JOB SCHEDULING APPROACHES BASED ON FIREFLY ALGORITHM FOR COMPUTATIONAL GRID

ADIL YOUSIF ABOALGASSIM ALFAKI

UNIVERSITI TEKNOLOGI MALAYSIA

JOB SCHEDULING APPROACHES BASED ON FIREFLY ALGORITHM FOR COMPUTATIONAL GRID

ADIL YOUSIF ABOALGASSIM ALFAKI

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

APRIL 2013

To my late father ***Yousif Aboalgassim****. May Allah SWT be pleased with him and grant him Al Jannah (Ameen)*

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to ALLAH (SWT), who guided, helped and made my life meaningful even outside the confines of this PhD candidature.

I would like to thank my supervisor Professor Dr. Abdul Hanan Abdullah for his support and guidance throughout my research study. He has been incredibly wise, helpful, understanding, and generous. Also my sincere thanks to my co-supervisor Associate Professor Dr. Sulaiman Mohd Nor, who was always there for me, returning chapters drafts with insightful comments and great helps. I would like to thank him for his patience and personal kindness.

I am grateful to Universiti Teknologi Malaysia for the IDF financial support during my study. I am also thankful to all the staff at Universiti Teknologi Malaysia for their support and help. I would also like to thank University of Kassala, Sudan.

I must express my sincerest and heartiest thanks to my mother for her continuous support and believing in whatever I do. Without her Duaa, love and supports, I would never be able to finish this thesis. Also, I would like to thank all the members of my family, friends and relatives for their constant love and help that moves me towards my goals.

I deeply thank my wife. Her support and understanding have helped me to overcome the most difficult period, encouraging me to forget the bad things and giving me a new hope whenever I was down.

ABSTRACT

Computational Grid emerged to satisfy the rising demand for bandwidth, storage, and computational resources. Job Scheduling on computational grids is identified as NP-hard problem due to the heterogeneity of grid resources. Numerous researches have applied metaheuristics to find polynomial times for the job scheduling problem. These metaheuristics generated good but not optimal schedules. The current metaheuristics suffer from several limitations that cause long makespan time and flowtime. The aim of this research is to design and implement grid job scheduling approaches to map clients' jobs to the available resources in order to finish the submitted jobs within the optimal makespan time and flowtime. This research presents novel static, hybrid static and dynamic metaheuristics approaches based on Firefly Algorithm for grid job scheduling. Based on the review of the available literature, Firefly Algorithm has yet to be applied in the job scheduling on computational grid. Experiments using simulations and real workload traces were conducted to study the performance of the proposed scheduling approaches. Empirical results revealed that the proposed scheduling approaches outperform other scheduling approaches in the case of typical and heavy workloads in terms of both makespan time and flowtime. The average improvement ratios achieved by the static, hybrid static and dynamic scheduling approaches over Genetic Algorithm in the case of makespan time were 23%, 32% and 28% respectively for typical workloads, and 51%, 59% and 42% for heavy workloads. In the case of flowtime, the average improvement ratios were 62%, 81 % and 21% respectively for typical workloads, and 40%, 58% and 57% for heavy workloads.

ABSTRAK

Pengkomputeran Grid muncul bagi memenuhi keperluan jalur lebar, storan dan juga sumber pengkomputeran. Penjadualan kerja bagi pengkomputeran grid dikenalpasti sebagai masalah rumit NP disebabkan kepelbagaian sumber grid. Penyelidikan banyak menumpu mencari anggaran metaheuristik masa polinomial bagi mekanisma penjadualan. Metaheuristik sebegini secara mampu menjana nilai baik, namun tidak menghasilkan jadual yang optima. Metaheuristik masakini mempunyai beberapa kelemahan yang menyebabkan masa *makespan* dan *flowtime* yang panjang. Tujuan penyelidikan ini adalah merekabentuk dan melaksana pendekatan penjadualan kerja grid yang mampu memeta kerja pelanggan kepada sumber yang ada agar kerja tersebut boleh disiap dalam tempoh optima mengikut masa *makespan* dan *flowtime*. Penyelidikan ini membentangkan pendekatan metaheuristik statik, statik hibrid dan dinamik berasaskan Algoritma Kelip-Kelip bagi penjadualan kerja grid. Berdasarkan kajian literatur yang ada, Algoritma Kelip-Kelip masih belum lagi diaplikasi dalam penjadualan kerja grid. Ujikaji secara simulasi mengguna sampel sebenar beban kerja dibuat bagi mengkaji prestasi pendekatan penjadualan yang dicadangkan. Keputusan empirikal menunjukkan pendekatan penjadualan yang dicadangkan mempunyai *makespan* dan *flowtime* yang lebih baik berbanding pendekatan penjadualan yang lain bagi beban kerja lazim dan berat. Nisbah pembaikan purata bagi masa *makespan* diperolehi melalui pendekatan penjadualan statik, statik hybrid dan dinamik dibanding dengan kaedah Algoritma Genetik adalah 23%, 32% dan 28% bagi beban kerja lazim dan 51%, 59% dan 42% bagi beban kerja berat. Bagi *flowtime,* nisbah pembaikan purata adalah 62%, 81 % dan 21% bagi beban kerja lazim dan 40%, 58% dan 57% bagi beban kerja berat.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACO - Ant Colony Optimization
ANOVA - Analysis of Variance
ASP - Application service Providers
BACO - Balanced Ant Colony Optimization
BCO - Bee Colony Optimization
CE - Cross Entropy
DAG - Directed Acyclic Graphs
DE - Differential Evolution
DFAS - Discrete Firefly Algorithm for Static scheduling
DPSO - Discrete Particle Swarm Optimization
EA - Evolutionary Algorithm
FA - Firefly Algorithm
FAAD - Firefly Algorithm for Adaptive Dynamic scheduling
FAFNER - Factoring via Network-Enabled Recursion
FCFS - First Come First Served
GA - Genetic Algorithm
GIS - Grid Information Service
GRAM - Globus Grid Resource Allocation Manager
GWA - Grid Workload Achieve
HDFAS - Hybrid Discrete Firefly Algorithm for Static scheduling
I-WAY - Information-Wide-Area-Year
JVM - Java Virtual Machine
JVM - Java Virtual Machine
LDAP - Lightweight Directory Access Protocol
LJFR - Longest Job on the Fastest Resource
MCT - Minimum Completion Time
MET - Minimum Execution Time

MI - Million Instructions
MIPS - Million Instructions Per Second
MODE - Multi-Objective Differential Evolution
OGSA - Open Grid Services Architecture
OLB - Opportunistic Load Balancing
OS - Operating System
PEs - Processing Elements
PSO - Particle Swarm Optimization
QoS - Quality of Service
RIP - Routing Information Protocol
SI - swarm intelligence
SJFR - Shortest Job on the Fastest Resource
SLA - Service Level Agreement
SOA - Service Oriented Architecture
SPV - Smallest Position Value
SSP - Storage Service Providers
T-RAG - Task Resource Assigning Graph
TS - Tabu Search
TSP - Travelling Salesman Problem
UML - Unified Modeling Language
VOs - Virtual organizations
VQ - Vector Quantization
β_0 - The firefly attractiveness value
γ - The media light absorption coefficient
α - Randomization parameter

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter introduces the context of the work explored in this thesis. It puts forward the fundamental motivations behind studying job scheduling on computational grid and the utilization of swarm intelligence techniques to tackle the scheduling problem. This is followed by a brief background of the problem, the problem statement and its research questions. Then, this chapter presents the objectives, the significance, the contributions and the scope of the research. The final section shows the outlines of the structure of this thesis.

1.2 Research Motivation

The demand for high computational power has grown faster than the increase in hardware capability of processing. Moreover the existing computer power cannot fulfill the rising number of intensive database systems that need special and expensive devices in order to be handled (Foster and Kesselman, 2004; Liu et al., 2010). The result is an ever-increasing shortage of computational resources, as the individual computers separately are less and less able to satisfy these needs. To turn the tide we must look for techniques that result in significant computational gains.

Grid technologies emerged in the middle of 1990s to satisfy the rising demand for bandwidth, storage, and computational resources (Foster et al., 2001).

Grid technologies contributed in solving several computational problems such as exploiting underutilized resources by executing local jobs on remote machines (De Roure et al., 2003). Moreover, grid technologies allow the implementation of parallel CPU capacity by dividing the applications into small parts and sending these parts to a number of parallel CPUs and hence the tasks could complete in a shorter time. Furthermore, grid computing allows access to additional resources and software rather than just the resources and software available in the user organization. This additional software and hardware access has a great impact since it supports the user with expensive hardware and licensed software (Ferreira et al., 2003).

Computational grid is a large scale distributed system consisting of a huge number of heterogeneous resources that belong to different administrative domains. Scheduling jobs in such environments represents a great challenge (Izakian et al., 2009; Liu et al., 2010). Therefore, heuristics and metaheuristics mechanisms have been applied to handle the job scheduling problem on computational grid. However, metaheuristics are commonly able to find good but not necessarily efficient and optimal solutions for the job scheduling problem. Nonetheless, nature-inspired metaheuristics has demonstrated an excellent degree of effectiveness and efficiency for handling combinatorial optimization problems (Zang et al., 2010). The remarkable rise in the size of the solution search space motivated researchers to employ swarm intelligence (SI) mechanisms to solve computational grid scheduling problem. The aim of this research is to design and implement grid job scheduling mechanisms based on a newly introduced SI optimization mechanism to map the clients' jobs to the available resources in order to finish the submitted jobs within minimum makespan and flowtime. This research presents novel static and dynamic optimization scheduling mechanisms based on Firefly Algorithm (FA) for scheduling jobs on grid computing.

