algorithms will enhance ACS in terms of exploration mechanism and solution refinement by implementing low and high levels hybridization of ACS, GA, and TS algorithms. The proposed algorithms were evaluated against twelve metaheuristic algorithms in static (expected time to compute model) and dynamic (distribution pattern) grid computing environments. A simulator called ExSim was developed to mimic the static and dynamic nature of the grid computing. Experimental results show that the proposed algorithms outperform ACS in terms of best makespan values. Performance of ACS(GA), ACS+GA, ACS(TS), and ACS+TS are better than ACS by 0.35%, 2.03%, 4.65% and 6.99% respectively for static environment. For dynamic environment, performance of ACS(GA), ACS+GA, ACS+TS, and ACS(TS) are better than ACS by 0.01%, 0.56%, 1.16%, and 1.26% respectively. The proposed algorithms can be used to schedule tasks in grid computing with better performance in terms of makespan.

Keywords: Metaheuristic algorithms, Ant colony system, Genetic algorithm, Tabu search, Job scheduling in grid computing.
Acknowledgement

Each part of this study is guided, inspired, and supported by many people. The most important support and guidance were from my research supervisor Prof. Dr. Hjh. Ku Ruhana Ku Mahamud. Thank you very much for your great help and support. It is an honour for me to do a research under your supervision.

I would like to thank all the academic and technical staff in Utara Universiti Malaysia for their help in the study process and providing all the excellent facilities.

Furthermore, I would like to thank all the members of my family for their unconditional support. My goal would not be achieved without them.

Finally, I would like to thank all my friends for their support.
CHAPTER ONE  INTRODUCTION................................. 1
  1.1 Problem Statement ................................................ 8
  1.2 Objective of the Study ............................................ 10
  1.3 Significance of the Study ......................................... 11
  1.4 Scope and Assumption of the Study .............................. 12
  1.5 Thesis Organization ................................................ 12

CHAPTER TWO LITERATURE REVIEW ............................. 14
  2.1 Job Scheduling Algorithms in Computational Grid System ...... 14
    2.1.1 Heuristic Algorithms ........................................ 15
    2.1.2 Evolutionary Algorithms ..................................... 19
    2.1.3 Local Search ................................................ 35
    2.1.4 Swarm Intelligence Algorithms ............................ 44
  2.2 Hybrid Approaches in Job Scheduling ........................... 67
  2.3 Grid Simulator .................................................... 77
  2.4 Conceptual Framework ............................................ 79
  2.5 Summary ............................................................ 81

CHAPTER THREE  RESEARCH METHODOLOGY .................... 83
  3.1 Research Framework .............................................. 83
  3.2 Research Methodology ............................................ 85
    3.2.1 Problem Formulation ........................................ 85
### Chapter Six - Job Scheduling in Dynamic Grid Computing

6.1 Dynamic Environment ................................................................. 152
6.2 Algorithm Parameters ................................................................. 154
   6.2.1 Genetic Algorithm Parameters .............................................. 154
   6.2.2 Ant System Parameters ........................................................ 156
   6.2.3 Ant Colony System Parameters ............................................. 157
   6.2.4 Tabu Search Parameters ...................................................... 157
   6.2.5 BABC, EBABC1, and EBABC2 Parameters .................................. 158
   6.2.6 PSO-GELS Parameters ......................................................... 159
   6.2.7 AS(TS), AS+TS, ACS(TS), and ACS+TS Parameters ...................... 160
   6.2.8 AS(GA), AS+GA, ACS(GA), and ACS+GA Parameters ................... 160
6.3 Experimental Result and Analysis ................................................ 162
   6.3.1 Best Makespan Results ....................................................... 162
   6.3.2 Average Makespan Results .................................................... 164
   6.3.3 Best Flowtime Results ......................................................... 166
   6.3.4 Average Flowtime Results ..................................................... 167
   6.3.5 Best Utilization Results ....................................................... 169
   6.3.6 Average Utilization Results ................................................... 171
   6.3.7 Discussion ........................................................................... 172
6.4 Summary ...................................................................................... 173

### Chapter Seven - Conclusion and Future Work

7.1 Research Contribution ................................................................. 174
7.2 Limitation of the Study ............................................................... 176
7.3 Recommendation for Future Work ................................................. 176
List of Tables

