UNIVERSITI PUTRA MALAYSIA

TOPOLOGY-AWARE HYPERGRAPH BASED APPROACH TO OPTIMIZE SCHEDULING OF PARALLEL APPLICATIONS ONTO DISTRIBUTED PARALLEL ARCHITECTURES

SINA ZANGBARI KOOHI

FSKTM 2020 26

TOPOLOGY-AWARE HYPERGRAPH BASED APPROACH TO OPTIMIZE SCHEDULING OF PARALLEL APPLICATIONS ONTO DISTRIBUTED PARALLEL ARCHITECTURES

By

SINA ZANGBARI KOOHI

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Doctor of Philosophy

July 2020

DEDICATIONS

This thesis is dedicated to

my beloved
Father & Mother
whose affection, love and encouragement made me able to get through this challenge. Thank you for earning an honourable living for us and for supporting me,

my sweet
Sister, Tina
whose pure love and encouragement have enriched my soul and inspired me to pursue this work.

I love you all, and I appreciate your sacrifice, devotion, and everything have done to me.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Doctor of Philosophy

TOPOLOGY-AWARE HYPERGRAPH BASED APPROACH TO OPTIMIZE SCHEDULING OF PARALLEL APPLICATIONS ONTO DISTRIBUTED PARALLEL ARCHITECTURES

By

SINA ZANGBARI KOOHI

July 2020

Chairman : Associate Professor Nor Asilah Wati Abdul Hamid, PhD
Faculty : Computer Science and Information Technology

It has broadly acknowledged that the rapid progression of computer technology has brought dramatic growth in the complexity and scale of systems. These complex systems have designed to solve various types of problems from different areas, resulting in high-demanding Heterogeneous Parallel Applications (HPAs). HPAs use parallel processors and assist in parallel execution of tasks with complex interdependency between data and operations. In such architectures, having less waiting time and less response time are crucial. However, achieving an optimum solution for these two metrics is a trivial task because their efficiency relies on modelling, optimizing, partitioning, and job scheduling methods. In this thesis, an approach to optimize the scheduling of parallel applications over heterogeneous architectures to achieve optimum waiting and response time has proposed. The proposed technique has provided through four fundamental steps, including modelling of parallel applications, meta-heuristic optimization, partitioning, and parallel job scheduling.

The first step lies at the modelling of parallel applications running on heterogeneous parallel computers. Modelling refers to constructing a model to depict the structure of the application with its tasks and describing the interactions between them. The existing modelling approaches capture the processor heterogeneity information in the model. However, the network heterogeneity has not considered xv before, and the crucial data to reflect the network heterogeneity are missing. Consequently, the metrics provided by them does not cover the network heterogeneity. The first contribution of this thesis is to propose a new modelling approach named MEMPHA

that would consider heterogeneity and capture all vital metrics, resulting in more accurate modelling of HPAs. MEMPHA, a hypergraphs-based model, aims to aspire to the challenge by providing topology modelling of the target parallel machine and application modelling of the parallel application, which is hypergraph-based model, to abstract the details of HPAs. To demonstrate the effectiveness of MEMPHA, experiments have performed on a set of benchmark hypergraphs. As a result, when compared with previous modelling approaches, MEMPHA shows promising results in devising a better plan for assignments of tasks to processors, which in turn aims to achieve better performance.

Since scheduling and mapping fall into NP problems, and there is no efficient exact solution for solving scheduling and mapping, the second challenge in HPAs is optimization. Meta-heuristic algorithms have widely used in HPAs due to their global optimization ability. However, the current meta-heuristic algorithms do not ensure an optimum solution within a reasonable time. Hence, there is yet room for improvement. Moreover, evolutionary algorithms are generally limited in their problem-solving abilities. Any optimization algorithm is suitable for only a specific domain of optimization problems. For these reasons, to improve the time and accuracy of the coverage in population-based meta-heuristics and their utilization in HPAs, this thesis presents a novel optimization algorithm called the Raccoon Optimization Algorithm (ROA). Mimicking a raccoon's search behaviour, the ROA concentrates its searches in the solution space of non-linear continuous problems at finding the global optimum with higher accuracy and lower time coverage. To evaluate the capability of ROA at addressing complicated problems, it has subjected to experiment several benchmark functions. The ROA has then compared with nine well-known optimization algorithms. Subsequent results show that the ROA performs at a higher accuracy with lower coverage time.

The core approach in task scheduling is the partitioning of the tasks, which devise their distribution pattern over processors. Tasks partitioning refers to the effort of grouping tasks into several sets. Providing a balanced partitioning with equal weights are widely studied. However, in heterogeneous architecture, process heterogeneity demands partitions with different weights. Thus, an efficient partitioning to find an optimum dividing of a hypergraph into K imbalance partitioning is the third challenge of HPAs. This thesis provides a new topology-aware multi-level hypergraph partitioning schema to tackle this issue. The proposed partitioning scheme has based on a multi-level partitioning approach which consists of three main steps. In the first step, a sequence of coarsening on the hypergraph has applied to achieve a smaller coarsened hypergraph. Then, in the second phase, the coarsened hypergraph is partitioned to obtain the initial partitions. Finally, the initial partitioning is successively un-coarsened and re-refined back to the original hypergraph. These steps have conducted using the MEMPHA model and ROA algorithm to optimize three metrics: execution time, total communication volume, and imbalance ratio (load balancing). To experiment the efficiency of the proposed topology-aware multi-level hypergraph partitioning schema, a set of benchmark hypergraphs have used. The results have compared with other multi-level hypergraph partitioning tools and indicates that the proposed approach achieve optimum partitioning and significantly increase the speed of partitioning.

The final step in scheduling and mapping is the distribution of the jobs. Job distribution (Job scheduling) refers to planning the order and layout of execution for all submitted jobs. An inefficient layout yields to higher waiting and low speed response time. To achieve an optimum waiting and response time this thesis has proposed a new approach utilizing the aforementioned modelling, optimizing and partitioning algorithms. This approach has simulated on Alea v.4, which is a dedicated simulator for simulating exascale parallel scheduling. The results have compared with multiple scheduling methods and indicated that the proposed method achieves substantial performance improvements in terms of reducing the average waiting time and response time of the jobs.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

PENDEKATAN BERASASKAN HIPERGRAF TOPOLOGI-BERSEDIA UNTUK MENGOPTIMUMKAN PENJADUALAN APLIKASI SELARI KE ATAS SENIBINA SELARI TERAGIH

Oleh

SINA ZANGBARI KOOHI

Julai 2020

Pengerusi : Profesor Madya Nor Asilah Wati Abdul Hamid, PhD
Fakulti : Fakulti Sains Komputer dan Teknologi Maklumat

Perkembangan dari segi skala dan kerumitan sistem-sistem terkesan dari perkembangan pesat teknologi pengkomputeran. Sistem-sistem kompleks berskala besar yang direkabentuk sebagai penyelesaian kepada pelbagai masalah telah secara tidak langsung menjurus ke arah tuntutan tinggi Aplikasi Heterogen Selari (HPA). HPA mengguna pakai pemproses selari dan membantu dalam pelaksanaan tugas secara selari dengan kesalinggantungan rumit antara data dan operasi. Namun, pelaksanaan aplikasi perisian dalam rekabentuk sebegini merupakan suatu tugas yang mencabar, terutamanya apabila melibatkan pemetaan dan penjadualan tugas secara selari.

Matlamat tesis ini adalah untuk mengatasi secara optimum, cabaran-cabaran utama penjadualan tugas secara selari dalam domain pengaturcaraan heterogen selari. Cabaran yang terutama wujud dalam usaha pemodelan aplikasi selari yang berjalan di dalam komputer heterogen selari. Dalam fasa pemodelan, metrik-metrik utama seperti saiz komunikasi total, trafik, kesesakkan hubungan dan pengembangan perlu dititikberatkan. Pendekatan pemodelan semasa kini berkemampuan untuk mengambil kira beberapa metrik tersebut, namun begitu, sebuah pendekatan pemodelan yang mengambil kira kesemua metrik secara menyeluruh belum pernah dikaji. Sumbangan pertama tesis ini adalah untuk mencadangkan sebuah pendekatan pemodelan baru yang mampu mengambil kira kesemua metrik yang secara tidak langsung menjurus kepada pemodelan HPA yang lebih tepat. MEMPHA, sebuah model hipergraf, bermatlamat untuk menyahut cabaran tersebut. Beberapa eksperimen telah dijalankan keatas beberapa hipergraf terkemuka sebagai perbandingan dengan MEM-

PHA untuk mendemonstrasikan keberkesanannya. Hasilnya, apabila dibandingkan dengan pendekatan-pendekatan sebelum ini, MEMPHA menunjukkan hasil yang memberangsangkan dalam merancang pelan pembahagian tugas kepada pemproses, dan secara tidak langsung bermatlamat untuk mencapai prestasi yang lebih baik. Algoritma meta-heuristik telah banyak dipergunakan dalam HPA atas sebab kebolehan pengoptimuman globalnya. Namun, algoritma meta-heuristik semasa tidak dapat memastikan penyelesaian optimum dalam kadar masa yang berpatutan. Bersebabkan itu, ruang untuk penambahbaikan masih ada. Selain dari itu, algoritma evolusi secara umumnya terhad dari segi kebolehan untuk menyelesaikan masalah. Sebarang penambahbaikan hanya sesuai untuk penambahbaikan yang khusus. Bersebabkan itu, bagi memperbaiki kadar masa dan ketepatan pengliputan meta-heuristik populasi dan penggunaannya dalam HPA, tesis ini memperkenalkan Pengoptimuman Algoritma Racoon (ROA). Mengikut tingkah laku cari seekor rakun, ROA menumpukan pencariannya dalam ruang penyelesaian masalah berterusan tidak-linear untuk mencari pengoptimuman global yang lebih tepat dan cepat. Untuk menilai kebolehan ROA dalam menghadapi masalah rumit, ianya telah diuji dengan beberapa fungsi terkemuka. Ia juga diperbandingkan dengan 9 algoritma terkemuka lain. Hasil keputusan berterusan menunjukkan ROA melaksanakan tugasnya dengan lebih tepat dan cepat.

MEMPHA bersama dengan ROA telah digunakan untuk menghasilkan sebuah skema pembahagian hipergraf pelbagai peringkat yang peka topologi bagi tujuan menambah baik kecekapan pembahagian tugas HPA. Pembahagian tugas di sini merujuk kepada usaha mengumpul tugasan ke dalam beberapa set yang menjurus kepada pengimbangan bebanan. MEMPHA menyediakan pemodelan hipergraf yang mengasingkan butiran HPA. Ia juga menyediakan sebuah corak topologi mesin selari yang disasarkan. Hasil dari kaedah model pembahagian, impak keatas prestasi pengimbangan bebanan, penjadualan dan pemetaan adalah besar. Dalam hal ini, ROA memainkan peranan yang besar dalam pengoptimuman pembahagian tugas. Pembahagian dalam hipergraf dibahagi kepada 3 peringatkat utama. Di peringkat 1, beberapa urutan pengkasaran keatas hipergraf dijalankan bagi memperoleh hipergraf yang lebih halus. Di peringkat 2 pula, pengkasaran diteruskan lagi bagi memperoleh bahagian-bahagian awal. Akhir sekali, hipergraf diperhalusi berturut-turut bagi mendapatkan kembali hipergraf yang asal. Kesemua peringkat dijalankan berdasarkan model MEMPHA dan algoritma ROA. Bagi mengukur kecekapan skema pembahagian hipergraf pelbagai peringkat yang peka topologi, beberapa aras ukur telah digunakan. Hasil keputusannya diperbandingkan pula dengan skema pembahagian hipergraf yang lain.

Langkah terakhir keseluruhan melibatkan pemetaan. Bagi mencapai pemetaan prestasi tinggi dalam komputer selari, ianya bergantung kepada kecekapan pebahagian tugas pemproses. Algoritma pemetaan memberi tugas kepada pemproses dengan cara supaya kadar masa keseluruhan di minimumkan. Perkataan tugas dalam tesis ini bermaksud unit pengkomputeran aplikasi yang telah dimodelkan dan dibahagikan melalui peringkat yang disebut di atas. Tesis ini menggunakan ALEA v.4 bagi mensimulasikan penjadualan dan pemetaaan persekitaran pengkomputeran selari exaskala. Keputusan kajian telah diperbandingkan dengan berbagai kaedan pemetaan dan menunjukkan kaedah cadangan ini boleh mencapai penambahbaikan prestasi yang lebih besar dalam mengurangkan masa pelaksanaan dan pengunaan tenaga.