1.3 Problem Background

Resource management in grid systems represents a great challenge due to the heterogeneity of resources in grid environments. In addition, grid resources belong to diverse administrative domains and apply different management policies. Resource Management systems do not focus on the basic functionalities of the resources provided to clients, rather they focus on the way these functionalities are achieved and controlled (De Roure et al., 2003; Foster and Kesselman, 2004). For instance, if there is a computational resource, the resource management system is not concerned with the primary functions of the computation process, rather it focuses on the management of these computation functions, such as when they start computing, how long is the duration of the computing processes and when they terminate. One of the main issues in resource management is the process of scheduling jobs to the appropriate resources so that the jobs finish in an acceptable time and the resources are utilized effectively.

To handle the job scheduling problem on computational grid, heuristics and metaheuristics are employed as scheduling mechanisms. Heuristics and metaheuristics are commonly able to find good but not necessarily efficient and optimal solutions for the job scheduling problem. The remarkable rise in the size of the solution search space motivated researchers to employ nature-inspired metaheuristics to solve computational grid scheduling problems. Nature-inspired metaheuristics have demonstrated an excellent degree of effectiveness and efficiency for handling combinatorial optimization problems (Zang et al., 2010).

The scheduling mechanism or approach consists of three main elements, the scheduling policies, the fitness functions and the scheduling algorithm (Yahyapour, 2002). Generally, the scheduling policy is a set of rules that manage the process of resource allocation for the received jobs. The fitness function is used to evaluate, rank and find out the quality of schedule. The scheduling algorithm is responsible for generating valid schedules for the submitted jobs on the available resources. This thesis will use the words mechanism and approach interchangeably.

1.3.1 Job Scheduling On Computational Grid

Grid resources belong to different administrative domains and apply different management policies. Each grid resource needs to register at the grid information service (GIS). Figure 1.1 illustrates the process of resource management and scheduling in grid computing. The main roles of the grid resource broker are discovering available grid resources and scheduling jobs submitted from grid clients to the available resources. The process of resource allocation starts when the grid clients submit their jobs to the broker. The broker starts the process of discovering the available resources by communicating with GIS. After the discovery of available resources that fulfill the lowest permitted level of the requirements, the broker starts the scheduling process using scheduling mechanisms to map the submitted jobs to the available resources. When scheduling process is completed the broker allocates the submitted jobs to the selected resources.

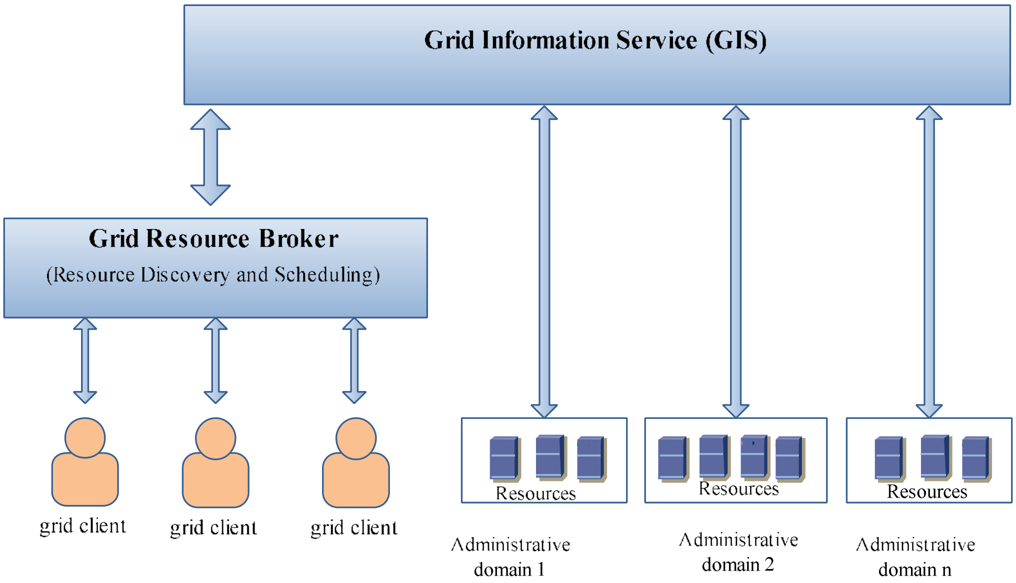

Figure 1.1: Grid Resource Management

Many applications are switching to computational grid to satisfy their needs for high computational power and large data storage. Using grid resources can provide applications with many advantages. Nevertheless, this is feasible only if jobs are scheduled effectively. However, scheduling jobs on such environments represents a great challenge and identified as NP-hard problem of $O(m^n)$ complexity (Izakian et al., 2009; Liu et al., 2010). This implies that running time complexity is exponential to the input size. Therefore, researches are focused on finding polynomial time estimation heuristics for this problem that is fast methods which can build schedules with fitness values near to the optimum values.

There are some features of grid systems that make the job scheduling problem a great challenge. Computational grid has a dynamic structure that is always changing. Resources can leave and join the grid in a dynamic manner. This is because of some resources or connections failure, some users may turn off their computers and some new resources may join the grid. The management of resources belong to different domains with different policies represents one of the greatest difficulties in grid systems. In addition to the dynamic structure of grid systems, the grid resources are heterogeneous and they may vary in the computational speeds, the resource architectures and the operating systems. Furthermore, grid systems consist of resources belonging to different organizations connected via heterogeneous network connections. Dealing with resources with very disparate interconnections network represent additional difficulty to the resource management process. As the computation grid is a coordinated resources sharing from different organization, the existence of local management polices and local schedulers require additional efforts from the global resource managers. Moreover, the jobs submitted to the grid systems are heterogeneous in the length of job and the processing requirements including the operating system and software needed for job execution (Xhafa and Abraham, 2010).

Added to that, computational grid is a large scale distributed systems containing huge number of resources. Therefore, management and scheduling mechanisms for grid resources need to handle the scalability and security issues of the grid resources.

Generally, job scheduling process on computational is classified into two mapping types, static and dynamic scheduling. The online mode and the batch mode are the two modes of the dynamic scheduling.

1.3.1.1 Static Job Scheduling

In static mapping, information regarding all resources as well as the submitted jobs is supposed to be available prior to the start of scheduling process and hence, the scheduling process is done at the compile time (Dong and Akl, 2006). On the other hand, for dynamic mapping, the grid broker performs the job scheduling process during the application execution. Therefore, the scheduling process is done at the real time (Magoules, 2009).

In static scheduling all users submit their jobs to the broker. The broker receives the jobs submitted by users in one list to be scheduled. Larger size of list of jobs to be scheduled results in better schedule. However, so large size of list of jobs to be scheduled makes some resources to become idle before the broker completes the scheduling process. Hence, static scheduling mechanisms need to balance between obtaining efficient schedule by using long list of jobs and not wasting the resources by finishing the scheduling process before resources become idle. After the scheduling process is completed, the broker assigns the jobs to resources statically based on the obtained schedule. The core feature of static scheduling is that, the mapping between jobs and resources is done statically. That is, once the schedule is determined, it remains permanent without change. Static scheduling avoids the overhead caused by the rescheduling process in dynamic scheduling as in the rescheduling process the job is rescheduled and moved from one resource to another more than one time (Dong and Akl, 2006; Xhafa and Abraham, 2010).

1.3.1.2 Dynamic Job Scheduling

Grid environments are considered as dynamic environments since at any time new resources may join and leave the grid system. Furthermore, some existing resources may become unavailable due to network failure or any other reasons (Khan, 2012; Lorpunmanee et al., 2008). The performance of the grid resource changes overtime due to the competition among a lot of elements which share the grid resource. A good scheduler should handle the dynamic issues of grid environments. For instance, if a grid resource becomes unavailable or extremely busy, the scheduler should perform rescheduling process to provide the reliability of the system. Static scheduling is not able to offer acceptable system performance in the case of continuously changing systems, as it assumes that the initial jobs schedules are still optimum after system changes (Dong and Akl, 2006; Xhafa and Abraham, 2010).

In dynamic mapping, the grid broker performs the job scheduling process during the application execution. Therefore, the scheduling process is done at the run time. The dynamic scheduling mechanisms use either online mode or batch mode. In the online mode, jobs are scheduled to resources as soon as they arrive at grid broker (Dong and Akl, 2006; Xhafa and Abraham, 2010). However, in batch mode the jobs are not assigned to resources as they arrive at the grid broker, rather they are gathered into a group that is examined for job allocation at prescheduled times called allocation events (Magoules, 2009).

1.3.2 Heuristics and Metaheuristics for Grid Job Scheduling Problem

Heuristics are mechanisms for deciding which set of actions promises to be the most efficient to achieve certain objectives. Heuristics do not necessarily guarantee to recognize the most efficient solution. However, generally the solutions they find are considered to be sufficient.