Table 2.1 Difference between each variant algorithm in ACO .................................. 49
Table 2.2 Various research on different Domains and problems .................................. 50
Table 3.1 The implemented algorithms source .......................................................... 112
Table 4.1 Algorithms evaluated with ETC model ....................................................... 115
Table 4.2 Experimental parameters ........................................................................... 122
Table 4.3 ETC matrix for 3 resources and 13 jobs ....................................................... 128
Table 5.1 Algorithms resource for parameter values ............................................... 132
Table 5.2 GA parameter values .................................................................................. 132
Table 5.3 AS parameter value .................................................................................... 133
Table 5.4 ACS parameter values ................................................................................ 134
Table 5.5 TS parameter values .................................................................................... 135
Table 5.6 BABC, EBABC1, EBABC2 parameter values ............................................. 136
Table 5.7 PSO-GELS Algorithm Parameters Values ................................................ 137
Table 5.8 AS, ACS, and TS Algorithms Parameter Values ....................................... 138
Table 5.9 AS(GA) and AS+GA Algorithms Parameter Values ................................. 138
Table 5.10 ACS(GA) and ACS+GA Algorithms Parameter Values ......................... 139
Table 6.1 Datasets descriptions .................................................................................. 153
Table 6.2 Parameters for Generating Dynamic Benchmark ........................................ 153
Table 6.3 Algorithms resource for parameter values ............................................... 154
Table 6.4 GA parameter values .................................................................................. 155
Table 6.5 AS parameter value .................................................................................... 156
Table 6.6 ACS parameter values ................................................................................ 157
Table 6.7 TS parameter values ................................................................................... 157
Table 6.8 BABC, EBABC1, EBABC2 parameter values ............................................. 158
Table 6.9 PSO-GELS Algorithm Parameters Values ................................................ 159
Table 6.10 AS, ACS, and TS Algorithms Parameter Values ....................................... 160
Table 6.11 AS(GA) and AS+GA Algorithms Parameter Values ................................. 161
Table 6.12 ACS(GA) and ACS+GA Algorithms Parameter Values ......................... 161
List of Figures

Figure 2.1: Basic Genetic Algorithm (Zapfel et al., 2010) ..............................................26
Figure 2.2: Visualization of GA population (Zapfel et al., 2010) ......................................26
Figure 2.3: Examples of crossover operators (Zapfel et al., 2010) .....................................28
Figure 2.4: Process of Genetic Algorithm (Zapfel et al., 2010) .......................................29
Figure 2.5: Process of Tabu Search algorithm (Zapfel et al., 2010) ..................................39
Figure 2.6: Tabu Search algorithm pseudocode (Zapfel et al., 2010) .................................41
Figure 2.7: Research conceptual framework ....................................................................80
Figure 3.1: The Research Framework ..............................................................................84
Figure 3.2: The solution vector used by the ants ...............................................................88
Figure 3.3: Solution vectors used by genetic algorithm ....................................................89
Figure 3.4: The new solution vectors produced by crossover operator ...............................89
Figure 3.5: Chromosomes for five tasks and three machines ............................................92
Figure 3.6: ACS(GA) (low level) algorithm pseudocode ..................................................98
Figure 3.7: ACS(TS) (low level) algorithm pseudocode ..................................................101
Figure 3.8: ACS+GA (high level) pseudocode ................................................................105
Figure 3.9: ACS+TS (high level) algorithm pseudocode ................................................108
Figure 4.1: Workload modelling (Feitelson, 2013) ..........................................................119
Figure 4.2: DETC simulator interface ............................................................................122
Figure 4.3: Benchmark for dynamic grid computing .......................................................123
Figure 4.4: Grid computing simulator interface ..............................................................124
Figure 4.5: Simulator charts .........................................................................................125
Figure 4.6: ACS(TS) Schedule table .............................................................................129
Figure 4.7: ACS+TS Schedule table .............................................................................129
Figure 4.8: ACS(GA) Schedule table .............................................................................130
Figure 4.9: ACS+GA Schedule table .............................................................................130
Figure 5.1: Geometric mean for the best makespan values .............................................140
Figure 5.2: The percentage enhancement of each hybrid algorithm in terms of the best makespan values .................................................................141
Figure 5.3: Geometric mean for the average makespan values .......................................142
Figure 5.4: The percentage enhancement of each algorithm in terms of the average makespan values .................................................................143
Figure 5.5: Geometric mean for best flowtime values ....................................................144
Figure 5.6: The percentage enhancement of each algorithm in terms of the best flowtime values .................................................................145
Figure 5.7: Geometric mean for average flowtime values .............................................146
Figure 5.8: The percentage enhancement of each algorithm in terms of the average flowtime values .................................................................147
Figure 5.9: Geometric mean for best utilization value ....................................................148
Figure 5.10: The percentage enhancement of each algorithm in terms of the best utilization values .................................................................149
Figure 5.11: Geometric mean for average utilization values ............................................149
Figure 5.12: The percentage enhancement of each algorithm in terms of the average utilization values .................................................................150
Figure 6.1: Geometric mean for the best makespan values ............................................163
Figure 6.2: The percentage enhancement of each hybrid algorithm in terms of the best makespan values .................................................................164
Figure 6.3: Geometric mean for the average makespan values .......................................165
Figure 6.4: The percentage enhancement of each hybrid algorithm in terms of the average makespan values .......................................................... 165
Figure 6.5: Geometric mean for the best flowtime values .......................... 166
Figure 6.6: The percentage enhancement of each algorithm in terms of the best flowtime values .......................................................... 167
Figure 6.7: Geometric mean for the average flowtime values ....................... 168
Figure 6.8: The percentage enhancement of each hybrid algorithm in terms of the average flowtime values .......................................................... 169
Figure 6.9: Geometric mean for the best utilization value ............................. 170
Figure 6.10: The percentage enhancement of each hybrid algorithm in terms of the best utilization values .......................................................... 170
Figure 6.11: Geometric mean for the average utilization value ...................... 171
Figure 6.12: The percentage enhancement of each hybrid algorithm in terms of the average utilization values .......................................................... 172
List of Appendices