ACKNOWLEDGEMENTS

Firstly, I am greatly thankful to God for the well-being and strength that is necessitated for the accomplishment of this journey. Then, I would like to offer my most sincere and deepest thanks to my supervisor, Associate Professor Nor Asilah Wati Abdul Hamid, whose continuous support during this PhD journey is priceless. She patiently indicated the direction to me and gave me motivation and true guides.

I am thankful to my respectful and gentle co-supervisor, Professor Mohamed Othman, for constructive critics and supports.

I would like to express my very sincere gratitude to my co-supervisor, Associate Professor Ibragimov Gafurjan, for his enthusiasm, wide knowledge and friendliness.

I am honoured to take my PhD under the supervision of this committee as I would never imagine finding a better supervisor committee. Thanks for believing in me.

I must offer my very deep gratitude to my generous and kind parents and sister to provide unfailing support throughout my life. Last but not least, I would like to thank Faranak who has been a constant source of motivation, kindness and support.

Finally, I also thank Malaysia and Universiti Putra Malasia for its hospitality, geniality and friendly memories.

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Doctor of Philosophy. The members of the Supervisory Committee were as follows:

Nor Asilah Wati Abdul Hamid, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairperson)

Mohamed Othman, PhD
Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Gafurjan Ibragimov, PhD
Associate Professor
Faculty of Science
Universiti Putra Malaysia
(Member)

____
ZALILAH MOHD SHARIFF, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 08 October 2020

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature:______________________________ Date:____________________________

Name and Matric No: Sina Zangbari Koohi, GS40362

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) are adhered to.

Signature: ________________________
Name of Chairman of Supervisory Committee:
Associate Professor Dr. Nor Asilah Wati Abdul Hamid

Signature: ________________________
Name of Member of Supervisory Committee:
Professor Dr. Mohamed Othman

Signature: ________________________
Name of Member of Supervisory Committee:
Associate Professor Dr. Gafurjan Ibragimov

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

HPA	Heterogeneous Parallel Architectures
TIG	Task Interaction Graph
TPG	Task Precedence Graph
TTIG	Task Temporal Interaction Graph
TTIGHa	Temporal Task Interaction Graph in Heterogeneous Architectures
MPAHA	Model on Parallel Algorithms on Heterogeneous Architectures
ROA	Raccoon Optimization Algorithm
HPC	Heterogeneous Parallel Computers
DAG	Directed Acyclic Graph
EC	Evolutionary Computing
BC	Block of Computation
CT	Computation Time
TCV	Total Communication Volume
TSM	Total number of messages being sent(or received)
MMS	Maximum number of messaged being sent(or received)
TSV	Total volume of messages being sent(or received)
GP	Graph Partitioning
HGP	Hypergraph Partitioning
SGP	Streaming Graph Partitioning
MEMPHA	Model of Exa-scale Message-Passing Programs in Heterogeneous Architectures
TAMLHP	Topology-aware Multilevel Hypergraph Partitioning
IDE	Integrated Development Environment
RZR	Reachable Zone Radius
RZP	Reachable Zone Population
VZR	Visible Zone Radius
VZP	Visible Zone Population
NI	Number of Iterations
MF	Migration Factor

CHAPTER 1

INTRODUCTION

This chapter describes an overview of this thesis. This covers motivation behind this research, problem statements that have been dealt with, research questions that were the guidance of the steps, research scope, objectives, and a summary of contributions of this thesis.

1.1 Motivation

It is broadly acknowledged that the rapid progress of computer technology imposes significant growth in the complexity and scale of systems. These modern complex systems could be exascale supercomputers with hundreds of millions of processors, which are designed to solve numerous types of problems from different fields. Technically, modern systems should execute complex applications arisen in different fields, such as monitoring and controlling critical systems, military, communication, and multimedia. One of the successful paradigms to handle such complexity in the systems is Heterogeneous Parallel Architectures (HPAs). HPAs take advantage of having multiple kinds of processing units with different speeds and capabilities. In contrast to homogeneous architecture, HPAs are more cost-effective. This is because when all processors are identical, the sequential part of the applications occupies one of the processors and results in wasting the real power of the processor and considerably increases the execution time (Amdahl, 2007). Amdahl's Law simply demonstrates this issue (Amdahl, 2007):

> "The serial fraction of processing dominates the execution time for any large parallel ensemble of processors, limiting the advantages of parallel supercomputers."

Menasce and Almedah in (Menascé and Almeida, 1990; Menasce and Almeida, 1991; Menascé and Almeida, 1991) demonstrate the performance-effectiveness of a set of small processes that tightly coupled to a larger processor. Similar methods and their improved versions are presented in the literature (Andrews and Polychronopoulos, 1991; Schneider, 1991).

These works indicate the importance of HPAs for high-demand modern applications. However, running software applications in such architectures is a challenging task, especially in the field of parallel mapping and scheduling. Parallel job scheduling makes decisions on when and where different tasks are going to be executed. Based on this fact, parallel job scheduling divide into dynamic (run time) and static (compile) scheduling. The static task scheduling algorithms are the most important topic to deal with the problem of task allocation at compile time in HPAs. To achieve a high-performance static task scheduling algorithms in HPAs in the sense of lower waiting and response time, taking a few steps are necessary.

The first step to achieve a successful scheduling is modelling. Modelling refers to constructing a model to depict the structure of the application with its tasks and describing the interactions between them. The model utilises various mathematical structures, such as graphs and hyper-graphs to encapsulate this information. The employment of modelling turns to be more complicated when it comes to heterogeneous architectures. These architectures manipulate several types of processing units along with varieties of network connections. To achieve better performance on these structures, in addition to previous factors being used for homogeneous architectures, tasks should be distributed according to the power of processing units and connecting media. The modelling approach should present adequate information to help the scheduler and mapper in devising better execution plans.

Any modelling approach specifies a set of metrics for the mapping function. The metrics are the fundamental characteristics of any modelling approach. They raise the significant differences between modelling methods and assist in optimizing the scheduling patterns. Minimizing or maximizing these metrics helps to achieve optimized scheduling results, and consequently aids to reduce the overall execution time. The metrics are different for each model, depending on the structures it uses.

In the scope of mapping and scheduling in parallel and distributed machines, six different modelling approaches are currently in use: Task Interaction Graph (TIG) (Long and Clarke, 1989a), Task Precedence Graph (TPG) (Hironori and Seinosuke, 1985), Task Temporal Interaction Graph (TTIG) (Roig *et al.*, 2007a), Temporal Task Interaction Graph in Heterogeneous Architectures (TTIGHa) (De Giusti *et al.*, 2007), Model on Parallel Algorithms on Heterogeneous Architectures (MPAHA) (Giusti *et al.*, 2009), and UMPa (Catalyiirek, 2013). The next Chapter will discuss these approaches in detail. However, each of these modelling approaches captures only a few different metrics of the applications. Since optimizing these metrics have a direct impact on the performance of scheduler, this thesis is motivated to propose a new modelling approach and maximize the aforementioned metrics.

Modelling the application and obtaining the tasks and their information holds all the essential information about the application. A partitioning method has to be applied to this hyper-graph to partition the tasks. Apparently, the partitioning method has a significant impact on the performance of scheduling. Due to this, in the next step of achieving a successful HPAs, the modelled hyper-graph partitions to split the jobs to smaller sets of tasks. Both steps of modelling and partitioning should take the data interdependency between the tasks and load balancing (imbalance ration) into account. In the literature, there are plenty of methods proposed, such as Spectral Bipartitioning (Karypis and Kumar, 1998a), Geometric Partitioning (Miller *et al.*, 1991, 1993), and multilevel partitioning. All these approaches have their advantages and disadvantages. Next chapter addresses the partitioning methods in details. However, the multilevel partitioning approach is known as a fast partitioning method compared to the other state-of-the-art partitioning methods (Chen, 1990; Bui and Jones, 1993; Hendrickson and Leland, 1995b). Due to this, the thesis proposes a modified multilevel partitioning method to improve the performance of scheduling.

Previous studies proved the NP-Completeness of graph and hypergraph partitioning (Koivisto *et al.*, 2018; Bui and Jones, 1992). NP refers to Non-deterministic Polynomial problems, which are a set of problems that have no solution in the feasible time. Thus, the output of multilevel partitioning or any other partitioning method is not the exact solution but a good approximation (Williamson and Shmoys, 2011). Multilevel partitioning methods use meta-heuristics (in the initial partitioning and un-coarsening steps) to enhance the result and achieve a more optimized solution (Buluç *et al.*, 2016; Benlic and Hao, 2011). The current meta-heuristic algorithms do not guarantee the most optimum solution within a reasonable time. Hence, this motivate the researchers to to have novel algorithms in this field (Sörensen, 2015). Moreover, this type of algorithms cannot generally solve all sorts of problems, and each algorithm is suitable in particular domains of optimization problems (Dixit *et al.*, 2015). Therefore, this field still needs to have novel algorithms. This thesis presents a new meta-heuristic algorithm named Raccoon Optimization Algorithm (ROA) to have more optimum solutions. Later, this algorithm is used to enhance the results of multilevel hyper-graph partitioning.

The success of static scheduling and mapping in HPAs tightly depends on the performance of each aforementioned steps: modelling, heuristics algorithms and partitioning. If these steps do not perform efficiently, the mapping of the processes quickly become intractable. This raises the need for more reliable approaches and algorithms to achieve a better performance in the static scheduling and mapping in HPAs. Dealing with this need is the major motivation behind this thesis.

1.2 Problem Statement

Static scheduling in HPAs involves matching of tasks with available processing resources. There are various types of processors with different processing power (Processor Heterogeneity). Moreover, there are different types of communication media between processors (Network Heterogeneity). The execution time of processes on different processors and the communication time between them are different. As a consequence of this variety in processors and communication media, managing the heterogeneity of the processors and the network media are the main challenges of the static task scheduling on HPAs. Inability in managing these factors causes longer waiting times to run and longer execution time of the applications (Dongarra and Lastovetsky, 2009). The main issues in static task scheduling are optimizing the waiting time and execution time of the parallel applications on heterogeneous architectures (Dongarra and Lastovetsky, 2009; Catalyiirek, 2013; Deveci, 2015; Deveci *et al.*, 2015b). Hence, this thesis focuses on solving these issues to optimize the static scheduling and mapping methods in HPAs. However, four main problems have to tackled to improve static scheduling in HPAs.

Modelling of the HPAs and parallel applications is the first step of the scheduling. Heterogeneity is one of the main characteristics of HPAs, and the modelling approach should record its relevant information. The existing modelling approaches capture the processor heterogeneity information in the model (Catalyiirek, 2013; Giusti *et al.*, 2009; De Giusti *et al.*, 2007). However, the network heterogeneity, despite its recent demand, is not studied well, and the crucial data to reflect the network heterogeneity are missing. Consequently, the metrics provided by them does not cover the network heterogeneity. Lack of recording network heterogeneity and providing its relevant metrics are the main issues in modelling parallel applications.

Meta-heuristic algorithms have widely used in different steps of scheduling and mapping. Scheduling and mapping fall into NP category of problems, and there is no efficient exact solution to solve them (Du and Leung, 1989). Therefore, meta-heuristics algorithms provide a sufficient approximation of these problems. Current meta-heuristics algorithms need a high number of iterations to provide an accurate solution (Akbaripour and Masehian, 2013; Catalyiirek, 2013), and consequently, they will need a longer execution time. Additionally, these approaches are going to run on the HPAs and shorter execution time (lesser use of the resources) is demanding (Soleymani and Nematzadeh, 2017). In a nutshell, the high number of the iterations to provide accurate solutions, and long execution time is the main issues in this category of the problems.