Several heuristics mechanisms have been applied to handle the job scheduling problem on computational grid. In First-Come-First-Serve (FCFS) heuristic, the grid broker allocates the jobs in the order of their submission times or arrival times. If there are no available resources or the available resources cannot handle the current job, the grid scheduler waits until the job can be started and the other jobs in the queue are delayed by the scheduler. This scheduling mechanism provides some sort of fairness. However, it may lead to poor scheduling in the case of a job with high resource requirement being submitted to the grid scheduler. This can result in unnecessary loss of time for some resources (A.Iosup et al., 2006; Di Martino and Mililotti, 2004). Another scheduling heuristic is Backfilling an enhanced version of the FCFS mechanism that attempts to avoid the unnecessary loss of time for some resources. In this scheduling mechanism, if a job with high resource requirements is waiting for execution, other jobs can be scheduled and executed under the condition that the waiting long job is not delayed (Di Martino and Mililotti, 2004). Random grid job scheduling heuristic is a non-deterministic job scheduling mechanism in which the next job to be executed is chosen randomly among all the jobs in the waiting queue. No job has preferences; however, the earlier arrived jobs have a higher probability of being executed (Braun et al., 2001).

Metaheuristics are set of algorithmic notions that can be employed to describe heuristics mechanisms appropriate to a broad set of different problems. Metaheuristics can be defined as a general-purpose heuristic mechanism intended to direct an underlying problem-specific heuristic toward promising area of good solutions in the solution search space. In other words, metaheuristics can be defined as a heuristic for the heuristics (Dorigo and Stützle, 2004). The main challenge for optimization mechanisms is to increase the possibility of finding global optimal solutions. Greedy optimization mechanism such as Hill Climbing (HC) and Tabu Search (TS) strive to improve each single step. Greedy methods can find the solution fast. However, greedy optimization mechanisms are often trapped in local optimal solutions (Abraham et al., 2000). HC is a local search optimization mechanism. HC is an iterative technique that begins with a random solution in the search space, and then tries to discover optimized solutions by continuously modifying a single element of the current solution. If the modification generates a better candidate solution, the modification is considered, otherwise the modification is discarded. HC is a local search mechanism which is suitable for finding local optimal solutions and so it is not appropriate in searching for global optimization. Besides, HC optimization mechanism suffers from the plateau problem when the solution search space is flat. In that situation, HC is not capable of finding out which way it should go, or it may choose directions that never lead to the optimal solution. Similar to HC mechanism is TS which is a metaheuristic local search mechanism that can be used for handling optimization problems (Brucker, 2007). TS has been applied to handle job scheduling on computational grid (Abraham et al., 2000). TS has superiority over HC as it has a memory. This memory helps in keeping on with the exploration even if the improving movement is absent. Moreover, the memory prevents the TS scheduling mechanism from getting trapped in a local optimum that has been visited previously. However, TS uses a single search path of solutions and not population search or tree search. In the single search path technique, a set of moves throughout the solution search space are assessed to choose the best candidate solution.

Evolutionary Algorithm (EA) mechanisms, such as Differential Evolution (DE) generate a random initial population of chromosomes. Each chromosome represents a valid solution. After the initial population is generated, the population chromosomes are refined using crossover and mutation operations.

EAs have a limited range of movements, which reduces the likelihood of trapping in local or sub optimal solutions. However, they are slower in finding optimal solutions as a result of the complexity in managing the population movements (Li et al., 2007). GA and DE are used to schedule the jobs on the computational grid (De Falco et al., 2007; Di Martino and Mililotti, 2002; S. Selvi et al., 2011). Evolutionary metaheuristics scheduling mechanisms outperform the grid basic scheduling mechanisms in most cases (Entezari-Maleki and Movaghar, 2011; Izakian et al., 2010). Evolutionary algorithms such GA and DE in some cases are trapped in local optimal and cannot evolve any more. This is because the population diversity is becoming low after some number of iterations so there is no diversity between the population chromosomes which make the crossover and mutation operations are not able to generate improved chromosomes(Kang et al., 2008). To tackle this problem, GA applies a limited range of movements, which decreases the possibility of trapping in sub optimal. However, this makes GA to be slower in finding optimal solutions. Furthermore, evolutionary algorithms may have a memory to store previous status. This memory may help in minimizing the number of individuals close to positions in candidate solutions that have been visited before. However, this may also decrease the search space from convergence since successive generations may die out. The limitations of GA and DE affected the performance of the job scheduling problem on computational grid as GA and DE produced long makespan and flowtime compared to other mechanisms (Abraham et al., 2006; Chen et al., 2006; Kang et al., 2008; Liu et al., 2010; Zhang et al., 2006).

Swarm Intelligence (SI) is a new class of nature-inspired metaheuristics based on population optimizations. The population elements are particles that aim to find the global optimal candidate solution by communicating with other particles and with the environment. In SI such as PSO and ACO, particles do not die; rather, they move throughout the search space themselves. PSO and ACO have been used as scheduling mechanisms to map the jobs to resources on computational grid in several research (Abraham et al., 2006; Dorigo and Stützle, 2004; Liu et al., 2010).

ACO has been applied as an optimization mechanism for scheduling jobs on computational grid (Basu and Mahanti, 2011; Xu et al., 2003; Yan et al., 2005). ACO is an optimization method inspired by the real ants in discovering the shortest path from source to destination. Real ants move randomly searching for food and return back to the nest while dropping pheromone on the path to identify their chosen path to encourage other ants to use (Zang et al., 2010). If other ants use the same path, they will deposit more pheromone and if the path is no longer been used, the pheromone will start to evaporate. The ants always choose the path that has higher pheromone concentration, and then they give feedback by depositing their own pheromone to make other ants use the path.

In view of the fact that the pheromone evaporates over time, the longer the path from source to destination, the faster the pheromone decreases its concentration.

The movement probability from the position i to the position j p_{ij}^{k} is determined by equation (1.2).

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}} \qquad if\ j \in N_i^k \tag{1.2}$$

where η_{ij} is heuristic information for ant k to choose position j from position i, $\tau_{ij}$the pheromone rate in the path ij. Equation (1.2) considers the exploitation of previous and gathered data through the pheromone value and the exploration of new paths through the heuristics information. The value of α and β are between 1 and 0. If $\alpha = 0$, then the path selection decision is then based only on the heuristics information (exploration only). However, if $\beta=0$, then the selection decision will depend only on the pheromone trail (exploitation only). In ACO, although solution search space convergence is guaranteed, however, the time to convergence is uncertain (V. Selvi and Umarani, 2010). Generally due to the limitations of the ACO, scheduling jobs on computational grid using ACO produces good but not optimal schedules in term of makespan time and flowtimes (Hu and Gong, 2009; MadadyarAdeh and Bagherzadeh, 2011; Meihong et al., 2010).

Particle Swarm Optimization (PSO) is one of the swarm intelligence(SI) optimization methods, inspired by social behavior of swarms such as bird flocking or fish schooling (Zhang et al., 2008). Several works have been done to optimize job scheduling on computational grid using PSO (Chen et al., 2006; Izakian et al., 2009; Kang et al., 2008; Liu et al., 2010). In PSO, particles never die. Particles are considered as simple agents that move and interact throughout the search space and record the best solution that they have visited. PSO technique is an adaptive optimization method (Liu et al., 2010; Zhang et al., 2008). In particle swarm optimization, each particle represents a feasible solution in the search space which is a valid schedule of the client's submitted jobs to the available resources. Each particle in PSO has a position vector and velocity. After comparing the fitness of each schedule, the particles move based on their local knowledge which is represented by the particle knowledge and global knowledge which is the knowledge that the particles gain from the swarm.

PSO has a number of disadvantages, for example, PSO slow its convergence speed when it is near the optimal solution. This is because PSO applies the linearly decreasing of inertia weights. Applying the linearly decreasing inertia weights affects the search capabilities at the end of run even if the global search capacity is needed to escape from local optimum in some cases (Shi and Eberhart, 1999). Furthermore, PSO suffer suffers from the partial optimism. This problem affects the PSO speeds and directions (Dian et al., 2011). The disadvantages characteristics of PSO affected the performance of PSO in the process of scheduling jobs on computational grid as the standard PSO produces acceptable but not optimal schedules in terms of makespan and makespan times (Izakian et al., 2009; Kang et al., 2008; S. Selvi et al., 2011).

1.3.3 Minimum Completion Time for Job Scheduling

Minimum Completion Time (MCT) scheduling algorithm assigns each job to the resource with the expected minimum completion time for that job (R. F. Freund et al. 1998). The completion time of a job is calculated by adding the expected time to execute the job to the current resource schedule length. MCT is considered as a successful heuristic as it considers execution times and resource loads (Ritchie and Levine 2004).MCT may result in mapping jobs to resources that do not have the shortest execution time for jobs. The advantages of MCT is to combine the benefits of OLB and MET scheduling mechanisms, at the same time it avoids the situations in which OLB and MET execute poorly (Braun et al. 2001).

1.3.4 Firefly Algorithm (FA)

Firefly algorithm (FA) is a metaheuristic algorithm, inspired by the flashing behavior of fireflies (Yang, 2008). The Firefly Algorithm (FA) is a population-based technique to find the global optimal solution based on swarm intelligence, investigating the foraging behavior of fireflies (Senthilnath et al., 2011). The main function of the firefly's flash is to operate as a signal method to attract other fireflies. The flashing signal by fireflies is to attract mating partners and preys and share food with others (Senthilnath et al., 2011; Yang, 2009, 2010). Firefly Algorithm generates random initial population of feasible candidate solutions. All fireflies of the population are handled in the solution search space with the aim that knowledge is collectively shared among fireflies to guide the search to the best location in the search space. Each particle in the population is a firefly that moves in the multi-dimensional search space with an attractiveness that is dynamically updated based on the knowledge of the firefly and its neighbors.

