

Review Article

Comprehensive Review on Divisible Load Theory: Concepts, Strategies, and Approaches

Shamsollah Ghanbari¹ and Mohamed Othman^{1,2}

¹ Computational Science and Mathematical Physics Lab., Institute for Mathematical Research, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia

² Department of Communication Tech and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia

Correspondence should be addressed to Mohamed Othman; mothman@upm.edu.my

Received 25 February 2014; Accepted 20 June 2014; Published 20 July 2014

Academic Editor: Carsten Proppe

Copyright © 2014 S. Ghanbari and M. Othman. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There is extensive literature concerning the divisible load theory. The divisible load theory is mainly applied for scheduling in the area of distributed computing. It is based on the fact that the load can be divided into some arbitrarily independent parts, in which each part can be processed independently by a processor. This paper reviews the literature concerning the divisible load theory, while focusing on the details of the basic concepts, approaches, strategies, typologies, and open problems.

1. Introduction

The first articles about the divisible load theory (DLT) were published in 1988 [1, 2]. Based on the DLT, the load can be partitioned into some arbitrary size, in which each part can be executed independently by a processor [3]. Some advantages of the DLT have been listed in [4]. In the past two decades the DLT has found a wide variety of applications in the area of parallel processing. During these two decades, many researchers have investigated various aspects of the DLT [1–131]. In this paper, we have classified the existing research and provided a survey on the DLT in which we focus on the concepts, strategies, approaches, applications, and open problems. The main objectives of this paper are to

- (i) provide a comprehensive review of the literature;
- (ii) categorize the literature according to the typology of the research;
- (iii) explore the open problems in the field of the divisible load theory;
- (iv) summarize the existing research results for the different types of problem.

2. A Bird's Eye Review on the Divisible Load Theory

The divisible load theory originated in 1988 [1, 2]. Five years later, a formal mathematical proof appeared [6], and closed forms for the divisible load theory under the bus and tree networks topologies were proposed [7]. Subsequently, the DLT model with start-up costs was proposed [28] and the effects of start-up delays on divisible load scheduling on bus networks were investigated [96]. A few years later, the model was examined concerning the processors with front-end properties [46].

According to the literature, three main strategies have been applied on the DLT in order to improve the performance. The *first* strategy is multi-installment processing, which was proposed for the first time in 1995 [13]. Multi-installment processing can reduce the scheduling time in the DLT [93]. Subsequently, multi-installment divisible load was improved by other researchers. The *second* strategy is an adaptive strategy that estimates network parameter values using a probing technique. The adaptive model can also reduce the scheduling time. It was proposed in [119] for the first time. Subsequently, a genetic algorithm based method for the adaptive DLT was proposed [66]. A few years later, that

work was improved and other adaptive algorithms, including ADLT, A^2DLT , and IDLT, were proposed [65, 66, 68]. The *third* proposed strategy for improving the performance of the DLT was a multi-source divisible load which was proposed by [114, 115]. A few years later, multi-source DLT was developed in order to calculate closed form formulation [70]. Subsequently, multi-source divisible load was continued by the other researchers [52, 67, 114, 124].

In the early part of 2002, a group of researchers proved that the DLT with limited memory is NP-hard problem [33, 35]. Subsequently, the DLT with finite-size buffers in the heterogeneous single level tree networks was investigated [30]. Simultaneously, the investigation was continued by analyzing the effects of finite-size buffers on the multi-installment divisible load [47]. Subsequently, the DLT with limited memory was investigated [56]. The investigation was continued by the application of limited memory on both in the multi-installment divisible load [93] and heterogeneous divisible load [100]. A few years later, the limitation of memory was comprehensively investigated [64]. We have found some intensive surveys on the DLT. The *first* one is a technical review concerning the DLT, which includes models, approaches, performance analysis, applications, and concepts [42]. The *second* survey explores open problems in the DLT [105]. The *third* survey categorizes the research concerning the DLT as well as its applications [106]. In addition, for further studies some comprehensive information concerning the DLT can be found in [17, 92, 112]. Table 1 indicates a brief list of topics in the divisible load area.