In task scheduling, the partitioning is the core approach to devise a distribution pattern for tasks over the processors. K-balanced partitioning (providing k partitions with equal weights) is a well-studied topic in the literature. However, in heterogeneous architectures, process heterogeneity demands partitions with different weights. Additionally, the network heterogeneity in HPAs causes a difference in communication times on different mediums (Panda *et al.*, 2018; Catalyiirek, 2013). Using a typical balanced partitioning method will decrease the load balance and increase communication costs (Baruah, 2004; Panda *et al.*, 2018). High imbalance ratio and communication costs are the main concerns in the partitioning of parallel applications for HPAs.

The final step in scheduling and mapping is the distribution of the jobs[1] (job scheduling). Job distribution (Job scheduling) refers to planning the order and layout of execution for all submitted jobs. An inefficient layout yields to higher waiting times (The time each job should wait to access the resources and get executed). Addi-

1. Every job is a parallel application with numerous tasks. The tasks inside a job are communicating and are dependant. However, jobs are independent units and do not communicate with each other.

tionally, inefficient task scheduling causes longer job execution times and higher response time[2] (Hung *et al.*, 2015; Dakkak *et al.*, 2016). Therefore, high waiting time and response time are the main issues on job scheduling in HPAs. Optimizing these factors ends in better resource utilization and lower power consumption.

This research presents a methodology for static scheduling and mapping in heterogeneous parallel architectures to overcome these issues. It helps parallel development in these architectures to run faster and utilize fewer resources. Overcoming these issues contributes to an optimized execution of the parallel application on heterogeneous parallel architectures.

1.3 Research Questions

The following list summarizes the main challenges of the presented research, in the form of several research questions. The research questions are used as a guideline to achieve optimum job scheduling and mapping.

- What are the necessary rules to model a parallel application that accommodates its network heterogeneity and cover fundamental metrics of network heterogeneity?
- What the extensions needed in meta-heuristics algorithms to improve its accuracy and execution time and reduce the required number of iterations?
- How to partition a parallel application model to decrease its imbalance ratio (increase its load balance) and reduce communication costs?
- How to minimize the waiting time and response time of the jobs submitted to the distributed parallel machine, and manage the resources efficiently?

1.4 Research Objectives

The main research objective of this thesis is to focus on providing a technique to support parallel computing in the heterogeneous parallel architectures. The list of sub-objectives of this research is as below.

- To propose a modelling approach to model tasks in parallel applications with their network heterogeneity characteristics.

2. The response time for any job is the sum of its waiting time and execution time.

- To propose a new meta-heuristic algorithm to enhance the accuracy, the number of required iterations, and execution time of the optimization process.
- To propose a task partitioning method to decrease imbalance ratio[3] and decrease communication costs between processors.
- To propose a job scheduling method to minimize the waiting time and response time of the jobs.

1.5 List of Contributions

This thesis provides a set of methods to support and improve the statistic scheduling, and mapping of the parallel applications to run on heterogeneous parallel architectures.

Chapter 4 presents the first contribution of this thesis, which is a new modelling technique to model parallel applications in HPAs. This chapter involves following sub-contributions:

- Modelling parallel applications using hypergraphs
- Abstracting network heterogeneity in model
- Recognizing all the essential metrics during modelling
- Recording the most useful details of the tasks and their communications

Chapter 5 offers a novel meta-heuristic algorithm called Raccoon Optimization Algorithm (ROA). The ROA inspires from the rummaging behaviours of real raccoons for food. Later, in the next chapter, it will be used to enhance the multi-level partitioning method. The sub-contributions of this algorithm are:

- Reducing the risk of getting stuck in local optima
- Saving time through exploration
- Reducing the complexity of heuristic algorithms

Chapter 6 provides a topology-aware multi-level hypergraph partitioning schema for task partitioning. Then, the model hypergraph, obtained from Chapter 4, split up

3. increase load balancing

using this schema along with taking advantages of ROA. The method is elaborated, formalized, and evaluated using real-world examples.

- Partitioning task hypergraph based on the target machine topology
- Reducing the imbalance ratio (Increasing load balancing)
- Reducing the execution time
- Reducing the total communication cost

Chapter 7 introduces a method based on previous modelling, metaheuristics algorithm and partitioning schema to enhance parallel computing in HPAs in terms of resource allocation, waiting time and response time. This chapter covers the fourth and last contribution of this thesis. The sub-contributions of this chapter are:

- Reducing the waiting time of the jobs in the schedule
- Reducing the response time of the jobs in the schedule
- Optimizing the resource management of the parallel systems

1.6 Research Scope

The presented thesis focuses on providing a technique to improve the scheduling and mapping of jobs in high-performance heterogeneous parallel and distributed computers. This improvement has done through the proposition of a modelling approach, a meta-heuristic algorithm, a multilevel partitioning schema, and a scheduling and mapping algorithm. The proposed modelling approach, named MEMPHA, has based on hypergraphs, to consider all fundamental metrics of the parallel application and distributed machine. The meta-heuristic algorithm, called ROA, is an optimization method which improves the convergence and accuracy of the optimization. The partitioning schema named TAMLHP is a topology-aware multilevel method, which employs the benefits of the modelling and ROA. It reduces the imbalance ratio and total communication costs of partitioning the parallel application's model. Based on the proposed modelling approach, ROA meta-heuristic, and partitioning schema, a job scheduling and mapping approach, named ROA-CONS, has presented. ROA-CONS has used to optimize the waiting and response time of the jobs.

1.7 Thesis Structure

This thesis is prepared according to the thesis preparation guidelines of "University Putra Malaysia". The author has tried to present the contents and information of this research in precise details. The final version of this thesis contains eight chapters. These chapters are organized as below.

Chapter 1 is the introduction that introduces the motivation behind this research. Moreover, it elaborates the problem statement and research objectives. The research questions, scope and contributions are also summarised in this chapter.

Chapter 2 is related to the background and literature on multiple aspects of modelling heterogeneous parallel architecture, hyper-graph partitioning, meta-heuristic algorithms, and job scheduling and mapping. The primary references for this chapter are resource materials such as journals, conference proceedings, seminars, thesis, books, and online resources.

Chapter 3 presents the designed research methodology which has been utilized in conducting this thesis.

Chapter 4 formally defines hyper-graphs and modelling methods. Later, it exhibits the proposed modelling schema. The applicability of the presented model is evaluated, and results are compared to current modelling approaches.

Chapter 5 proposes a new optimization algorithm named ROA (Raccoon Optimization Algorithm). The efficiency and quality of the proposed ROA are tested using different benchmark functions with different properties. The results obtained from the ROA have compared with those of well-known meta-heuristic algorithms.

Chapter 6 presents the successful partitioning schema of model hyper-graph. This hyper-graph has obtained as the result of the previous chapter. This chapter also shows the applicability of the proposed partitioning schema in a real world data sets.

Chapter 7, based on all the steps of modelling, partitioning and mapping, presents an optimized job scheduling method. Then, Alea 4 job scheduling simulator has been utilized to measure the performance of the proposed framework compared to the current schedulers.

Chapter 8 presents the conclusion, contributions and limitations of the research and indicates potential areas for future studies.

REFERENCES

2007, hMETIS - Hypergraph & Circuit Partitioning, http://glaros.dtc.umn.edu /gkhome/metis/hmetis.

2012, Software package and libraries for sequential and parallel graph partitioning, static mapping and clustering, sequential mesh and hypergraph partitioning, and sequential and parallel sparse matrix block ordering, https://www.labri.fr/perso/pelegrin/scotch/.

2013, METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering, http://glaros.dtc.umn.edu/gkhome/metis/metis/overview.

2013, ParMETIS - Parallel Graph Partitioning and Fill-reducing Matrix Ordering, http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview.

2018, Apache NetBeans 10.0, https://netbeans.apache.org/download/archive/ index.html.

2019, MetaCentrum National Grid Infrastructure, https://www.metacentrum.cz.

Abualigah, L. M., Khader, A. T. and Hanandeh, E. S. 2018. Hybrid clustering analysis using improved krill herd algorithm. *Applied Intelligence* 48 (11): 4047–4071.

Agarwal, K. P. 1991, Geometric Partitioning and Its Applications, Tech. rep., Durham, NC, USA.

Agarwal, S., Branson, K. and Belongie, S. 2006. Higher order learning with graphs. In *Proceedings of the 23rd international conference on Machine learning*, 17–24. ACM.

Agrawal, P. K., Pandit, M. and Dubey, H. M. 2016. Improved Krill Herd Algorithm with neighborhood distance concept for optimization. *International Journal of Intelligent Systems and Applications* 8 (11): 34.

Ahmad, I. and Kwok, Y.-K. 1998. On exploiting task duplication in parallel program scheduling. *IEEE Transactions on Parallel and Distributed Systems* 9 (9): 872–892.

Ahmad, I. and Kwok, Y.-K. K. Y.-K. 1994. A new approach to scheduling parallel programs using task duplication. In *1994 Internatonal Conference on Parallel Processing Vol. 2*, 47–51. IEEE.

Akbar, M. F., Munir, E. U., Rafique, M. M., Malik, Z., Khan, S. U. and Yang, L. T. 2016. List-based task scheduling for cloud computing. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 652–659. IEEE.

Akbaripour, H. and Masehian, E. 2013. Efficient and robust parameter tuning for heuristic algorithms. *International Journal of Industrial Engineering and Production Research* 24 (2): 143–150.

Alami, J., El Imrani, A. and Bouroumi, A. 2007. A multipopulation cultural algorithm using fuzzy clustering. *Applied Soft Computing* 7 (2): 506–519.

Ali, N., Othman, M. A., Husain, M. N. and Misran, M. H. 2014. A review of firefly algorithm. *ARPN Journal of Engineering and Applied Sciences* 9 (10): 1732–1736.

Amdahl, G. M. 2007. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, Reprinted from the AFIPS Conference Proceedings, Vol. 30 (Atlantic City, NJ, Apr. 18–20), AFIPS Press, Reston, Va., 1967, pp. 483–485, when Dr. Amdahl was at International Business Machines Corporation, Sunnyvale, California. *IEEE Solid-State Circuits Society Newsletter* 12 (3): 19–20.

Andreev, K. and Racke, H. 2006. Balanced graph partitioning. *Theory of Computing Systems* 39 (6): 929–939.

Andrews, J. B. and Polychronopoulos, C. D. 1991. An analytical approach to performance/cost modeling of parallel computers. *Journal of Parallel and Distributed Computing* 12 (4): 343–356.

Arabnejad, H. 2013. List based task scheduling algorithms on heterogeneous systems-an overview. In *Doctoral Symposium in Informatics Engineering*, 93.

Arafeh, B., Day, K. and Touzene, A. 2008. A multilevel partitioning approach for efficient tasks allocation in heterogeneous distributed systems. *Journal of Systems Architecture* 54 (5): 530–548.

Arivudainambi, D. and Rekha, D. 2013. Memetic algorithm for minimum energy broadcast problem in wireless ad hoc networks. *Swarm and Evolutionary Computation* 12: 57–64.

Arjmand, M., Sheikhi Azqandi, M. and Delavar, M. 2018. Hybrid improved dolphin echolocation and ant colony optimization for optimal discrete sizing of truss structures. *Journal of Rehabilitation in Civil Engineering* 6 (1): 70–87.

Armbruster, M. 2007. *Branch-and-Cut for a Semi definite Relaxation of Large-scale Minimum Bisection Problems*. PhD thesis, Chemnitz University of Technology.

Arora, S., Rao, S. and Vazirani, U. 2009. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)* 56 (2): 5.

Ashlock, D. 2006. *Evolutionary computation for modeling and optimization*. Springer Science & Business Media.

Astaneh, B. 2015. *Algorithms for Graph Partitioning*. PhD thesis, Imperial College London.

Atashpaz-Gargari, E. and Lucas, C. 2007. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *2007 IEEE congress on evolutionary computation*, 4661–4667. IEEE.

Aykanat, C., Cambazoglu, B. B., Findik, F. and Kurc, T. 2007. Adaptive decomposition and remapping algorithms for object-space-parallel direct volume rendering of unstructured grids. *Journal of Parallel and Distributed Computing* 67 (1): 77–99.

Azmi, Z. R. M., Bakar, K. A., Abdullah, A. H., Shamsir, M. S. and Manan, W. N. W. 2011. Performance comparison of priority rule scheduling algorithms using different inter arrival time jobs in grid environment. *International Journal of Grid and Distributed Computing* 4 (3): 61–70.