Firefly Algorithm has introduced by Yang (2008), and it has been utilized in several fields as an optimization methods and the firefly utilization shows promosing results. In the work by Yang (2009), Firefly Algorithm for mulitmodal optimization was formulated. To handle the continuous constraint optimization a new method based on firefly algorithm was introduced by Łukasik and Żak (2009). Flow shop scheduling problem is considered as NP hard problem. A discrete Firefly Algorithm was developed to minimize the makespan time for flow shop scheduling problem (Sayadi et al., 2010). The introduced firefly algorithm for shop scheduling outperformed ACO optimization mechanism.

The previous work in FA indicates that FA has proven to be a good metaheuristics search technique on optimization problems (Apostolopoulos and Vlachos, 2010; Aungkulanon et al., 2011; Chai-ead et al., 2011; dos Santos Coelho et al., 2011; Farahani et al.; Gandomi et al., 2011; Hönig, 2010; Horng and Liou, 2011; JEKLENE, 2011; Łukasik and Żak, 2009; Sayadi et al., 2010; Senthilnath et al., 2011; Tilahun and Ong, 2013). The results in these works showed that FA is promising method that can be used to optimize scheduling jobs problem on computational grid.

1.4 Problem Statement

Due to the characteristics of grid systems, scheduling jobs on computational grid is a challenging problem with exponential running time to the input size. The discussion in the problem background has shown that several heuristics and metaheuristics mechanisms have been introduced to handle the scheduling problem. However, these mechanisms have some limitations that need to be tackled.

The basic scheduling heuristics such as Minimum Execution Time (MET, and First Come First Serve (FCFS) choose the best solution based on single criterion, without taking into account the decision effects on the coming steps. For example, the motivation behind MET is to give each job its best resource. This can cause a severe load imbalance across resources. Likewise, FCFS heuristic may lead to poor scheduling in the case of a job with high resource requirement being submitted to the grid scheduler, which can result in unnecessary loss of time for some resources.

The greedy algorithms such as Tabu Search (TS) and Hill Climbing (HC) use a single search path of solutions and not population search or tree search. This limitation makes TS and HC suitable for scheduling problem with lightweight workload trace in which the search space is not so huge and TS may find the optimal solution easily and hence get good makespan and flowtimes. However, in heavy workload it is difficult for the single path mechanism such as TS to find the optimum solution. Moreover, single path search suffer from plateau and solution search space diversity problems. Due to these limitations TS and HC produce long makespan and flowtimes in the case of heavy load systems.

Evolutionary algorithms such GA and DE in some cases become trapped in local optimal and cannot evolve any more. This is because the population diversity is becoming low after some number of iterations so there is no diversity between the population chromosomes which make the crossover and mutation operations are not able to generate improved chromosomes (Kang et al., 2008). Furthermore, EAs are slower in finding optimal solutions as a result of the complexity in managing the population movements. The limitations of GA and DE affected the performance of the job scheduling problem on computational grid as GA and DE produced long makespan and flowtime compared to other mechanisms (Abraham et al., 2006; Chen et al., 2006; Kang et al., 2008; Liu et al., 2010; Zhang et al., 2006).

Current swarm intelligence mechanisms such as PSO and ACO have a number of issues to be tackled. For example, PSO slow its convergence speed when it is near the optimal solution. Furthermore, PSO suffers from partial optimism. In ACO, although solution search space convergence is guaranteed, however, time to convergence is uncertain. This problem affects the PSO speeds and directions (Dian et al., 2011). Generally due to the limitations of the ACO, scheduling jobs on computational grid using ACO produces good but not optimal schedules in term of makespan time and flowtimes (Hu and Gong, 2009; MadadyarAdeh and Bagherzadeh, 2011; Meihong et al., 2010).The disadvantages characteristics of PSO affected the performance of PSO in the process of scheduling jobs on computational grid as the standard PSO produces acceptable but not optimal schedules in terms of makespan and makespan times (Izakian et al., 2009; Kang et al., 2008; S. Selvi et al., 2011).

To manage resources in computational grid, efficient job scheduling mechanism is required to take the advantages of high computational power provided by the grid. The current scheduling mechanisms suffer from several limitations. These limitations affect the scheduling performance in terms of makespan and flow times. The current metaheuristics produce acceptable but non-optimal schedules. There is an essential need for a new scheduling mechanism that minimizes job makespan and flowtimes and tackles the existing scheduling mechanisms limitation as well.

1.5 Research Questions

This research aims to address the following questions:

How to develop an efficient job scheduling mechanism on computational grid that minimizes the makespan and flowtimes as well as to handle the limitation of the existing scheduling mechanisms?

This main question is supported by the following sub-questions:

1. Can Firefly Algorithm optimization be applied to the grid scheduling problem to produce efficient schedules and address the limitation of the existing scheduling mechanisms as well?
2. What level of optimization is achieved when Firefly Algorithm is used as an optimization method for scheduling jobs on computational grid?
3. How to represent the job scheduling problem on computational grid using Firefly Algorithm optimization?
4. How to develop static job scheduling mechanisms on computational grid that avoids the limitations of the current scheduling mechanisms and minimizes the makespan and flowtimes?
5. Can the integration between firefly optimization mechanism and basic heuristic techniques play any role in the enhancement of job scheduling process?
6. How to develop an adaptive dynamic job scheduling mechanisms on computational grid based on Firefly Algorithm, which handles the dynamic nature of computational grid?

1.6 Research Objectives

In the order to achieve the aim of the study, the objectives of this research are stated as follows:

1. To represent the job scheduling problem on computational grid using Firefly Algorithm optimization and to use this representation to develop, implement and evaluate a static job scheduling approach on computational grid based on Firefly Algorithm optimization to minimize job makespan and flowtime and to handle the current scheduling mechanisms limitations.
2. To enhance the proposed static job scheduling approach to optimize the scheduling process further by integrating MCT and firefly algorithm.
3. To develop, implement and evaluate the proposed adaptive dynamic job scheduling mechanism on computational grid based on firefly algorithm optimization that can handle the dynamic nature of computational grid and minimize the makespan and flowtimes.

The above objectives that an optimized job scheduling mechanisms on computational grid need to be developed in order to minimize the job makespan and flowtimes and to avoid the limitation of the existing scheduling mechanisms.

1.7 Significance of Research and Contributions

This research focuses mainly on developing job scheduling mechanisms using optimization methods to find optimal schedules. In all optimization mechanisms, one of the key issues in designing a successful optimization solution is the representation method which tries to find a suitable mapping between the problem domain and the optimization technique. Inappropriate or wrong representation of the scheduling mechanism may lead to inconsistent and incorrect scheduling results. One of our contributions is the representation and mapping of

Firefly Algorithm to the grid job scheduling problem. This mapping can represent the basis for FA for job scheduling mechanism and hence the researchers can build an enhanced FA job scheduling by exploiting this mapping.

Many applications are switching to computational grid to satisfy their needs for high computational power and large data storage. Using grid resources can provide applications with many advantages. Nevertheless, this is feasible only if resources are scheduled effectively. However, scheduling jobs in such environments represents a great challenge. This research introduces static and enhanced static scheduling mechanisms to improve and optimize the job scheduling process. Utilizing the proposed mechanism by the grid system developers and grid companies will enhance the job management process which will help in motivating more users and applications to convert and accept the grid technologies.

Grid environments are considered as dynamic environments since at any time new resources may join and leave the grid system. Furthermore, some existing resources may become unavailable due to network failure or any other reasons. The performance of the grid resource changes over time due to the competition among a lot of elements that share the grid resource. A good scheduler should handle the dynamic issues of grid environments. If a grid resource becomes unavailable or extremely busy, the scheduler should perform rescheduling process to produce a more reliable system. This research proposed an adaptive dynamic job scheduling mechanism based on firefly algorithm to handle the dynamic nature of the grid .

The current grid project with dynamic features such as Oracle Dynamic Grid Computing and UNCW Grid Computing Project can employ the proposed mechanism to optimize the dynamic job scheduling process and enhance the performance of their systems.

The main contributions of this research can be summarized as follows:

i. A firefly algorithm representation of job scheduling problem on computational grid. This mapping can become the basis for FA as a job scheduling algorithm and therefore the researchers can build enhanced FA job scheduling by utilizing this mapping.

ii. Discrete firefly algorithm for Static job Scheduling (DFAS) and Hybrid Discrete Firefly Algorithm for Static Scheduling (HDFAS) that integrate FA with MCT. The introduced mechanisms enhance the job management process which will help in motivating more users and applications to convert and accept the grid technologies.

iii. Firefly Algorithm for Adaptive Dynamic (FAAD) Jobs Scheduling on computational grid. FAAD can help the dynamic grid projects to optimize the job scheduling process and enhance the performance of their systems. FAAD can help the current grid project with dynamic features such as Oracle grid computing and UNCW Grid Computing Project to optimize the dynamic job scheduling process and to enhance the performance of their systems.