Appendix A Ant Colony System (C# Code).............................................................. 201
Appendix B Genetic Algorithm (C# Code) ............................................................. 204
Appendix C Tabu Search Algorithm (C# Code)..................................................... 213
Appendix D DETC Simulator................................................................................. 220
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>Artificial Bee Colony</td>
</tr>
<tr>
<td>ACO</td>
<td>Ant Colony Optimization</td>
</tr>
<tr>
<td>ACS</td>
<td>Ant Colony System</td>
</tr>
<tr>
<td>ACS(GA)</td>
<td>Low level hybridization between ACS and GA algorithms</td>
</tr>
<tr>
<td>ACS(TS)</td>
<td>Low level hybridization between ACS and TS algorithms</td>
</tr>
<tr>
<td>ACS+GA</td>
<td>High level hybridization between ACS and GA algorithms</td>
</tr>
<tr>
<td>ACS+TS</td>
<td>High level hybridization between ACS and TS algorithms</td>
</tr>
<tr>
<td>AS</td>
<td>Ant System</td>
</tr>
<tr>
<td>AS(GA)</td>
<td>Low level hybridization between AS and GA algorithms</td>
</tr>
<tr>
<td>AS(TS)</td>
<td>Low level hybridization between AS and TS algorithms</td>
</tr>
<tr>
<td>AS+GA</td>
<td>High level hybridization between AS and GA algorithms</td>
</tr>
<tr>
<td>AS+TS</td>
<td>High level hybridization between AS and TS algorithms</td>
</tr>
<tr>
<td>AS_{rank}</td>
<td>Rank-Based Ant System</td>
</tr>
<tr>
<td>BABC</td>
<td>Binary Artificial Bee Colony</td>
</tr>
<tr>
<td>BACO</td>
<td>Balanced Ant Colony Optimization</td>
</tr>
<tr>
<td>BFO</td>
<td>Bacterial Foraging Optimization</td>
</tr>
<tr>
<td>BWAS</td>
<td>Best-Worst Ant System</td>
</tr>
<tr>
<td>cMA</td>
<td>Cellular Memetic Algorithm</td>
</tr>
<tr>
<td>CV</td>
<td>Coefficient of Variation</td>
</tr>
<tr>
<td>DE</td>
<td>Differential Evolution</td>
</tr>
<tr>
<td>DETC</td>
<td>Dynamic Expected Time to Compute</td>
</tr>
<tr>
<td>EA</td>
<td>Evolutionary Algorithms</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>EAS</td>
<td>Elitist strategy for Ant System</td>
</tr>
<tr>
<td>EBABC1</td>
<td>Efficient Binary Artificial Bee Colony</td>
</tr>
<tr>
<td>EBABC2</td>
<td>Efficient Binary Artificial Bee Colony with flexible ranking</td>
</tr>
<tr>
<td>ET</td>
<td>Execution Time</td>
</tr>
<tr>
<td>ETC</td>
<td>Expected Time to Compute</td>
</tr>
<tr>
<td>FANT</td>
<td>Fast Ant System</td>
</tr>
<tr>
<td>FCFS</td>
<td>First Come First Served</td>
</tr>
<tr>
<td>FPLTF</td>
<td>Fastest Processor to Largest Task First</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GA(TS)</td>
<td>Low level hybridization between GA and TS algorithms</td>
</tr>
<tr>
<td>GA+TS</td>
<td>High level hybridization between GA and TS algorithms</td>
</tr>
<tr>
<td>GBF</td>
<td>Genetic Bacterial Foraging</td>
</tr>
<tr>
<td>GSA</td>
<td>Genetic and Simulated Annealing</td>
</tr>
<tr>
<td>GTSP</td>
<td>Generalized Traveling Salesman Problem</td>
</tr>
<tr>
<td>HACO</td>
<td>Hybrid Ant Colony Optimization</td>
</tr>
<tr>
<td>HC</td>
<td>Hill Climbing</td>
</tr>
<tr>
<td>HCACO</td>
<td>Hybrid Converse Ant Colony Optimization</td>
</tr>
<tr>
<td>HGS</td>
<td>Hierarchic Genetic Strategy</td>
</tr>
<tr>
<td>HPDSs</td>
<td>High Performance Distributed Systems</td>
</tr>
<tr>
<td>IACO</td>
<td>Improved Ant Algorithm</td>
</tr>
<tr>
<td>JSP</td>
<td>Job Scheduling Problem</td>
</tr>
<tr>
<td>KPB</td>
<td>K-Parents Best</td>
</tr>
<tr>
<td>LJFR-SJFR</td>
<td>Longest Job to Fastest Resource-Shortest Job to Fastest Resource</td>
</tr>
<tr>
<td>LM</td>
<td>Local Move</td>
</tr>
<tr>
<td>LMCTS</td>
<td>Local Minimum Completion Time Swap</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MA</td>
<td>Memetic Algorithms</td>
</tr>
<tr>
<td>MA+TS</td>
<td>High level hybridization between Memetic and Tabu Search</td>
</tr>
<tr>
<td>MACO</td>
<td>Multiple Ant Colonies Optimization</td>
</tr>
<tr>
<td>MCT</td>
<td>Minimum Completion Time</td>
</tr>
<tr>
<td>MDS</td>
<td>Metacomputing Directory Service</td>
</tr>
<tr>
<td>MET</td>
<td>Minimum Execution Time</td>
</tr>
<tr>
<td>MI</td>
<td>Million Instructions</td>
</tr>
<tr>
<td>MIPS</td>
<td>Million Instructions Per Second</td>
</tr>
<tr>
<td>MMAS</td>
<td>Max-Min Ant System</td>
</tr>
<tr>
<td>MTEDD</td>
<td>Minimum Time Earliest Due Date</td>
</tr>
<tr>
<td>MTERD</td>
<td>Minimum Time Earliest Release Date</td>
</tr>
<tr>
<td>OLB</td>
<td>Optimization Load Balancing</td>
</tr>
<tr>
<td>PGA1</td>
<td>Player’s Genetic Algorithm</td>
</tr>
<tr>
<td>PGA2</td>
<td>Parallel Genetic Algorithm</td>
</tr>
<tr>
<td>PMCT</td>
<td>Player’s Minimum Completion Time</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>PSO-GELS</td>
<td>Particle Swarm Optimization and Gravitational Emulation Local Search</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RGA</td>
<td>Risky Genetic Algorithm</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>SGA</td>
<td>Struggle Genetic Algorithm</td>
</tr>
<tr>
<td>SLM</td>
<td>Steepest Local Move</td>
</tr>
<tr>
<td>SS</td>
<td>Scatter Search</td>
</tr>
<tr>
<td>SSGA</td>
<td>Steady-State Genetic Algorithm</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>SwA</td>
<td>Switching Algorithm</td>
</tr>
<tr>
<td>TS</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>TSP</td>
<td>Traveling Salesman Problem</td>
</tr>
</tbody>
</table>
CHAPTER ONE
INTRODUCTION