Azrag, M., Kadir, T. A., Odili, J. and Essam, M. 2017. A global african buffalo optimization. *Int. J. Softw. Eng. Comput. Syst* 3 (3): 138–145.

Bäck, T., Fogel, D. B. and Michalewicz, Z. 2000a. *Evolutionary computation 1: Basic algorithms and operators*. , vol. 1. CRC Press.

Bäck, T., Fogel, D. B. and Michalewicz, Z. 2000b. *Evolutionary computation 2: Advanced Algorithms and Operators*. , vol. 1. CRC Press.

Bader, D. A., Meyerhenke, H., Sanders, P., Schulz, C., Kappes, A. and Wagner, D. 2014. Benchmarking for graph clustering and partitioning. *Encyclopedia of Social Network Analysis and Mining* 73–82.

Balci, M. A. and Akgüller, Ö. 2014. Average weakly hyperedge domination number for a hypergraph and actor-network application. *International Journal of Modeling and Optimization* 4 (5): 346.

Banharnsakun, A., Achalakul, T. and Sirinaovakul, B. 2011. The best-so-far selection in artificial bee colony algorithm. *Applied Soft Computing* 11 (2): 2888–2901.

Baños, R., Gil, C., Ortega, J. and Montoya, F. G. 2003. Multilevel heuristic algorithm for graph partitioning. In *Workshops on Applications of Evolutionary Computation*, 143–153. Springer.

Bansal, J. C., Sharma, H., Jadon, S. S. and Clerc, M. 2014. Spider monkey optimization algorithm for numerical optimization. *Memetic computing* 6 (1): 31–47.

Barnard, S. T. and Simon, H. D. 1994. Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and experience* 6 (2): 101–117.

Baruah, S. K. 2004. Task Partitioning Upon Heterogeneous Multiprocessor Platforms. In *IEEE real-time and embedded technology and applications symposium*, 536–543. Citeseer.

Basak, A., Pal, S., Das, S., Abraham, A. and Snasel, V. 2010. A modified invasive weed optimization algorithm for time-modulated linear antenna array synthesis. In *IEEE Congress on Evolutionary Computation*, 1–8. IEEE.

Behnamian, J. 2016. Graph colouring-based algorithm to parallel jobs scheduling on parallel factories. *International Journal of Computer Integrated Manufacturing* 29 (6): 622–635.

Benlic, U. and Hao, J.-K. 2011. A multilevel memetic approach for improving graph k-partitions. *IEEE Transactions on Evolutionary Computation* 15 (5): 624–642.

Binitha, S., Sathya, S. S. *et al.* 2012. A survey of bio inspired optimization algorithms. *International journal of soft computing and engineering* 2 (2): 137–151.

Boeres, C., Chochia, G. and Thanisch, P. 1995. On the scope of applicability of the ETF algorithm. In *Parallel Algorithms for Irregularly Structured Problems* (eds. A. Ferreira and J. Rolim), 159–164. Berlin, Heidelberg: Springer Berlin Heidelberg.

Boeres, C., Rebello, V. E. *et al.* 2004a. A cluster-based strategy for scheduling task on heterogeneous processors. In *16th symposium on computer architecture and high performance computing*, 214–221. IEEE.

Boeres, C., Rebello, V. E. *et al.* 2004b. A cluster-based strategy for scheduling task on heterogeneous processors. In *16th symposium on computer architecture and high performance computing*, 214–221. IEEE.

Boman, E. G., Catalyurek, U. V., Chevalier, C. and Devine, K. D. 2012. The Zoltan and Isorropia Parallel Toolkits for Combinatorial Scientific Computing: Partitioning, Ordering, and Coloring. *Scientific Programming* 20 (2): 129–150.

Boman, E. G., Devine, K. D. and Rajamanickam, S. 2013. Scalable matrix computations on large scale-free graphs using 2D graph partitioning. In *SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 1–12. IEEE.

Bontoux, B. and Feillet, D. 2008. Ant Colony Optimization for the Traveling Purchaser Problem. *Comput. Oper. Res.* 35 (2): 628–637.

Bonyadi, M. R. and Michalewicz, Z. 2017. Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. *Evolutionary Computation* 25 (1): 1–54.

Boppana, R. B. 1987. Eigenvalues and graph bisection: An average-case analysis. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, 280–285. IEEE.

Brandes, U., Gaertler, M. and Wagner, D. 2003. Experiments on graph clustering algorithms. In *European Symposium on Algorithms*, 568–579. Springer.

Bremermann, H. J. 1962. Optimization through evolution and recombination. *Self-organizing systems* 93: 106.

Briggs, W. L., McCormick, S. F. *et al.* 2000. *A multigrid tutorial.* , vol. 72. Siam.

Brunetta, L., Conforti, M. and Rinaldi, G. 1997. A branch-and-cut algorithm for the equicut problem. *Mathematical Programming* 78 (2): 243–263.

Bui, T. N. and Jones, C. 1992. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters* 42 (3): 153–159.

Bui, T. N. and Jones, C. 1993, A heuristic for reducing fill-in in sparse matrix factorization, Tech. rep., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (United States).

Buluç, A. and Boman, E. G. 2008. Towards scalable parallel hypergraph partitioning. *CSRI Summer Proceedings 2008* 109.

Buluç, A., Meyerhenke, H., Safro, I., Sanders, P. and Schulz, C. 2016, In Algorithm Engineering, In *Algorithm Engineering*, 117–158, Springer, 117–158.

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): 1175–1220.

Catalyiirek, U. V. 2013. UMPa: A multi-Objective, multi-level partitioner for communication minimization Umit V. Catalyiirek, Mehmet Deveci, Kamer Kaya, and Bora Ucar. *Graph Partitioning and Graph Clustering* 588: 53.

Çatalyürek, Ü. and Aykanat, C. 2011. Patoh (partitioning tool for hypergraphs). *Encyclopedia of Parallel Computing* 1479–1487.

Chapin, S. J., Cirne, W., Feitelson, D. G., Jones, J. P., Leutenegger, S. T., Schwiegelshohn, U., Smith, W. and Talby, D. 1999. Benchmarks and standards for the evaluation of parallel job schedulers. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 67–90. Springer.

Chelladurai, S. R. 2017. Gridsim: a flexible simulator for grid integration study. Master's thesis, Universidad of Northern British Colombia.

Chen, J. and Safro, I. 2011. Algebraic distance on graphs. *SIAM Journal on Scientific Computing* 33 (6): 3468–3490.

Chen, J.-l. 1990. *A Parallel Graph Partitioning Algorithm for Message-passing Multiprocessors and Its Application to Choleskey Factorization*. PhD thesis, University of California, Santa Barbara.

Chevalier, C. and Pellegrini, F. 2008. PT-Scotch: A tool for efficient parallel graph ordering. *Parallel computing* 34 (6-8): 318–331.

Chingchit, S., Kumar, M. and Bhuyan, L. N. 1999. A flexible clustering and scheduling scheme for efficient parallel computation. In *Proceedings 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing. IPPS/SPDP 1999*, 500–505.

Chingchit, S., Kumar, M. and Bhuyan, L. N. 1999. A flexible clustering and scheduling scheme for efficient parallel computation. In *Proceedings 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing. IPPS/SPDP 1999*, 500–505. IEEE.

Cho, H., Olivera, F. and Guikema, S. D. 2008. A derivation of the number of minima of the Griewank function. *Applied Mathematics and Computation* 204 (2): 694–701.

Chrobak, M., Feige, U., Hajiaghayi, M. T., Khanna, S., Li, F. and Naor, S. 2017. A greedy approximation algorithm for minimum-gap scheduling. *Journal of Scheduling* 20 (3): 279–292.

Cirou, B. and Jeannot, E. 2001. Triplet: a clustering scheduling algorithm for heterogeneous systems. In *Proceedings International Conference on Parallel Processing Workshops*, 231–236. IEEE.

COBHAM, A. 1964. The intrinsic computational difficulty of functions. In *Proc. 1964 Congress for Logic, Methodology, and the Philosophy of Science*, 24–30. North-Holland.

Culler, D. E., Patterson, D., Sahay, A., Santos, E. E., Schauser, K. E., Subramonian, R., von Eicken, T. *et al.* 1996. A practical model of parallel computation. *Communications of the ACM* 39 (11): 78–86.

Dakkak, O., Nor, S. A. and Arif, S. 2016. Scheduling through backfilling technique for HPC applications in grid computing environment. In *2016 IEEE Conference on Open Systems (ICOS)*, 30–35. IEEE.

Darwin, C. 1909. The origin of species. *PF Collier & son New York* 95–96.

De Giusti, L., Chichizola, F., Naiouf, M. and De Giusti, A. 2008a. Mapping tasks to processors in heterogeneous multiprocessor architectures: The MATEHa Algorithm. In *2008 International Conference of the Chilean Computer Science Society*, 85–91. IEEE.

De Giusti, L., Chichizola, F., Naiouf, M., De Giusti, A. and Luque, E. 2010. Automatic mapping tasks to cores-evaluating amtha algorithm in multicore architectures. *International Journal of Computer Science Issues* 7 (1).

De Giusti, L. C., Chichizola, F., Naiouf, M. and De Giusti, A. E. 2008b. Robustness analysis for the method of assignment MATEHa. *Journal of Computer Science & Technology* 8.

De Giusti, L. C., Chichizola, F., Naiouf, M., Ripoll, A. and De Giusti, A. E. 2007. A model for the automatic mapping of tasks to processors in heterogeneous multicluster architectures. *Journal of Computer Science & Technology* 7.

De Giusti, L. C., Naiouf, M., Chichizola, F., Luque Fadón, E. and De Giusti, A. E. 2009. Dynamic scheduling in heterogeneous multiprocessor architectures. In *XV Congreso Argentino de Ciencias de la Computación*.

Dehaene, S. 2011. *The Number Sense: How the Mind Creates Mathematics*.

Delling, D., Goldberg, A. V., Razenshteyn, I. and Werneck, R. F. 2011. Graph partitioning with natural cuts. In *2011 IEEE International Parallel & Distributed Processing Symposium*, 1135–1146. IEEE.

Delling, D., Goldberg, A. V., Razenshteyn, I. and Werneck, R. F. 2012. Exact combinatorial branch-and-bound for graph bisection. In *2012 Proceedings of the Fourteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, 30–44. SIAM.

Delling, D. and Werneck, R. F. 2012. Better bounds for graph bisection. In *European Symposium on Algorithms*, 407–418. Springer.

Deveci, M. 2015. *Load-Balancing and task mapping for exascale systems*. PhD thesis, The Ohio State University.

Deveci, M., Kaya, K., Uçar, B. and Çatalyürek, Ü. V. 2015a. Fast and high quality topology-aware task mapping. In *2015 IEEE International Parallel and Distributed Processing Symposium*, 197–206. IEEE.

Deveci, M., Kaya, K., Uçar, B. and Çatalyürek, Ü. V. 2015b. Hypergraph partitioning for multiple communication cost metrics: Model and methods. *Journal of Parallel and Distributed Computing* 77: 69–83.

Devi, R., Barlaskar, E., Devi, O., Medhi, S. and Shimray, R. 2014. Survey on evolutionary computation tech techniques and its application in different fields. *International Journal on Information Theory (IJIT)* 3 (3): 73–82.

Devine, K. D., Boman, E. G., Heaphy, R. T., Bisseling, R. H. and Catalyurek, U. V. 2006. Parallel hypergraph partitioning for scientific computing. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, 10–pp. IEEE.

Devine, K. D., Boman, E. G., Riesen, L. A., Catalyurek, U. V. and Chevalier, C. 2009. Getting started with zoltan: A short tutorial. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Diekmann, R., Preis, R., Schlimbach, F. and Walshaw, C. 2000. Shape-optimized mesh partitioning and load balancing for parallel adaptive FEM. *Parallel Computing* 26 (12): 1555–1581.

Dixit, M., Upadhyay, N. and Silakari, S. 2015. An exhaustive survey on nature inspired optimization algorithms. *International Journal of Software Engineering and Its Applications* 9 (4): 91–104.