1.8 Research Scope

The scope of the research is limited to the following:

1. This research only focuses on job scheduling on computational grid.
2. This research is based on simulation and real workload data to evaluate the proposed mechanisms. Furthermore, this research employs GridSim as a simulation tool (Buyya and Murshed, 2002) which is a discrete-event grid simulation tool based on java. The objectives of the simulations are, firstly to evaluate the performance for the proposed mechanism under different level of workloads and secondly, to compare the performance of the proposed mechanism with state of the arts job scheduling mechanisms.
3. In this research, the local schedulers of the different grid resources are responsible for scheduling, managing and controlling over the resources at lower-level scheduling instance that are only considered in space

sharing fashion. This work focus on global scheduling done by the grid broker.

4. This research considers independent jobs and job preemptions are not allowed.
5. This research is concerned with scheduling process on grid computing at the resource level.
6. Furthermore, the security issues of grid resources and client's job and the resource failure are not within the scope of this research.
7. Finally, this study does not consider the network details, and internal threads for provider resource.

1.9 Organization of the thesis

This thesis contains seven chapters. The structure of the thesis is grouped into four sections: positioning the research, the framework and methods of the research, the body of the research and the outcomes. The chapters composing each part are depicted in Figure 1.2. This chapter introduces the research works and states the problems and the objectives of the research.

Chapter 2 describes the previous and related works. It starts by highlighting the computational grid importance and the heterogeneity of grid resources. Classification of job scheduling mechanisms is provided in chapter 2 including a critical analysis for the grid job scheduling mechanisms. Furthermore, chapter 2 explores the swarm intelligence mechanisms and their use in the context of job scheduling on computational grid. A detailed description of the standard Firefly Algorithm is illustrated in this chapter including the attractiveness of fireflies and the firefly movements. Moreover, chapter two explains the reasons behind selecting the implementation, verification and validation methods incorporated in this research.

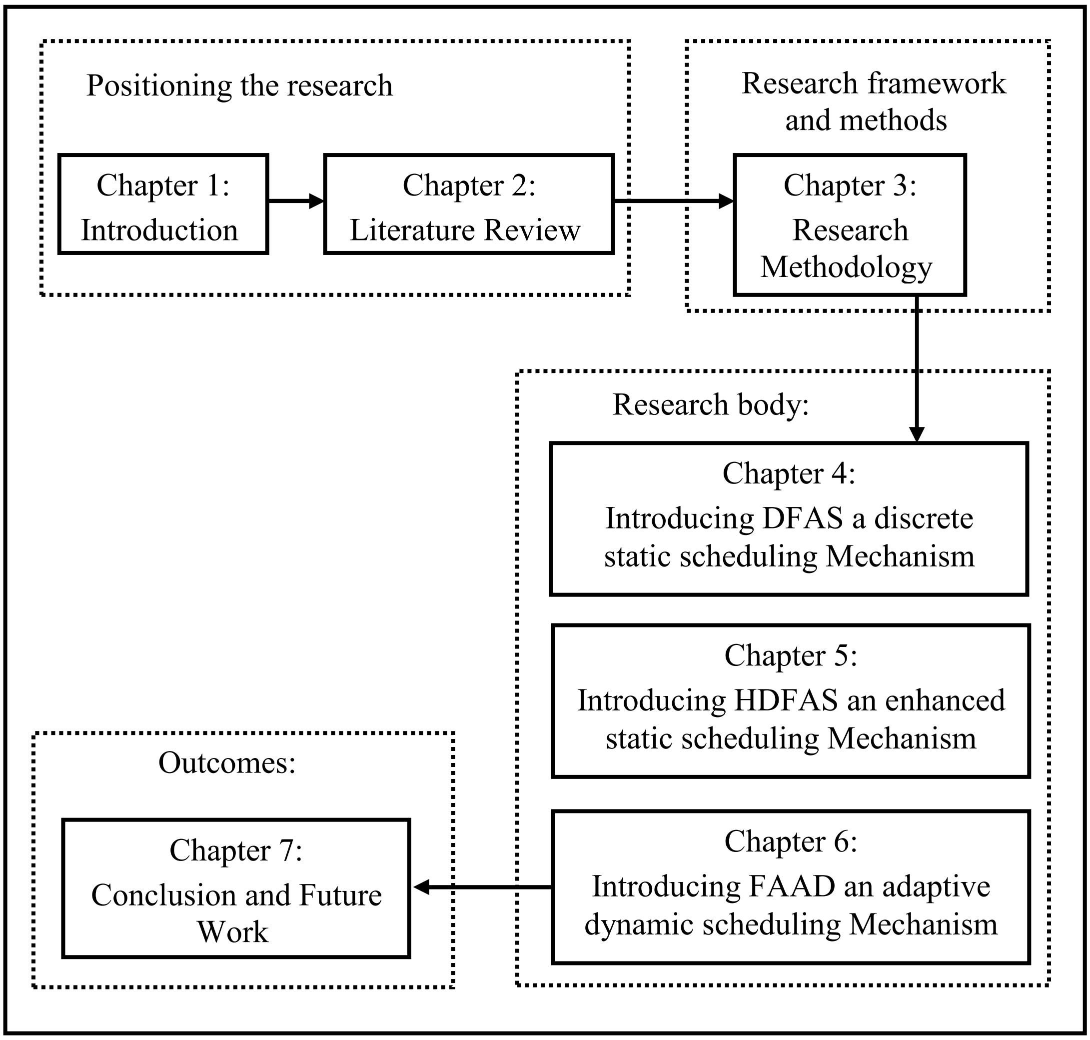

Figure 1.2: Structure of the thesis

Finally the Grid Workload Achieve (GWA) is discussed as a real workload trace and the GridSim simulator as well is described.

Chapter 3 presents the research methodology employed in this study. It starts by formulating the job scheduling problem on computation grid. Then it illustrates the implementation, validation and verification methods used to test the proposed mechanism. The simulation model, the workload data and the evaluation metrics are described as well.

Chapter 4 presents the proposed Discrete Firefly Algorithm for Static job scheduling (DFAS) on computational grid. Furthermore, the detailed architecture, the flowcharts as well as the pseudo codes of the algorithms used are described. The proposed DFAS scheduling mechanism is evaluated using simulation and real workload data.

Chapter 5 illustrates the proposed Hybrid Discrete Firefly Algorithm for Static (HDFAS) scheduling jobs on computational grid by enhancing the DFAS described in chapter 4. The proposed mechanism combines DFAS with Minimum Completion time (MCT) greedy algorithm. Chapter 5 describes the general metaheuristics mechanism for scheduling problems, this followed by an overview of the greedy algorithm and a description of MCT scheduling algorithm. The details of the proposed mechanism are described in chapter 5 as well as the process of performance evaluation. Finally, the experiments and results for HDFAS are illustrated.

Chapter 6 introduces the proposed Firefly Algorithm (FAAD) for Adaptive Dynamic job scheduling on computational grid. The details of the mechanism, the flowcharts as well as the pseudo codes of the algorithms used are described. The proposed FAAD scheduling mechanism is evaluated using simulation and real workload data. Furthermore, the details of the simulation model including its parameter, experimentations design and simulation results are demonstrated in chapter 6.

Chapter 7 summarizes the thesis, presents our conclusions, and indicates several future research directions stemming from this thesis.

REFERENCES

Abdulal, W., Jabas, A., Ramachandram, S., and Al Jadaan, O. (2010), 'Rank based genetic scheduler for grid computing systems', *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on* (IEEE), 644-49.

Abraham, A., Buyya, R., and Nath, B. (2000a), 'Nature’s heuristics for scheduling jobs on computational grids', *The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000)*, 45–52.

Abraham, A., Liu, H., Zhang, W., and Chang, T.G. (2006), 'Scheduling jobs on computational grids using fuzzy particle swarm algorithm', (Springer), 500-07.

Adzigogov, L, Soldatos, J, and Polymenakos, L (2005), 'EMPEROR: An OGSA Grid meta-scheduler based on dynamic resource predictions', *Journal of Grid computing,* 3 (1), 19-37.

Andrieux, A, Czajkowski, K, Dan, A, Keahey, K, Ludwig, H, Pruyne, J, Rofrano, J, Tuecke, S, and Xu, M (2004), 'Web services agreement specification (WS-Agreement)'.

Anjomshoaa, A, Brisard, F, Drescher, M, Fellows, D, Ly, A, McGough, S, Pulsipher, D, and Savva, A (2005), 'Job submission description language (JSDL) specification', *Version,* 1, 15.

Apostolopoulos, T. and Vlachos, A. (2010), 'Application of the firefly algorithm for solving the economic emissions load dispatch problem', *International Journal of Combinatorics,* 2011, 1-23.

Arafah, M. A., Al-Harbi, H. S., and Bakry, S. H. (2007), 'Grid computing: A STOPE view', *International Journal of Network Management,* 17 (4), 295-305.

Arsuaga-Rios, M., Vega-Rodriguez, MA, and Prieto-Castrillo, F. (2011), 'Multi-objective artificial bee colony for scheduling in grid environments', *Swarm Intelligence (SIS), 2011 IEEE Symposium on* (IEEE), 1-7.

Aungkulanon, P., Chai-Ead, N., and Luangpaiboon, P. (2011), 'Simulated manufacturing process improvement via particle swarm optimisation and firefly algorithms', *Proceedings of the International MultiConference of Engineers and Computer Scientists* (2).