The concept of grid systems goes back to 1969 when Leonard Kleinrock wrote, “We will probably see the spread of computer utilities, which, like present electric and telephone utilities, will service individual homes and offices across the country” (Wankar, 2008). From that time, many researchers presented various works and contributed in grid systems fields. The popularity of grid systems started by the late 1990s when Foster developed a grid system called Globus Toolkit (Foster & Kesselman, 1997; 2004).

Grid systems evolves from existing technology such as distributed computing, web service, and Internet (Magoules, Pan, Tan, & Kumar, 2009). According to Xhafa and Abraham (2010), grid computing is defined as “Geographically distributed computers, linked through the internet in a Grid-like manner, which are used to create virtual supercomputers of vast amount of computing capacity able to solve complex problem from e-Science in less time than known before”.

Magoules, Nguyen, and Yu (2009) presented an extensive definition for grid systems as “A hardware and software infrastructure that provides transparent, dependable, pervasive and consistent access to large-scale distributed resources owned and shared by multiple administrative organizations in order to deliver support for a wide range of applications with the desired qualities of service. These applications can perform either: high throughput computing, on-demand computing, data intensive computing, or collaborative computing”.

1
The contents of the thesis is for internal user only
REFERENCES


Chen, J.-M. Gil, & N. Y. Yen (Eds.), *Multimedia and Ubiquitous Engineering* (pp. 17–26). Heidelberg: Springe.