Dogan, A. and Ozguner, R. 2002. LDBS: a duplication based scheduling algorithm for heterogeneous computing systems. In *Proceedings International Conference on Parallel Processing*, 352–359.

Donath, W. E. and Hoffman, A. J. 1972. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. *IBM Technical Disclosure Bulletin* 15 (3): 938–944.

Donath, W. E. and Hoffman, A. J. 2003, In Selected Papers Of Alan J Hoffman: With Commentary, In *Selected Papers Of Alan J Hoffman: With Commentary*, 437–442, World Scientific, 437–442.

Dongarra, J. and Lastovetsky, A. L. 2009. *High performance heterogeneous computing.* , vol. 78. John Wiley & Sons.

Dorigo, M. 1992. *Optimization, learning and natural algorithms*. PhD thesis, Politecnico di Milano, Italy.

Du, J. and Leung, J. Y.-T. 1989. Complexity of scheduling parallel task systems. *SIAM Journal on Discrete Mathematics* 2 (4): 473–487.

Eberhart, R. C., Kennedy, J. *et al.* 1995. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, 39–43. New York, NY.

Echbarthi, G. 2017. *Big Graph Processing: Partitioning and Aggregated Querying*. PhD thesis, Lyon.

El-Ashmawi, W. H. 2018. An Improved African Buffalo Optimization Algorithm for Collaborative Team Formation in Social Network. *International Journal of Information Technology and Computer Science* 5: 16–29.

El-Rewini, H. and Lewis, T. G. 1990. Scheduling parallel program tasks onto arbitrary target machines. *Journal of parallel and Distributed Computing* 9 (2): 138–153.

El-Rewini, H., Lewis, T. G. and Ali, H. H. 1994. *Task Scheduling in Parallel and Distributed Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Eles, P., Peng, Z., Kuchcinski, K. and Doboli, A. 1996. Hardware/software partitioning with iterative improvement heuristics. In *Proceedings of 9th International Symposium on Systems Synthesis*, 71–76. IEEE.

Elio, R., Hoover, J., Nikolaidis, I., Salavatipour, M., Stewart, L. and Wong, K. 2011, About computing science research methodology.

Ewees, A. A., Abd Elaziz, M., Al-Qaness, M. A., Khalil, H. A. and Kim, S. 2020. Improved Artificial Bee Colony Using Sine-Cosine Algorithm for Multi-Level Thresholding Image Segmentation. *IEEE Access* 8: 26304–26315.

Faiz, A., Arief, U. M. *et al.* 2018. An efficient meta-heuristic algorithm for solving capacitated vehicle routing problem. *International Journal of Advances in Intelligent Informatics* 4 (3): 212–225.

Farhat, C., Lanteri, S. and Simon, H. D. 1995. TOP/DOMDEC—a software tool for mesh partitioning and parallel processing. *Computing Systems in Engineering* 6 (1): 13–26.

Farhat, C. and Lesoinne, M. 1993. Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics. *International Journal for Numerical Methods in Engineering* 36 (5): 745–764.

Fattahi, E., Bidar, M. and Kanan, H. R. 2014. Fuzzy krill herd optimization algorithm. In *2014 First International Conference on Networks & Soft Computing (ICNSC2014)*, 423–426. IEEE.

Feitelson, D. G. 2005. Experimental analysis of the root causes of performance evaluation results: a backfilling case study. *IEEE Transactions on Parallel and Distributed Systems* 16 (2): 175–182.

Feitelson, D. G., Rudolph, L., Schwiegelshohn, U., Sevcik, K. C. and Wong, P. 1997. Theory and practice in parallel job scheduling. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 1–34. Springer.

Feitelson, D. G., Tsafrir, D. and Krakov, D. 2014. Experience with using the parallel workloads archive. *Journal of Parallel and Distributed Computing* 74 (10): 2967–2982.

Fernandez Musoles, C., Coca, D. and Richmond, P. 2019. Communication sparsity in distributed Spiking Neural Network Simulations to improve scalability. *Frontiers in neuroinformatics* 13: 19.

Ferreira, C. E., Martin, A., de Souza, C. C., Weismantel, R. and Wolsey, L. A. 1998. The node capacitated graph partitioning problem: a computational study. *Mathematical programming* 81 (2): 229–256.

Fiduccia, C. M. and Mattheyses, R. M. 1982. A linear-time heuristic for improving network partitions. In *19th Design Automation Conference*, 175–181. IEEE.

Fiedler, M. 1975. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal* 25 (4): 619–633.

Fogel, D. B. 1997. Evolutionary algorithms in theory and practice. *Complexity* 2 (4): 26–27.

Fogel, D. B. 1998. *Evolutionary computation: the fossil record*. Wiley-IEEE Press.

Fogel, L. J., Owens, A. J. and Walsh, M. J. 1966. *Artificial intelligence through simulated evolution*. J. Wiley & sons.

Fomin, F. V. and Kratsch, D. 2010. *Exact exponential algorithms*. Springer Science & Business Media.

Foster, I. and Kesselman, C. 2003. *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier.

Frachtenberg, E. and Feitelson, D. G. 2005. Pitfalls in parallel job scheduling evaluation. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 257–282. Springer.

Frankl, P. 2017. On the maximum number of edges in a hypergraph with given matching number. *Discrete Applied Mathematics* 216: 562–581.

Galinier, P., Boujbel, Z. and Fernandes, M. C. 2011. An efficient memetic algorithm for the graph partitioning problem. *Annals of Operations Research* 191 (1): 1–22.

Gamukama, E. A. and Popov, O. 2008. The Level of Scientific Methods Use in Computing Research Programs. In *31st International Convention on Information and Communication Technology*, 176–183.

Gandomi, A. H. and Alavi, A. H. 2012. Krill herd: a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation* 17 (12): 4831–4845.

Geem, Z. W., Kim, J. H. and Loganathan, G. 2001. A new heuristic optimization algorithm: harmony search. *Simulation* 76 (2): 60–68.

Gholizadeh, S. and Poorhoseini, H. 2015. Optimum design of steel frame structures by a modified dolphin echolocation algorithm. *Struct Eng Mech* 55 (3): 535–554.

Giusti, L. D., Luque, E., Chichizola, F., Naiouf, M. and Giusti, A. D. 2009. AMTHA: An Algorithm for Automatically Mapping Tasks to Processors in Heterogeneous Multiprocessor Architectures. In *2009 WRI World Congress on Computer Science and Information Engineering*, 481–485.

Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & operations research* 13 (5): 533–549.

Gómez-Martín, C., Vega-Rodríguez, M. A. and González-Sánchez, J.-L. 2016. Fattened backfilling: An improved strategy for job scheduling in parallel systems. *Journal of Parallel and Distributed Computing* 97: 69–77.

Gortázar, F. and Gallego, M. 2018. A Simple Path Towards Testing Cloud Applications. In *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 28–29.

Griewank, A. O. 1981. Generalized descent for global optimization. *Journal of optimization theory and applications* 34 (1): 11–39.

Groves, T., Knockel, J. and Schulte, E. 2009. Bfs vs. cfs scheduler comparison. *The University of New Mexico* 11.

Guirado, F., Roig, C. and Ripoll, A. 2013. Enhancing throughput for streaming applications running on cluster systems. *Journal of Parallel and Distributed Computing* 73 (8): 1092–1105.

Guo, Y., Hong, S., Chafi, H., Iosup, A. and Epema, D. 2017. Modeling, analysis, and experimental comparison of streaming graph-partitioning policies. *Journal of Parallel and Distributed Computing* 108: 106–121.

Gupta, A., Karypis, G. and Kumar, V. 1997. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Transactions on Parallel and Distributed systems* 8 (5): 502–520.

Györgyi, P. and Kis, T. 2017. Approximation schemes for parallel machine scheduling with non-renewable resources. *European Journal of Operational Research* 258 (1): 113 – 123.

Hager, W. W., Phan, D. T. and Zhang, H. 2013a. An exact algorithm for graph partitioning. *Mathematical Programming* 137 (1-2): 531–556.

Hager, W. W., Phan, D. T. and Zhang, H. 2013b. An exact algorithm for graph partitioning. *Mathematical Programming* 137 (1): 531–556.

Hansen, N., Auger, A., Finck, S. and Ros, R. 2010. Real-parameter black-box optimization benchmarking BBOB-2010: Experimental setup. *INRIA, Tech. Rep. RR-7215* 1–19.

Hassani, H. 2017. Research methods in computer science: The challenges and issues. *arXiv preprint arXiv:1703.04080* 1–16.

Haupt, R. L. and Haupt, S. E. 2004. *Practical genetic algorithms*. John Wiley & Sons.

Henderson, R. L. 1995. Job scheduling under the portable batch system. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 279–294. Springer.

Hendrickson, B. 2010, Graph Partitioning, Tech. Rep. SAND2010-5486J, University of New Mexico, Sandia National Laboratories, Department of Computer Science.

Hendrickson, B. and Kolda, T. G. 2000. Graph partitioning models for parallel computing. *Parallel computing* 26 (12): 1519–1534.

Hendrickson, B. and Leland, R. 1993, The Chaco users guide. Version 1.0, Tech. rep., Sandia National Labs., Albuquerque, NM (United States).

Hendrickson, B. and Leland, R. 1995a. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM Journal on Scientific Computing* 16 (2): 452–469.

Hendrickson, B. and Leland, R. W. 1995b. A Multi-Level Algorithm For Partitioning Graphs. *SC* 95 (28): 1–14.

Hironori, K. and Seinosuke, N. 1985. Practical multiprocessor scheduling algorithms for efficient parallel processing. *Systems and Computers in Japan* 16 (2): 11–19.

Hofmann, E., Maucher, D., Kotula, M. and Kreienbrink, O. 2013. *Performance Measurement and Incentive Systems in Purchasing: more than just savings*. , vol. 3. Springer Science & Business Media.

Holland, J. H. 1973. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing* 2 (2): 88–105.

Horng, M.-H. 2011. Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. *Expert Systems with Applications* 38 (11): 13785–13791.

Hovestadt, M., Kao, O., Keller, A. and Streit, A. 2003. Scheduling in HPC resource management systems: Queuing vs. planning. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 1–20. Springer.

Huang, L., Ding, S., Yu, S., Wang, J. and Lu, K. 2016. Chaos-enhanced Cuckoo search optimization algorithms for global optimization. *Applied Mathematical Modelling* 40 (5-6): 3860–3875.

Hui, C.-C. and Chanson, S. T. 1997. Allocating task interaction graphs to processors in heterogeneous networks. *IEEE Transactions on Parallel and Distributed Systems* 8 (9): 908–925.

Hung, C.-C., Golubchik, L. and Yu, M. 2015. Scheduling jobs across geo-distributed datacenters. In *Proceedings of the Sixth ACM Symposium on Cloud Computing*, 111–124.

Igiri, C. P., Singh, Y. and Bhargava, D. 2018. An improved African buffalo optimization algorithm using chaotic map and chaotic-levy flight. *International Journal of Engineering & Technology* 7 (4): 4570–4576.

Ihler, E., Wagner, D. and Wagner, F. 1993. Modeling hypergraphs by graphs with the same mincut properties. *Information Processing Letters* 45 (4): 171–175.

Ijaz, S. and Munir, E. U. 2019. MOPT: list-based heuristic for scheduling workflows in cloud environment. *The Journal of Supercomputing* 75 (7): 3740–3768.

Iverson, M. A., Özgüner, F. and Follen, G. J. 1995. Parallelizing Existing Applications in a Distributed Heterogeneous Environment. In *4TH HETEROGENEOUS COMPUTING WORKSHOP (HCW '95*, 93–100.

Jánošíková, L. and Haviar, M. 2016. Imperialist competitive algorithm in combinatorial optimization 1–8.

Jedari, B. and Dehghan, M. 2009, In Computer and Information Science 2009, In *Computer and Information Science 2009*, 249–261, Springer, 249–261.

Jiang, W. 2018. Highly efficient distributed hypergraph analysis: real-time partitioning and quantized learning. Master's thesis, University of Melbourne.

Josiński, H., Kostrzewa, D., Michalczuk, A. and Świtoński, A. 2014. The expanded invasive weed optimization metaheuristic for solving continuous and discrete optimization problems. *The Scientific World Journal* 2014.