Banati, H. and Bajaj, M. (2011), 'Fire Fly Based Feature Selection Approach', *IJCSI International Journal of Computer Science Issues,* 8 (4), 473-80.

Banks, J. (1998), *Handbook of simulation: principles, methodology, advances, applications and practice. ed* (Wiley Online Library).

Banks, Jerry, et al. (2005), *Discrete-event system simulation* (Upper Saddle River :Prentice Hall).

Basu, B. and Mahanti, G.K. (2011), 'Fire Fly and Artificial Bees Colony Algorithm for Synthesis of Scanned and Broadside Linear Array Antenna', *Progress In Electromagnetics Research,* 32, 169-90.

Bell, W.H., Cameron, D.G., Millar, A.P., Capozza, L., Stockinger, K., and Zini, F. (2003), 'Optorsim: A grid simulator for studying dynamic data replication strategies', *International Journal of High Performance Computing Applications,* 17 (4), 403-16.

Berman, F., Wolski, R., Casanova, H., Cirne, W., Dail, H., Faerman, M., Figueira, S., Hayes, J., Obertelli, G., and Schopf, J. (2003), 'Adaptive computing on the grid using AppLeS', *Parallel and Distributed Systems, IEEE Transactions on,* 14 (4), 369-82.

Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B., and Hensgen, D. (2001), 'A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems* 1', *Journal of Parallel and Distributed computing,* 61 (6), 810-37.

Brucker, P. (2007), *Scheduling algorithms* (Springer Verlag).

Buyya, R. and Murshed, M. (2002), 'Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing', *Concurrency and Computation: Practice and Experience,* 14 (13 15), 1175-220.

Capannini, G., Baraglia, R., Puppin, D., Ricci, L., and Pasquali, M. (2007), 'A job scheduling framework for large computing farms', *Proceedings of the 2007 ACM/IEEE conference on Supercomputing* (ACM), 54.

Casanova, H. (2001), 'Simgrid: A toolkit for the simulation of application scheduling', *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on* (IEEE), 430-37.

Chai-ead, N., Aungkulanon, P., and Luangpaiboon, P. (2011), 'Bees and Firefly Algorithms for Noisy Non-Linear Optimization Problems', *Proceedings of the International Multi Conference of Engineering and Computer Scientists* (2).

Chandrasekaran, S., Ponnambalam, SG, Suresh, RK, and Vijayakumar, N. (2006), 'A hybrid discrete particle swarm optimization algorithm to solve flow shop scheduling problems', (IEEE), 1-6.

Chapman, C., Goonatilake, C., Emmerich, W., Farrellee, M., Tannenbaum, T., Livny, M., Calleja, M., and Dove, M. (2005), 'Condor Birdbath-web service interface to Condor'.

Chauhan, S.S. and Joshi, RC (2010), 'A weighted mean time min-min max-min selective scheduling strategy for independent tasks on grid', (IEEE), 4-9.

Chen, T., Zhang, B., Hao, X., and Dai, Y. (2006), 'Task scheduling in grid based on particle swarm optimization', *Parallel and Distributed Computing, 2006. ISPDC'06. The Fifth International Symposium on* (IEEE), 238-45.

Cheon, Y., Yeep, C., and Vela, M. (2011), 'Cleanjava: A formal notation for functional program verification', *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on* (IEEE), 221-26.

Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., and Tuecke, S. (2000), 'The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets', *Journal of network and computer applications,* 23 (3), 187-200.

Chone, J. and Khodabakchian, E. (2006), 'Lightweight directory access protocol workflow management system', (Google Patents).

Cirne, W., Paranhos, D., Costa, L., Santos-Neto, E., Brasileiro, F., Sauvé, J., Silva, F.A.B., Barros, C.O., and Silveira, C. (2003), 'Running bag-of-tasks applications on computational grids: The mygrid approach', *Parallel Processing, 2003. Proceedings. 2003 International Conference on* (IEEE), 407-16.

Clematis, A., Corana, A., D'Agostino, D., Galizia, A., and Quarati, A. (2010), 'Job-resource matchmaking on Grid through two-level benchmarking', *Future Generation Computer Systems,* 26 (8), 1165-79.

Coello, C.A.C., Lamont, G.B., and Van Veldhuizen, D.A. (2007), *Evolutionary algorithms for solving multi-objective problems* (5: Springer-Verlag New York Inc).

Curti, C., Ferrari, T., Gommans, L., Van Oudenaarde, S., Ronchieri, E., Giacomini, F., and Vistoli, C. (2005), 'On advance reservation of heterogeneous network paths', *Future Generation Computer Systems,* 21 (4), 525-38.

De Falco, I., Scafuri, U., Tarantino, E., and Della Cioppa, A. (2007), 'A distributed differential evolution approach for mapping in a grid environment', *Parallel, Distributed and Network-Based Processing, 2007. PDP'07. 15th EUROMICRO International Conference on* (IEEE), 442-49.

De Roure, D, Baker, MA, Jennings, NR, and Shadbolt, NR (2003), 'The evolution of the Grid', *Grid computing: making the global infrastructure a reality,* 13, 14-15.

Delavar, A.G., Nejadkheirallah, M., and Motalleb, M. (2010), 'A new scheduling algorithm for dynamic task and fault tolerant in heterogeneous grid systems using Genetic Algorithm', *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on* (9: IEEE), 408-12.

Di Martino, V. and Mililotti, M. (2002), 'Scheduling in a grid computing environment using genetic algorithms', 297.

Di Martino, V. and Mililotti, M. (2004), 'Sub optimal scheduling in a grid using genetic algorithms', *Parallel Computing,* 30 (5), 553-65.

Dian, P.R., Siti, M.S., and Siti, S.Y. (2011), 'Particle Swarm Optimization: Technique, System and Challenges', *International Journal of Computer Applications,* 14 (1), 19-27.

Dong, F. and Akl, S.G. (2006), 'Scheduling algorithms for grid computing: State of the art and open problems', *School of Computing, Queen's University, Kingston, Ontario*.

Dorigo, M. and Stützle, T. (2004), *Ant colony optimization* (the MIT Press).

dos Santos Coelho, L., de Andrade Bernert, D.L., and Mariani, V.C. (2011), 'A chaotic firefly algorithm applied to reliability-redundancy optimization', *Evolutionary Computation (CEC), 2011 IEEE Congress on* (IEEE), 517-21.

Elmroth, E. and Tordsson, J. (2009), 'A standards-based Grid resource brokering service supporting advance reservations, coallocation, and cross-grid interoperability', *Concurrency Computation Practice and Experience,* 21 (18), 2298-335.

Elmroth, Erik and Tordsson, Johan (2008), 'Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions', *Future Generation Computer Systems,* 24 (6), 585-93.

Entezari-Maleki, R. and Movaghar, A. (2011), 'A genetic algorithm to increase the throughput of the computational grids', *International Journal of Grid and Distributed Computing,* 4 (2).

Ernemann, C., Hamscher, V., Schwiegelshohn, U., Yahyapour, R., and Streit, A. (2002), 'On advantages of grid computing for parallel job scheduling', (Ieee), 39-39.

Falcon, R., Almeida, M., and Nayak, A. (2011), 'Fault identification with binary adaptive fireflies in parallel and distributed systems', *Evolutionary Computation (CEC), 2011 IEEE Congress on* (IEEE), 1359-66.

Farkas, Z. and Kacsuk, P. (2012), 'Evaluation of hierarchical desktop grid scheduling algorithms', *Future Generation Computer Systems,* 28 (6), 871-80.

Feitelson, D., Rudolph, L., and Schwiegelshohn, U. (2005), 'Parallel job scheduling—a status report', *Job Scheduling Strategies for Parallel Processing* (Springer), 1-16.

Feitelson, D.G. and Weil, A.M. (1998), 'Utilization and Predictability in Scheduling the IBM SP2 with Backlling', *IPPS/SPDP 1998* (IEEE Computer Society), 542-46.

Ferreira, L, Organization, International Business Machines Corporation. International Technical Support, and ebrary, Inc (2003), *Introduction to grid computing with globus* (IBM Corp., International Technical Support Organization).

Fidanova, S. and Durchova, M. (2006), 'Ant algorithm for grid scheduling problem', *Large-Scale Scientific Computing*, 405-12.

Foster, I, Kesselman, C, Nick, JM, and Tuecke, S (2003), 'The physiology of the grid', *Grid computing: making the global infrastructure a reality*, 217–50.

Foster, I. and Kesselman, C. (2004), *The grid: blueprint for a new computing infrastructure* (Morgan Kaufmann).

Foster, I., Kesselman, C., and Tuecke, S. (2001), 'The anatomy of the grid: Enabling scalable virtual organizations', *International Journal of High Performance Computing Applications,* 15 (3), 200-22.

Frey, J, Tannenbaum, T, Livny, M, Foster, I, and Tuecke, S (2002), 'Condor-G: A computation management agent for multi-institutional grids', *Cluster Computing,* 5 (3), 237-46.

Frıncu, Marc Eduard (2011), 'Adaptive scheduling for distributed systems, ' PhD Thesis (West University of Timisoara).

Gao, Y., Rong, H., and Huang, J.Z. (2005), 'Adaptive grid job scheduling with genetic algorithms', *Future Generation Computer Systems,* 21 (1), 151-61.