Kaboli, S. H. A., Selvaraj, J. and Rahim, N. 2016. Long-term electric energy consumption forecasting via artificial cooperative search algorithm. *Energy* 115: 857–871.

Kaboli, S. H. A., Selvaraj, J. and Rahim, N. 2017. Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems. *Journal of Computational Science* 19: 31–42.

Kalayci, T. E. and Battiti, R. 2018. A reactive self-tuning scheme for multilevel graph partitioning. *Applied Mathematics and Computation* 318: 227–244.

Kalinov, A. and Klimov, S. 2005. Optimal Mapping of a Parallel Application Processes onto Heterogeneous Platform. In *19th IEEE International Parallel and Distributed Processing Symposium*, 123b–123b.

Kanemitsu, H., Hanada, M. and Nakazato, H. 2016. Clustering-based task scheduling in a large number of heterogeneous processors. *IEEE Transactions on Parallel and Distributed Systems* 27 (11): 3144–3157.

Kang, Q., He, H. and Song, H. 2011. Task assignment in heterogeneous computing systems using an effective iterated greedy algorithm. *Journal of Systems and Software* 84 (6): 985–992.

Kang, Q., He, H. and Wei, J. 2013. An effective iterated greedy algorithm for reliability-oriented task allocation in distributed computing systems. *Journal of parallel and distributed computing* 73 (8): 1106–1115.

Karaboga, D. 2005, An idea based on honey bee swarm for numerical optimization, Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Karaboga, D., Gorkemli, B., Ozturk, C. and Karaboga, N. 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review* 42 (1): 21–57.

Karypis, G. 2003, In Multilevel Optimization in VLSICAD, In *Multilevel Optimization in VLSICAD*, 125–154, Springer, 125–154.

Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S. 1999. Multilevel hypergraph partitioning: applications in VLSI domain. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 7 (1): 69–79.

Karypis, G. and Kumar, V. 1995a. Analysis of multilevel graph partitioning. In *Supercomputing'95: Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, 29–29. IEEE.

Karypis, G. and Kumar, V. 1995b. A high performance sparse Cholesky factorization algorithm for scalable parallel computers. In *Proceedings Frontiers' 95. The Fifth Symposium on the Frontiers of Massively Parallel Computation*, 140–147. IEEE.

Karypis, G. and Kumar, V. 1998a. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* 20 (1): 359–392.

Karypis, G. and Kumar, V. 1998b. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing* 48 (1): 96–129.

Karypis, G. and Kumar, V. 2000. Multilevel k-way hypergraph partitioning. *VLSI design* 11 (3): 285–300.

Kaveh, A. and Farhoudi, N. 2013. A new optimization method: Dolphin echolocation. *Advances in Engineering Software* 59: 53–70.

Kaveh, A. and Mahjoubi, S. 2018. Lion pride optimization algorithm: A metaheuristic method for global optimization problems. *Scientia Iranica. Transaction B, Mechanical Engineering* 25 (6): 3113–3132.

Kawamoto, T. and Kabashima, Y. 2015. Limitations in the spectral method for graph partitioning: Detectability threshold and localization of eigenvectors. *Physical Review E* 91 (6): 062803.

Keleher, P. J., Zotkin, D. and Perkovic, D. 2000. Attacking the bottlenecks of backfilling schedulers. *Cluster Computing* 3 (4): 245–254.

Kernighan, B. W. and Lin, S. 1970. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal* 49 (2): 291–307.

Kiran, M. S., Hakli, H., Gunduz, M. and Uguz, H. 2015. Artificial bee colony algorithm with variable search strategy for continuous optimization. *Information Sciences* 300: 140–157.

Kirkpatrick, S., Vecchi, M. P. *et al.* 1983. Optimization by simmulated annealing. *science* 220 (4598): 671–680.

Kleban, S. D. and Clearwater, S. H. 2003. Fair share on high performance computing systems: What does fair really mean? In *CCGrid 2003. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proceedings.*, 146–153. IEEE.

Klusáček, D. and Rudová, H. 2010. Alea 2: job scheduling simulator. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, 61. ICST (Institute for Computer Sciences, Social Informatics and Telecommunications Engineering).

Klusáček, D. and Rudová, H. 2011. EFFICIENT GRID SCHEDULING THROUGH THE INCREMENTAL SCHEDULE-BASED APPROACH. *Computational Intelligence* 27 (1): 4–22.

Klusácek, D. and Rudová, H. 2012. Performance and fairness for users in parallel job scheduling. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 235–252. Springer.

Klusáček, D., Rudová, H., Baraglia, R., Pasquali, M. and Capannini, G. 2008. Comparison of multi-criteria scheduling techniques. In *Grid Computing*, 173–184. Springer.

Klusáček, D., Tóth, Š. and Podolníková, G. 2016. Complex job scheduling simulations with Alea 4. In *Proceedings of the 9th EAI International Conference on Simulation Tools and Techniques*, 124–129. ICST (Institute for Computer Sciences, Social Informatics and Telecommunications Engineering).

Knop, D. and Kouteckỳ, M. 2018. Scheduling meets n-fold integer programming. *Journal of Scheduling* 21 (5): 493–503.

Koivisto, M., Laakkonen, P. and Lauri, J. 2018. NP-completeness results for partitioning a graph into total dominating sets. *Theoretical Computer Science* 22–31.

Kostrzewa, D. and Josinski, H. 2013. Using the Expanded IWO Algorithm to Solve the Traveling Salesman Problem. In *ICAART (2)*, 451–456.

Kruatrachue, B. and Lewis, T. 1987. *Duplication scheduling heuristics (dsh): A new precedence task scheduler for parallel processor systems*. PhD thesis, Oregon State University, Corvallis, OR.

Kumar, V., Grama, A., Anshul, G. and Karypis, G. 1994. Introduction to parallel computing: design and analysis of algorithms. *Benjamin/Cummings Publishing Company* 18: 82–109.

Kwok, Y.-K. and Ahmad, I. 1999. Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys (CSUR)* 31 (4): 406–471.

LaSalle, D., Patwary, M. M. A., Satish, N., Sundaram, N., Dubey, P. and Karypis, G. 2015. Improving graph partitioning for modern graphs and architectures. In *Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms*, 14. ACM.

Latorre-Biel, J.-I., Jiménez-Macías, E. and Perez-Parte, M. 2018. Simulation Model of a Tomato Processing Facility for Sustainable Design and Operation. *IFAC-PapersOnLine* 51 (11): 471–477.

Lee, K. Y. and El-Sharkawi, M. A. 2008. *Modern heuristic optimization techniques: theory and applications to power systems*. , vol. 39. John Wiley & Sons.

Leopold, C. 2001. *Parallel and Distributed Computing: A survey of Models, Paradigms and approaches*. John Wiley & Sons, Inc.

Lewis, T. and El-Rewini, H. 1998. Distributed and parallel computing. *Greenwich: Manning* .

Li, B. and Zhao, D. 2007. Performance impact of advance reservations from the grid on backfill algorithms. In *Sixth International Conference on Grid and Cooperative Computing (GCC 2007)*, 456–461. IEEE.

Li, S., Kang, L. and Zhao, X.-M. 2014. A survey on evolutionary algorithm based hybrid intelligence in bioinformatics. *BioMed research international* 2014.

Li, X., Zhang, S., Huang, R., Huang, B., Xu, C. and Kuang, B. 2018. Structured modeling of heterogeneous CAM model based on process knowledge graph. *The International Journal of Advanced Manufacturing Technology* 96 (9-12): 4173–4193.

Lifka, D. A. 1995. The anl/ibm sp scheduling system. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 295–303. Springer.

Lifka, D. A. 1998. *An extensible job scheduling system for massively paralell processor architectures*. Illinois Institute of Technology.

Lin, Q., Zhu, Q., Huang, P., Chen, J., Ming, Z. and Yu, J. 2015. A novel hybrid multi-objective immune algorithm with adaptive differential evolution. *Computers & Operations Research* 62: 95–111.

Lisser, A. and Rendl, F. 2003. Graph partitioning using linear and semidefinite programming. *Mathematical Programming* 95 (1): 91–101.

Liu, X., Shi, H., Lu, Q. and Mao, Z. 2007. Visual task-driven based on task precedence graph for collaborative design. In *2007 11th International Conference on Computer Supported Cooperative Work in Design*, 246–251. IEEE.

Locatelli, M. 2003. A note on the Griewank test function. *Journal of global optimization* 25 (2): 169–174.

Long, D. L. and Clarke, L. A. 1989a. Task Interaction Graphs for Concurrency Analysis. In *Proceedings of the 11th International Conference on Software Engineering*, 44–52. New York, NY, USA: ACM.

Long, D. L. and Clarke, L. A. 1989b. Task interaction graphs for concurrency analysis. In *Proceedings of the 11th international conference on Software engineering*, 44–52.

Lovász, L. *et al.* 1993. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty* 2 (1): 1–46.

Mahale, R. A. and Chavan, S. 2012. A survey: evolutionary and swarm based bio-inspired optimization algorithms. *International Journal of Scientific and Research Publications* 2 (12): 1–6.

Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M. N., Salcedo-Sanz, S. and Geem, Z. W. 2013. A survey on applications of the harmony search algorithm. *Engineering Applications of Artificial Intelligence* 26 (8): 1818–1831.

Marinakis, Y. 2009. Heuristic and metaheuristic algorithms for the traveling salesman problem. *Encyclopedia of Optimization* 1498–1506.

Mark Elbroch, K. R. 2011. *Behavior of North American Mammals*. Houghton Mifflin Harcourt.

Martí, R., Gallego, M., Duarte, A. and Pardo, E. G. 2013. Heuristics and metaheuristics for the maximum diversity problem. *Journal of Heuristics* 19 (4): 591–615.

Mehrabian, A. R. and Lucas, C. 2006. A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics* 1 (4): 355–366.

Menascé, D. and Almeida, V. 1990. Cost-performance Analysis of Heterogeneity in Supercomputer Architectures. In *Proceedings of the 1990 ACM/IEEE Conference on Supercomputing*, 169–177. Los Alamitos, CA, USA: IEEE Computer Society Press.

Menascé, D. and Almeida, V. 1991. Heterogeneity in high performance computing. In *Proceedings of the 2 Symposium on High Performance Computing*.

Menasce, D. and Almeida, V. 1991. Heterogeneous Supercomputing: Why is it Cost-Effective? *Supercomputing review* 4 (8): 39–41.

Meng, X., Liu, Y., Gao, X. and Zhang, H. 2014. A new bio-inspired algorithm: chicken swarm optimization. In *International conference in swarm intelligence*, 86–94. Springer.

Meyerhenke, H., Monien, B. and Schamberger, S. 2009. Graph partitioning and disturbed diffusion. *Parallel Computing* 35 (10-11): 544–569.

Miller, G., Teng, S., Thurston, W. and Vavasis, S. 1993, Automatic mesh partitioning, in "Sparse Matrix Computations: Graph Theory Issues and Algorithms," An IMA Workshop Volume,(A. George, JR Gilbert, and JW-H. Liu, Eds.).

Miller, G. L., Teng, S.-H. and Vavasis, S. A. 1991. A unified geometric approach to graph separators. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, 538–547. IEEE.

Mirjalili, S. and Gandomi, A. H. 2017. Chaotic gravitational constants for the gravitational search algorithm. *Applied soft computing* 53: 407–419.

Mirjalili, S., Mirjalili, S. M. and Lewis, A. 2014. Grey wolf optimizer. *Advances in engineering software* 69: 46–61.

Mittal, N., Singh, U. and Sohi, B. S. 2016. Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing* 2016.

Mladenović, N. and Hansen, P. 1997. Variable neighborhood search. *Computers & operations research* 24 (11): 1097–1100.

Molga, M. and Smutnicki, C. 2005. Test functions for optimization needs. *Test functions for optimization needs* .

Mondal, R. K., Nandi, E. and Sarddar, D. 2015. Load balancing scheduling with shortest load first. *International Journal of Grid and Distributed Computing* 8 (4): 171–178.

Morrison, D. R., Jacobson, S. H., Sauppe, J. J. and Sewell, E. C. 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization* 19: 79–102.

Mozaffari, A. and Ebrahimnejad, M. 2014. The great salmon run metaheuristic for robust shape and size design of truss structures with dynamic constraints. *International Journal of Applied Metaheuristic Computing (IJAMC)* 5 (2): 54–79.

Mozaffari, A. and Fathi, A. 2012. Identifying the behaviour of laser solid freeform fabrication system using aggregated neural network and the great salmon run optimisation algorithm. *International Journal of Bio-Inspired Computation* 4 (5): 330–343.

Mozaffari, A., Fathi, A. and Behzadipour, S. 2012. The great salmon run: a novel bio-inspired algorithm for artificial system design and optimisation. *International Journal of Bio-Inspired Computation* 4 (5): 286–301.

Mu'alem, A. W. and Feitelson, D. G. 2001. Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling. *IEEE Trans. Parallel Distrib. Syst.* 12 (6): 529–543.

Muangkote, N., Sunat, K. and Chiewchanwattana, S. 2014. An improved grey wolf optimizer for training q-Gaussian Radial Basis Functional-link nets. In *2014 international computer science and engineering conference (ICSEC)*, 209–214. IEEE.

Nakrani, S. and Tovey, C. 2004. On honey bees and dynamic server allocation in internet hosting centers. *Adaptive Behavior* 12 (3-4): 223–240.

Nelson, J. and Palmieri, R. 2019. Understanding RDMA Behavior in NUMA Systems. In *2019 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, 273–274.

Neth, H. and Gigerenzer, G. 2015. Heuristics: Tools for an uncertain world. *Emerging trends in the social and behavioral sciences: An Interdisciplinary, Searchable, and Linkable Resource* 1–18.

Ngubiri, J. 2008, Techniques and evaluation of processor co-allocation in multi-cluster systems.

Nikbakht, H. and Mirvaziri, H. 2015. A new clustering approach based on K-means and Krill Herd algorithm. In *2015 23rd Iranian Conference on Electrical Engineering*, 662–667. IEEE.

Nishimura, J. and Ugander, J. 2013. Restreaming graph partitioning: simple versatile algorithms for advanced balancing. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1106–1114. ACM.

Nuzzolo, A. 2002, In Advanced Modeling for Transit Operations and Service Planning, In *Advanced Modeling for Transit Operations and Service Planning*, 125–163, Elsevier Science New York, 125–163.

Obaida, M. A. and Liu, J. 2017. Simulation of HPC job scheduling and large-scale parallel workloads. In *2017 Winter Simulation Conference (WSC)*, 920–931. IEEE.

Odili, J. B., Kahar, M. N. M. and Anwar, S. 2015. African buffalo optimization: A swarm-intelligence technique. *Procedia Computer Science* 76: 443–448.

Omran, M. G. 2016. A novel cultural algorithm for real-parameter optimization. *International Journal of Computer Mathematics* 93 (9): 1541–1563.

Oral, M. and Turgut, S. S. 2018. Performance Analysis of Relatively New Nature Inspired Computing Algorithms. In *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1–6. IEEE.

Orr, M. and Sinnen, O. 2019a. Optimal Task Scheduling Benefits From a Duplicate-Free State-Space. *arXiv preprint arXiv:1901.06899* .

Orr, M. and Sinnen, O. 2019b. Parallel and Memory-limited Algorithms for Optimal Task Scheduling Using a Duplicate-Free State-Space. *arXiv preprint arXiv:1905.05568* .

Osipov, V. and Sanders, P. 2010. n-Level graph partitioning. In *European Symposium on Algorithms*, 278–289. Springer.

Oskooei, A. R., Mirza-Aghatabar, M. and Khorsandi, S. 2008. Introduction of novel rule based algorithms for scheduling in grid computing systems. In *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, 138–143. IEEE.

Panda, S. K., Pande, S. K. and Das, S. 2018. Task partitioning scheduling algorithms for heterogeneous multi-cloud environment. *Arabian Journal for Science and Engineering* 43 (2): 913–933.

Park, G.-L., Shirazi, B. and Marquis, J. 1997. DFRN: A new approach for duplication based scheduling for distributed memory multiprocessor systems. In *Proceedings 11th International Parallel Processing Symposium*, 157–166. IEEE.

Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. and Zaidi, M. 2005. Technical note: bees algorithm. *Manufacturing Engineering Centre, Cardiff University, Cardiff* .

Pothen, A., Simon, H. D. and Liou, K.-P. 1990. Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications* 11 (3): 430–452.

Predari, M. and Esnard, A. 2016. A k-way Greedy Graph Partitioning with Initial Fixed Vertices for Parallel Applications. In *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 280–287. IEEE.

Pu, L. and Faltings, B. 2012. Hypergraph learning with hyperedge expansion. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 410–425. Springer.

Qureshi, K., Rehman, A. and Manuel, P. 2011. Enhanced GridSim architecture with load balancing. *The Journal of Supercomputing* 57 (3): 265–275.

Radulescu, A. and Van Gemund, A. J. 1999. FLB: Fast load balancing for distributed-memory machines. In *Proceedings of the 1999 International Conference on Parallel Processing*, 534–541. IEEE.

Rafieerad, A., Bushroa, A., Nasiri-Tabrizi, B., Fallahpour, A., Vadivelu, J., Musa, S. and Kaboli, S. 2016. GEP-based method to formulate adhesion strength and hardness of Nb PVD coated on Ti–6Al–7Nb aimed at developing mixed oxide nanotubular arrays. *Journal of the mechanical behavior of biomedical materials* 61: 182–196.

Rafieerad, A., Bushroa, A., Nasiri-Tabrizi, B., Kaboli, S., Khanahmadi, S., Amiri, A., Vadivelu, J., Yusof, F., Basirun, W. and Wasa, K. 2017. Toward improved mechanical, tribological, corrosion and in-vitro bioactivity properties of mixed oxide nanotubes on Ti–6Al–7Nb implant using multi-objective PSO. *Journal of the mechanical behavior of biomedical materials* 69: 1–18.

Raghavan, P. 1997. Parallel ordering using edge contraction. *Parallel Computing* 23 (8): 1045–1067.

Rajasekhar, A., Rani, R., Ramya, K. and Abraham, A. 2012. Elitist teaching learning opposition based algorithm for global optimization. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1124–1129. IEEE.

Ramamoorthy, C. V., Chandy, K. M. and Gonzalez, M. J. 1972. Optimal Scheduling Strategies in a Multiprocessor System. *IEEE Transactions on Computers* C-21 (2): 137–146.

Ranaweera, S. and Agrawal, D. P. 2000. A task duplication based scheduling algorithm for heterogeneous systems. In *Proceedings 14th International Parallel and Distributed Processing Symposium. IPDPS 2000*, 445–450. IEEE.

Rao, R. and Patel, V. 2012. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations* 3 (4): 535–560.

Rao, R. V., Savsani, V. J. and Vakharia, D. 2012. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences* 183 (1): 1–15.

Roig, C., Ripoll, A. and Guirado, F. 2007a. A new task graph model for mapping message passing applications. *IEEE transactions on Parallel and Distributed Systems* 18 (12): 1740–1753.

Roig, C., Ripoll, A. and Guirado, F. 2007b. A New Task Graph Model for Mapping Message Passing Applications. *IEEE Transactions on Parallel and Distributed Systems* 18 (12): 1740–1753.

Roig, C., Ripoll, A., Senar, M. A., Guirado, F. and Luque, E. 2000a. Modelling message-passing programs for static mapping. In *Proceedings 8th Euromicro Workshop on Parallel and Distributed Processing*, 229–236. IEEE.

Roig, C., Ripoll, A., Senar, M. A., Guirado, F. and Luque, E. 2000b. Modelling message-passing programs for static mapping. In *Proceedings 8th Euromicro Workshop on Parallel and Distributed Processing*, 229–236.

Roig, C., Ripoll, A., Senar, M. A., Guirado, F. and Luque, E. 2001. A new model for static mapping of parallel applications with task and data parallelism. In *Proceedings 16th International Parallel and Distributed Processing Symposium*, 8–pp. IEEE.

Sagan, H. 2012. *Space-filling curves*. Springer Science & Business Media.

Said, G. A. E.-N. A., Mahmoud, A. M. and El-Horbaty, E.-S. M. 2014. A comparative study of meta-heuristic algorithms for solving quadratic assignment problem. *arXiv preprint arXiv:1407.4863* .

Sakouhi, C., Aridhi, S., Guerrieri, A., Sassi, S. and Montresor, A. 2016. Dynamicd-fep: A distributed edge partitioning approach for large dynamic graphs. In *Proceedings of the 20th International Database Engineering & Applications Symposium*, 142–147. ACM.

Sanders, P. and Schulz, C. 2011. Engineering multilevel graph partitioning algorithms. In *European Symposium on Algorithms*, 469–480. Springer.

Sanders, P. and Schulz, C. 2013a. High quality graph partitioning. *Contemporary Mathematics* 588.

Sanders, P. and Schulz, C. 2013b. Think locally, act globally: Highly balanced graph partitioning. In *International Symposium on Experimental Algorithms*, 164–175. Springer.

Sandokji, S. and Eassa, F. 2019. Dynamic Variant Rank HEFT Task Scheduling Algorithm Toward Exascle Computing. *Procedia Computer Science* 163: 482–493.

Sanyal, S. and Das, S. K. 2005. MaTCH : Mapping Data-Parallel Tasks on a Heterogeneous Computing Platform Using the Cross-Entropy Heuristic. In *19th IEEE International Parallel and Distributed Processing Symposium*, 64b–64b.

Sarangal, A., Kaur, N. and Kaur, R. 2018. LIST SCHEDULING ALGORITHMS CLASSIFICATION: AN ANALYTICAL STUDY. *International Journal of Advanced Research in Computer Science* 9 (2): 385.

Sarkar, V. 1989, Partitioning and Scheduling Parallel Programsfor Multiprocessors.

Schlag, S. 2017, A Benchmark Set for Multilevel Hypergraph Partitioning Algorithms.

Schlag, S., Henne, V., Heuer, T., Meyerhenke, H., Sanders, P. and Schulz, C. 2016. K-way hypergraph partitioning via n-level recursive bisection. In *2016 Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, 53–67. SIAM.

Schmitt, M. and Wanka, R. 2015. Particle swarm optimization almost surely finds local optima. *Theoretical Computer Science* 561: 57–72.

Schneider, M. 1991. Tying the Knot Between Serial and Massively Parallel Computing: Pittsburgh's Not-So-Odd Couple. *Supercomputing Review* 4 (8): 36–38.

Schweickardt, G. A., Miranda, V. and Wiman, G. 2011. A comparison of metaheuristics algorithms for combinatorial optimization problems. Application to phase balancing in electric distribution systems .

Sebtahmadi, S. S., Azad, H. B., Kaboli, S. H., Islam, M., Mekhilef, S. *et al.* 2018. A PSO-DQ Current Control Scheme for Performance Enhancement of Z-Source Matrix Converter to Drive IM Fed by Abnormal Voltage. *IEEE Transactions on Power Electronics* 33 (2): 1666–1681.

Sellmann, M., Sensen, N. and Timajev, L. 2003. Multicommodity flow approximation used for exact graph partitioning. In *European Symposium on Algorithms*, 752–764. Springer.

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

Sensen, N. 2001. Lower bounds and exact algorithms for the graph partitioning problem using multicommodity flows. In *European Symposium on Algorithms*, 391–403. Springer.

Shaheen, A. M., Spea, S. R., Farrag, S. M. and Abido, M. A. 2018. A review of meta-heuristic algorithms for reactive power planning problem. *Ain Shams Engineering Journal* 9 (2): 215–231.

Sharma, D. and Yadav, N. K. 2019. Lion Algorithm with Levy Update: Load frequency controlling scheme for two-area interconnected multi-source power system. *Transactions of the Institute of Measurement and Control* 41 (14): 4084–4099.

Shaydulin, R., Chen, J. and Safro, I. 2019. Relaxation-based coarsening for multi-level hypergraph partitioning. *Multiscale Modeling & Simulation* 17 (1): 482–506.