García-Galán, S., Prado, RP, and Muñoz Expósito, JE (2011), 'Fuzzy scheduling with swarm intelligence-based knowledge acquisition for grid computing', *Engineering Applications of Artificial Intelligence*.

Glover, F. (1989), 'Tabu search-part I', *ORSA Journal on computing,* 1 (3), 190-206.

Hamscher, V., Schwiegelshohn, U., Streit, A., and Yahyapour, R. (2000), 'Evaluation of job-scheduling strategies for grid computing', *Grid Computing—GRID 2000*, 191-202.

Harold, R and Loukides, M (2000), *Java network programming* (O'Reilly & Associates, Inc. Sebastopol, CA, USA).

He, X.S., Sun, X.H., and Von Laszewski, G. (2003), 'QoS guided min-min heuristic for grid task scheduling', *Journal of Computer Science and Technology,* 18 (4), 442-51.

Hönig, U. (2010), 'A Firefly Algorithm-based Approach for Scheduling Task Graphs in Homogeneous Systems', (ACTA Press).

Horng, M.H. (2012), 'Vector quantization using the firefly algorithm for image compression', *Expert Systems with Applications*.

Horng, M.H. and Liou, R.J. (2011), 'Multilevel minimum cross entropy threshold selection based on the firefly algorithm', *Expert Systems with Applications,* 38 (12), 14805-11.

Hu, Y. and Gong, B. (2009), 'Multi-objective optimization approaches using a CE-ACO inspired strategy to improve grid jobs scheduling', *ChinaGrid Annual Conference, 2009. ChinaGrid'09. Fourth* (IEEE), 53-58.

Huedo, E, Montero, RS, and Llorente, IM (2009), 'A recursive architecture for hierarchical grid resource management', *Future Generation Computer Systems,* 25 (4), 401-05.

Hwa Min, Lee, Sung Ho, Chin, Jong Hyuk, Lee, Dae Won, Lee, Kwang Sik, Chung, Soon Young, Jung, and Heon Chang, Yu (2004), 'A resource manager for optimal resource selection and fault tolerance service in Grids', *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*, 572-79.

Iosup, H. Li, C. Dumitrescu, L. Wolters, and D.H.J.Epema (2006), 'The Grid Workload Format '.

Iosup, A., Li, H., Jan, M., Anoep, S., Dumitrescu, C., Wolters, L., and Epema, D.H.J. (2008), 'The grid workloads archive', *Future Generation Computer Systems,* 24 (7), 672-86.

Izakian, H., Tork Ladani, B., Zamanifar, K., and Abraham, A. (2009), 'A novel particle swarm optimization approach for grid job scheduling', *Information Systems, Technology and Management*, 100-09.

Izakian, H., Ladani, B.T., Abraham, A., and Snasel, V. (2010), 'A discrete particle swarm optimization approach for grid job scheduling', *International Journal of Innovative Computing, Information and Control,* 6 (9), 4219-33.

Jati, G. and Suyanto (2011), 'Evolutionary discrete firefly algorithm for travelling salesman problem', *Adaptive and Intelligent Systems*, 393-403.

Kandaswamy, G., Mandal, A., and Reed, D.A. (2008), 'Fault tolerance and recovery of scientific workflows on computational grids', *Cluster Computing and the Grid, 2008. CCGRID'08. 8th IEEE International Symposium on* (IEEE), 777-82.

Kang, Q. and He, H. (2011), 'A novel discrete particle swarm optimization algorithm for meta-task assignment in heterogeneous computing systems', *Microprocessors and Microsystems,* 35 (1), 10-17.

Kang, Q., He, H., Wang, H., and Jiang, C. (2008), 'A novel discrete particle swarm optimization algorithm for job scheduling in grids', *Natural Computation, 2008. ICNC'08. Fourth International Conference on* (1: IEEE), 401-05.

Kee, Y.S., Casanova, H., and Chien, A.A. (2004), 'Realistic modeling and svnthesis of resources for computational grids', *Proceedings of the 2004 ACM/IEEE conference on Supercomputing* (IEEE Computer Society), 54.

Kennedy, J. and Eberhart, R. (1995), 'Particle swarm optimization', *Neural Networks, 1995. Proceedings., IEEE International Conference on* (4: IEEE), 1942-48.

Kertész, A. and Kacsuk, P. (2007), 'A taxonomy of grid resource brokers', *Distributed and Parallel Systems*, 201-10.

Khan, S.U. (2012), 'Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment', *Information Sciences*.

Kleijnen, J.P.C. (1995), 'Verification and validation of simulation models', *European Journal of Operational Research,* 82 (1), 145-62.

Klusá ek, D., Matyska, L., and Rudová, H. (2007), 'Alea: grid scheduling simulation environment', (Springer-Verlag), 1029-38.

Krauter, K, Buyya, R, and Maheswaran, M (2002), 'A taxonomy and survey of grid resource management systems for distributed computing', *Software: Practice and Experience,* 32 (2), 135-64.

Ku-Mahamud, K.R. and Nasir, H.J.A. (2010), 'Ant colony algorithm for job scheduling in grid computing', *Mathematical/Analytical Modelling and Computer Simulation (AMS), 2010 Fourth Asia International Conference on* (IEEE), 40-45.

Li, Shujuan, Li, Yan, Liu, Yong, and Xu, Yuefei (2007), 'A GA-based NN approach for makespan estimation', *Applied Mathematics and Computation,* 185 (2), 1003-14.

Lifka, D. (1995), 'The anl/ibm sp scheduling system', *Job Scheduling Strategies for Parallel Processing* (Springer), 295-303.

Liu, H., Abraham, A., and Hassanien, A.E. (2010), 'Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm', *Future Generation Computer Systems,* 26 (8), 1336-43.

Lorpunmanee, S., Sap, M., Noor, M., and Abdullah, A.H. (2006), 'Fuzzy C-Mean And Genetic Algorithms Based Scheduling For Independent Jobs In Computational Grid', *Jurnal Teknologi Maklumat,* 18 (2), 1-13.

Lorpunmanee, S., Sap, M., Noor, M., and Abdullah, A.H. (2008), 'Adaptive intelligence job online scheduling within dynamic grid environment based on gridsim', *Jurnal Teknologi Maklumat,* 20 (3), 173-89.

Lorpunmanee, Siriluck (2010), 'Adaptive Intelligent Grid Scheduling System', *Phd Thesis*.

Łukasik, S. and Żak, S. (2009), 'Firefly algorithm for continuous constrained optimization tasks', *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, 97-106.

Ma, T. and Buyya, R. (2005), 'Critical-path and priority based algorithms for scheduling workflows with parameter sweep tasks on global grids', *Computer Architecture and High Performance Computing, 2005. SBAC-PAD 2005. 17th International Symposium on* (IEEE), 251-58.

MadadyarAdeh, M. and Bagherzadeh, J. (2011), 'An improved ant algorithm for grid scheduling problem using biased initial ants', *Computer Research and Development (ICCRD), 2011 3rd International Conference on* (2: IEEE), 373-78.

Magoules, F. (Frederic) (2009), *Grid resource management : towards virtual and services compliant grid computing*, ed. Chapman & Hall/CRC numerical analysis and scientific computing (Boca Raton :CRC Press).

Maheswaran, M., Ali, S., Siegal, HJ, Hensgen, D., and Freund, R.F. (1999), 'Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems', *Heterogeneous Computing Workshop, 1999.(HCW'99) Proceedings. Eighth* (IEEE), 30-44.

Mathiyalagan, P., Suriya, S., and Sivanandam, SN (2011), 'Hybrid enhanced ant colony algorithm and enhanced bee colony algorithm for grid scheduling', *International Journal of Grid and Utility Computing,* 2 (1), 45-58.

Meihong, W., Wenhua, Z., and Keqing, W. (2010), 'Grid Task Scheduling Based on Advanced No Velocity PSO', *Internet Technology and Applications, 2010 International Conference on* (IEEE), 1-4.

Mostaghim, S., Branke, J., and Schmeck, H. (2007), 'Multi-objective particle swarm optimization on computer grids', *Proceedings of the 9th annual conference on Genetic and evolutionary computation* (ACM), 869-75.

Nassif, L. N., Nogueira, J. M., and De Andrade, F. V. V. (2009), 'Resource selection in grid: A taxonomy and a new system based on decision theory, case-based reasoning, and fine-grain policies', *Concurrency Computation Practice and Experience,* 21 (3), 337-55.

Onwubolu, G. and Davendra, D. (2009), 'Differential Evolution for Permutation—Based Combinatorial Problems', *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, 13-34.

Pham, DT, Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., and Zaidi, M. (2006), 'The bees algorithm–a novel tool for complex optimisation problems', *Proceedings of IPROMS 2006 conference*, 454-61.

Pooranian, Z., Harounabadi, A., Shojafar, M., and Mirabedini, J. (2011), 'Hybrid PSO for Independent Task scheduling in Grid Computing to Decrease Makespan', *Proceedings of International Conference on Future Information Technology*, 327-31.

Pop, F., Dobre, C., Godza, G., and Cristea, V. (2006), 'A simulation model for grid scheduling analysis and optimization', *Parallel Computing in Electrical Engineering, 2006. PAR ELEC 2006. International Symposium on* (IEEE), 133-38.

Price, K.V., Storn, R.M., and Lampinen, J.A. (2005), *Differential evolution: a practical approach to global optimization* (Springer Verlag).

Quan, Liang, Yang, Yang, and Li, Liu (2007), 'A Method with Fault-Tolerance for Resource Reservation in Service-Oriented Grid Environment', *Natural Computation, 2007. ICNC 2007. Third International Conference on* (4), 479-83.

R. F. Freund, Gherrity., M., Ambrosius, S., Camp-bell, M., M. Halderman, D. Hensgen, E. Keith, T. Kidd, M. Kussow, J. D. Lima, F. Mirabile, L. Moore, B., Rust, and H. J. Siegel (1998), 'Scheduling resources in multi-user, heterogeneo', *7th IEEE Heterogeneous Computing Workshop* 184–99.

Reddy, K.H.K. and Roy, D.S. (2012), 'A hierarchical load balancing algorithm for efficient job scheduling in a computational grid testbed', *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on* (IEEE), 363-68.

Ritchie, G. and Levine, J. (2004), 'A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments'.

Ruay-Shiung Chang, Jih-Sheng Chang, and Lin, Po-Sheng (2009), 'An ant algorithm for balanced job scheduling in grids', *Future Generation Computer Systems,* 25, 20–27.

Sacks, G.E. (2010), *Saturated model theory* (World Scientific Pub Co Inc).

Sadasivam, D.G.S. (2009), 'An Efficient Approach to Task Scheduling in Computational Grids', *International Journal of Computer Science and Applications,* 6 (1), 53-69.

Sanyal, S., Jain, A., Das, S.K., and Biswas, R. (2003), 'A hierarchical and distributed approach for mapping large applications to heterogeneous grids using genetic algorithms', *Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on* (IEEE), 496-99.

Sayadi, M.K., Ramezanian, R., and Ghaffari-Nasab, N. (2010), 'A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems', *International Journal of Industrial Engineering Computations,* 1, 1-10.

Schnizler, B (2007), *Resource Allocation in the Grid: A Market Engineering Approach* (Univ.-Verl. Karlsruhe).

Schopf, JM (2003), 'Ten actions when grid scheduling', *International Series in Operations Research and Management Science*, 15-24.

Schwiegelshohn, U., Tchernykh, A., and Yahyapour, R. (2008), 'Online scheduling in grids', *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on* (IEEE), 1-10.

Selvi, S., Manimegalai, D., and Suruliandi, A. (2011), 'Efficient Job Scheduling on Computational Grid with Differential Evolution Algorithm', *International Journal of Computer Theory and Engineering,* 3, 277-81.

Selvi Somasundaram, Thamarai, Amarnath, Balachandar R., Kumar, R., Balakrishnan, P., Rajendar, K., Rajiv, R., Kannan, G., Rajesh Britto, G., Mahendran, E., and Madusudhanan, B. (2010), 'CARE Resource Broker: A framework for scheduling and supporting virtual resource management', *Future Generation Computer Systems,* 26 (3), 337-47.

Selvi, V. and Umarani, R. (2010), 'Comparative analysis of ant colony and particle swarm optimization techniques', *International Journal of Computer Applications IJCA,* 5 (4), 1-6.

Senthilnath, J., Omkar, SN, and Mani, V. (2011), 'Clustering using firefly algorithm: Performance study', *Swarm and Evolutionary Computation.*

Shi, Y. and Eberhart, R.C. (1999), 'Empirical study of particle swarm optimization', *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (3: IEEE).

Storn, R. and Price, K. (1997), 'Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces', *Journal of global optimization,* 11 (4), 341-59.

Suciu, A. and Potolea, R. (2008), 'A taxonomy for grid applications'.

Sulistio, A. and Buyya, R. (2004), 'A grid simulation infrastructure supporting advance reservation', *16th International Conference on Parallel and Distributed Computing and Systems (PDCS 2004)*, 9-11.

Sulistio, A., Poduvaly, G., Buyya, R., and Tham, C.K. (2005), 'Constructing a grid simulation with differentiated network service using GridSim', *6th International Conference on Internet Computing (ICOMP 2005)* (27-30).

Talbi, E.G. and Muntean, T. (1993), 'Hill-climbing, simulated annealing and genetic algorithms: a comparative study and application to the mapping problem', *System Sciences, 1993, Proceeding of the Twenty-Sixth Hawaii International Conference on* (2: IEEE), 565-73.

Talukder, AKM, Kirley, M., and Buyya, R. (2009), 'Multiobjective differential evolution for scheduling workflow applications on global Grids', *Concurrency and Computation: Practice and Experience,* 21 (13), 1742-56.

Tasgetiren, M.F., Sevkli, M., Liang, Y.C., and Gencyilmaz, G. (2004), 'Particle swarm optimization algorithm for single machine total weighted tardiness problem', (2: IEEE), 1412-19 Vol. 2.

Tasgetiren, M.F., Liang, Y.C., Sevkli, M., and Gencyilmaz, G. (2007), 'A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem', *European Journal of Operational Research,* 177 (3), 1930-47.

Tilahun, S.L. and Ong, H.C. (2013), 'Vector optimisation using fuzzy preference in evolutionary strategy based firefly algorithm', *International Journal of Operational Research,* 16 (1), 81-95.

Wang, Q., Gao, Y., and Liu, P. (2006), 'Hill Climbing-Based Decentralized Job Scheduling on Computational Grids', *Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on* (1: IEEE), 705-08.

Xhafa, F. and Abraham, A. (2010), 'Computational models and heuristic methods for Grid scheduling problems', *Future Generation Computer Systems,* 26 (4), 608-21.

Xu, Z., Hou, X., and Sun, J. (2003), 'Ant algorithm-based task scheduling in grid computing', (2: IEEE), 1107-10 vol. 2.

Yahyapour, R. (2002), 'Design and evaluation of job scheduling strategies for grid computing, PhD Thesis', (Universität Dortmund).

Yan-Ping, B., Wei, Z., and Jin-Shou, Y. (2008), 'An improved PSO algorithm and its application to grid scheduling problem', *Computer Science and Computational Technology, 2008. ISCSCT'08. International Symposium on* (1: IEEE), 352-55.

Yan, G. and Li, C. (2011), 'An effective refinement artificial bee colony optimization algorithm based on chaotic search and application for pid control tuning', *J Comput Inf Syst,* 7 (9), 3309-16.

Yan, H., Shen, X.Q., Li, X., and Wu, M.H. (2005), 'An improved ant algorithm for job scheduling in grid computing', *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on* (5: IEEE), 2957-61.

Yang, X.S. (2008), 'Firefly algorithm', *Nature-Inspired Metaheuristic Algorithms*, 79-90.

Yang, X.S. (2009), 'Firefly algorithms for multimodal optimization', *Stochastic Algorithms: Foundations and Applications*, 169-78.

Yang, X.S. (2010a), 'Firefly algorithm, stochastic test functions and design optimisation', *International Journal of Bio-Inspired Computation,* 2 (2), 78-84.

Yang, X.S. (2010b), *Nature-inspired metaheuristic algorithms* (Luniver Press).

Yang, X.S.(2010c), 'Firefly algorithm, Levy flights and global optimization', *Research and Development in Intelligent Systems XXVI*, 209-18.

Yang, X.S. and Deb, S. (2010), 'Eagle strategy using Levy walk and firefly algorithms for stochastic optimization', *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 101-11.

Yi, H. and Bin, G. (2009), 'A distributed cross-entropy ANT algorithm for network-aware grid scheduling', *Pervasive Computing (JCPC), 2009 Joint Conferences on* (IEEE), 253-56.

Yoonsik Cheon and Vela, Melisa (2011), 'A Tutorial on Functional Program Verification'.

Yousif, A., Abdullah, A. H., Latiff, M. S. A., and Abdelaziz, A.A. (2011), 'Scheduling Jobs On Grid Computing Using Firefly Algorithm', *Journal of Theoretical and Applied Information Technology,* 33 (2), 155-64.

Yu, J. and Buyya, R. (2006), 'Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms', *Scientific Programming,* 14 (3), 217-30.

Yuesheng, T., Lu, H., and Jingyu, W. (2009), 'Research of the XML-Based Personal Resource Description in Grid Computing', (2: IEEE), 639-41.

Zang, H., Zhang, S., and Hapeshi, K. (2010), 'A review of nature-inspired algorithms', *Journal of Bionic Engineering,* 7, S232-S37.

Zhang, L., Chen, Y., and Yang, B. (2006), 'Task scheduling based on PSO algorithm in computational grid'.

Zhang, L., Chen, Y., Sun, R., Jing, S., and Yang, B. (2008), 'A task scheduling algorithm based on pso for grid computing', *International Journal of Computational Intelligence Research,* 4 (1), 37-43.

Zhao, S., Lo, V., and Dickey, CG (2005), 'Result verification and trust-based scheduling in peer-to-peer grids', *Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on* (IEEE), 31-38.

Zhong, L., Long, Z.X., Zhang, J., and Song, H.Z. (2011), 'An Efficient Memetic Algorithm for Job Scheduling in Computing Grid', *Information and Automation*, 650-56.

Zhu, Y., Li, M., and Weng, C. (2009), 'Ant Algorithm with Execution Quality Based Prediction in Grid Scheduling', *ChinaGrid Annual Conference, 2009. ChinaGrid'09. Fourth* (IEEE), 77-83.