Shen, H., Lv, S., Dong, X., Deng, J., Wang, X. and Zhou, X. 2013. Hypergraph Modeling and Approximation Algorithms for the Minimum Length Link Scheduling in Multiuser MIMO Networks. *Journal of Applied Mathematics* .

Siarry, P., Pétrowski, A. and Bessaou, M. 2002. A multipopulation genetic algorithm aimed at multimodal optimization. *Advances in Engineering Software* 33 (4): 207–213.

Sih, G. C. and Lee, E. A. 1993. A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures. *IEEE transactions on Parallel and Distributed systems* 4 (2): 175–187.

Simon, H. D. 1991. Partitioning of unstructured problems for parallel processing. *Computing systems in engineering* 2 (2-3): 135–148.

Singh, K. V. and Raza, Z. 2015. A GA based job scheduling strategy for computational grid. In *Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in*, 29–34.

Singh, P. R., Moussa, D., Shengwu, X. and Singh, B. P. 2019. Improved Spider Monkey Optimization Algorithm to train MLP for data classification .

Singla, M. K. and Scholar, M. 2013. Task Scheduling Algorithms for Grid Computing with Static Jobs: A Review. *International Journal of Computer Science Engineering (IJCSE)* 2 (5): 218.

Soleymani, M. and Nematzadeh, H. 2017. Scheduling of Real-time Processes Distribution on Multiprocessor Using Meta-Heuristic Ant Colony Algorithms, Genetics and PSO. *Journal of Advances in Computer Research* 8 (3): 55–68.

Somasundaram, K. and Radhakrishnan, S. 2009. Task resource allocation in grid using swift scheduler. *International Journal of Computers Communications & Control* 4 (2): 158–166.

Soper, A. J., Walshaw, C. and Cross, M. 2004. A combined evolutionary search and multilevel optimisation approach to graph-partitioning. *Journal of Global Optimization* 29 (2): 225–241.

Sörensen, K. 2015. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research* 22 (1): 3–18.

Srinivasan, S., Kettimuthu, R., Subramani, V. and Sadayappan, P. 2002a. Characterization of backfilling strategies for parallel job scheduling. In *Proceedings. International Conference on Parallel Processing Workshop*, 514–519. IEEE.

Srinivasan, S., Kettimuthu, R., Subramani, V. and Sadayappan, P. 2002b. Selective reservation strategies for backfill job scheduling. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 55–71. Springer.

Srivastava, S. and Banicescu, I. 2018, In Topics in Parallel and Distributed Computing, In *Topics in Parallel and Distributed Computing*, 313–337, Springer, 313–337.

Stanton, I. and Kliot, G. 2012. Streaming graph partitioning for large distributed graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1222–1230. ACM.

Storn, R. and Price, K. 1995. *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces.* , vol. 3. ICSI Berkeley.

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

Sulistio, A., Poduval, G., Buyya, R. and Tham, C.-K. 2005. Constructing A Grid Simulation with Differentiated Network Service Using GridSim. In *International Conference on Internet Computing*, 437–444.

Suman, B. and Kumar, P. 2006. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the operational research society* 57 (10): 1143–1160.

Surjanovic, S. and Bingham, D. 2013. Virtual library of simulation experiments: test functions and datasets. *Simon Fraser University, in: http://www. sfu. ca/˜ ssurjano/optimization. html [Links]* .

Swami, V., Kumar, S. and Jain, S. 2018, In Soft Computing: Theories and Applications, In *Soft Computing: Theories and Applications*, 73–81, Springer, 73–81.

Talbi, E.-G. 2009. *Metaheuristics: from design to implementation.* , vol. 74. John Wiley & Sons.

Talu, M. F. 2017. Multi-level spectral graph partitioning method. *Journal of Statistical Mechanics: Theory and Experiment* 2017 (9): 093406.

Tan, W.-S., Hassan, M. Y., Majid, M. S. and Rahman, H. A. 2013. Optimal distributed renewable generation planning: A review of different approaches. *Renewable and Sustainable Energy Reviews* 18: 626–645.

Teresa, M. 1995. *A simple path*. New York: Ballantine Books; Toronto: Random House of Canada.

Tian, T., Liu, C., Guo, Q., Yuan, Y., Li, W. and Yan, Q. 2018. An improved ant lion optimization algorithm and its application in hydraulic turbine governing system parameter identification. *Energies* 11 (1): 95.

Tong, C. and Wong, S. 1999. A schedule-based time-dependent trip assignment model for transit networks. *Journal of Advanced Transportation* 33 (3): 371–388.

Topcuoglu, H., Hariri, S. and Wu, M.-y. 2002. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE transactions on parallel and distributed systems* 13 (3): 260–274.

Torn, A. and Zilinskas, A. 1989. *Global Optimization*. New York, NY, USA: Springer-Verlag New York, Inc.

Trifunovic, A. and Knottenbelt, W. J. 2004. Parkway 2.0: A parallel multilevel hypergraph partitioning tool. In *International Symposium on Computer and Information Sciences*, 789–800. Springer.

Tsourakakis, C., Gkantsidis, C., Radunovic, B. and Vojnovic, M. 2014. Fennel: Streaming graph partitioning for massive scale graphs. In *Proceedings of the 7th ACM international conference on Web search and data mining*, 333–342. ACM.

Tsuchiya, T., Osada, T. and Kikuno, T. 1998. Genetics-based multiprocessor scheduling using task duplication. *Microprocessors and Microsystems* 22 (3-4): 197–207.

Turing, A. M. 1948. Intelligent Machinery, A Heretical Theory. *The Turing Test: Verbal Behavior as the Hallmark of Intelligence* 105.

Ullman, J. D. 1975. NP-complete scheduling problems. *Journal of Computer and System sciences* 10 (3): 384–393.

Upadhyaya, G. and Rajan, H. 2015. Effectively mapping linguistic abstractions for message-passing concurrency to threads on the Java virtual machine. In *ACM SIGPLAN Notices*, 840–859. ACM.

Valiant, L. G. 1990. A bridging model for parallel computation. *Communications of the ACM* 33 (8): 103–111.

Vallada, E. and Ruiz, R. 2011. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research* 211 (3): 612–622.

van Dongen, S. 2000. *Graph clustering by flow simulation*. PhD thesis, University of Utrecht.

Van Driessche, R. 1997, Algorithms for static and dynamic load balancing on parallel computers.

Vastenhouw, B. and Bisseling, R. H. 2005. A two-dimensional data distribution method for parallel sparse matrix-vector multiplication. *SIAM review* 47 (1): 67–95.

Vijayvargiya, P. 2019. A Comparative Study of CPU Scheduling Algorithms. *Proceedings of Recent Advances in Interdisciplinary Trends in Engineering & Applications (RAITEA)* .

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing* 17 (4): 395–416.

Walshaw, C. and Cross, M. 2007. JOSTLE: parallel multilevel graph-partitioning software–an overview. *Mesh partitioning techniques and domain decomposition techniques* 27–58.

Walton, S., Hassan, O., Morgan, K. and Brown, M. 2011. Modified cuckoo search: a new gradient free optimisation algorithm. *Chaos, Solitons & Fractals* 44 (9): 710–718.

Wang, W., Du, S., Guo, Z. and Luo, L. 2015. Polygonal Clustering Analysis Using Multilevel Graph-Partition. *Transactions in GIS* 19 (5): 716–736.

Wang, Y., Lü, Z., Glover, F. and Hao, J.-K. 2012. A multilevel algorithm for large unconstrained binary quadratic optimization. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, 395–408. Springer.

Williams, R. D. 1991. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Practice and experience* 3 (5): 457–481.

Williamson, D. P. and Shmoys, D. B. 2011. *The design of approximation algorithms*. Cambridge university press.

Wolberg, J. 2006. *Data analysis using the method of least squares: extracting the most information from experiments*. Springer Science & Business Media.

Wolpert, D. H. and Macready, W. G. 1997. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on* 1 (1): 67–82.

Wu, D., Kong, F., Gao, W., Shen, Y. and Ji, Z. 2015. Improved chicken swarm optimization. In *2015 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER)*, 681–686. IEEE.

Wu, D., Xu, S. and Kong, F. 2016a. Convergence analysis and improvement of the chicken swarm optimization algorithm. *IEEE Access* 4: 9400–9412.

Wu, G., Mallipeddi, R., Suganthan, P. N., Wang, R. and Chen, H. 2016b. Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences* 329: 329–345.

Wu, M.-Y. 2000. MCP revisited. *Department of Electrical and Computer Engineering, University of New Mexico* .

Wu, M.-Y. and Gajski, D. D. 1990. Hypertool: A programming aid for message-passing systems. *IEEE transactions on parallel and distributed systems* 1 (3): 330–343.

Wu, Z., Yu, D. and Kang, X. 2018. Application of improved chicken swarm optimization for MPPT in photovoltaic system. *Optimal Control Applications and Methods* 39 (2): 1029–1042.

Xhafa, F. and Abraham, A. 2010. Computational models and heuristic methods for Grid scheduling problems. *Future generation computer systems* 26 (4): 608–621.

Xhafa, F., Kolodziej, J., Barolli, L. and Fundo, A. 2011. A ga+ ts hybrid algorithm for independent batch scheduling in computational grids. In *2011 14th International Conference on Network-Based Information Systems*, 229–235. IEEE.

Xue, B., Zhang, M., Browne, W. N. and Yao, X. 2015. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation* 20 (4): 606–626.

Xue, W. and Qi, S. 2007. Multilevel task partition algorithm for parallel simulation of power system dynamics. In *International Conference on Computational Science*, 529–537. Springer.

Yang, X.-S. 2008. Firefly algorithm. *Nature-inspired metaheuristic algorithms* 20: 79–90.

Yang, X.-S. 2010, In Nature inspired cooperative strategies for optimization (NICSO 2010), In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65–74, Springer, 65–74.

Yang, X.-S. 2012. Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation*, 240–249. Springer.

Yang, X.-S. and Deb, S. 2009. Cuckoo search via Lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 210–214. IEEE.

Yang, X.-S. and Deb, S. 2013. Multiobjective cuckoo search for design optimization. *Computers & Operations Research* 40 (6): 1616–1624.

Yang, X.-S. and He, X. 2013. Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation* 5 (3): 141–149.

Yang, X.-S., Karamanoglu, M. and He, X. 2014. Flower pollination algorithm: a novel approach for multiobjective optimization. *Engineering optimization* 46 (9): 1222–1237.

Yu, K., Wang, X. and Wang, Z. 2016. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. *Journal of Intelligent Manufacturing* 27 (4): 831–843.

Yu, Y., Ma, H., Yu, M., Ye, S. and Chen, X. 2018. Multipopulation management in evolutionary algorithms and application to complex warehouse scheduling problems. *Complexity* 2018.

Yu-Kwong Kwok, Y. K. and Ahmad, I. 1994. A Static Scheduling Algorithm Using Dynamic Critical Path for Assigning Parallel Algorithms onto Multiprocessors. In *1994 Internatonal Conference on Parallel Processing Vol. 2*, 155–159.

Zhang, L., Liu, L., Yang, X.-S. and Dai, Y. 2016. A novel hybrid firefly algorithm for global optimization. *PloS one* 11 (9): e0163230.

Zhang, S., He, J., Zhou, A. C. and He, B. 2019. BriskStream: Scaling Data Stream Processing on Shared-Memory Multicore Architectures. *ACM SIGMOD/PODS International Conference on Management of Data 2019, arXiv preprint arXiv:1904.03604* .

Zheng, G., Kakulapati, G. and Kalé, L. V. 2004. Bigsim: A parallel simulator for performance prediction of extremely large parallel machines. In *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, 78. IEEE.

Zhou, A.-H., Zhu, L.-P., Hu, B., Deng, S., Song, Y., Qiu, H. and Pan, S. 2019. Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming. *Information* 10 (1): 7.

Zhou, Y., Yang, X., Ling, Y. and Zhang, J. 2018. Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation. *Multimedia Tools and Applications* 77 (18): 23699–23727.

Zien, J. Y., Schlag, M. D. and Chan, P. K. 1999. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Transactions on computer-aided design of integrated circuits and systems* 18 (9): 1389–1399.

Zumbusch, G. 2012. *Parallel Multilevel Methods: Adaptive Mesh Refinement and Loadbalancing*. Springer Science & Business Media.