3. Concepts of Divisible Load Theory

3.1. Basic Definitions and Notations. Notation section indicates the basic notations in the divisible load theory.

3.2. Mathematical Model for the Divisible Load Theory

3.2.1. Load Allocation in Single Level Tree (Star) Network Topology. In general the DLT assumes that the computation and communication can be divided into some parts of arbitrary sizes, and these parts can be independently processed in parallel. The DLT assumes that, initially, amount V of load is held by the originator p_0 . A common assumption is that the originator does not conduct any computation. It only distributes the load into parts $\alpha_0, \alpha_1, \dots, \alpha_m$ to be processed on worker processors p_0, p_1, \dots, p_m . The condition for the optimal solution is that the processors stop processing at the same time; otherwise, the load could be transferred from busy to idle processors to improve the solution time [3]. The goal is to calculate $\alpha_0, \alpha_1, \dots, \alpha_m$ in the DLT timing equation. In general, the timing equations (i.e., closed form) for the DLT can be depicted as follows:

$$T_i(\alpha) = \begin{cases} \alpha_0 w_0 & i = 0 \\ \sum_{j=1}^i \alpha_j z_j + \alpha_i w_i & i = 1, 2, \dots, m, \end{cases} \quad (1)$$

TABLE 1: A brief list of studied topics in the area of divisible load theory.

Topics	Developer(s)	References
The DLT originated	Cheng, Agrawal	[1, 2]
Closed form for bus and tree	Robertazzi, Sameer, Hsiung	[6, 7]
Optimal condition for the DLT	Sohn, Robertazzi	[3]
Linear programming model for the DLT	Hillier, Robertazzi	[112, 131]
Multi-installment processing of the DLT	Veeravalli, Ghose, mani	[13]
The DLT with finite-size buffers	Veeravalli, Li, Ko	[32]
The DLT with memory limitation	Drozdowski, Berlinska	[33, 35, 64]
The DLT applied for grid computing	Moges, Yu, Robertazzi	[70, 75]
The adaptive DLT	Ghose, Kim, Kim	[55, 119]
Cheat processors in the DLT	Carroll, Grosu	[62, 63]
The DLT with nonlinear cost	Tsun, Suresh, Kim, Robertazzi	[53, 71]
The DLT and Markov chain models	Moges, Robertazzi	[51, 118]
Multi-source/Multi-site	Moges, Veeravalli, Li, Min, Yu, Robertazzi	[52, 67] [114, 115]
The real time DLT	Lin, Mamat, Lu	[58, 61, 108]
Time varying in the DLT	Sohn, Robertazzi	[16, 19]
Multi-criteria divisible load	Ghanbari, Othman	[77]
Complexity problem concerning the DLT	Yang, Casanova, Drozdowski, Berlinska	[64, 76]

where $\alpha_0 + \alpha_1 + \dots + \alpha_m = V$. Moreover, α_i can be calculated as the following equations:

$$\begin{aligned} \alpha_i &= k_i \alpha_{i-1}, \\ k_j &= \frac{w_{j-1}}{z_j + w_j}, \\ \alpha_0 &= \frac{V}{1 + \sum_{i=1}^m \prod_{j=1}^i k_j}. \end{aligned} \quad (2)$$

In this case, we assume that the load is sequentially distributed and simultaneously executed. As a result, the load is distributed from a root node to one child at a time. A Gantt chart-like timing diagram for this case can be depicted as Figure 1(a), while Figure 1(b) indicates that the load is simultaneously distributed with the starting staggered. Therefore, the worker processors are connected to the root processor by direct communication links. In this case, after receiving all of its assigned fraction of the load, each processor begins computing immediately. A timing equation for this form can be found in [112]. Furthermore, the time equation

and Gantt chart-like diagram for the other possible forms of distribution and execution in the DLT model can be found in [92, 112].

3.2.2. Load Allocation in Multi-Level Tree. In a multi-level tree, the load is distributed from top to bottom, passing through each level. The optimal solution is obtained by traversing the tree from bottom to top, replacing single level subtrees with single equivalent processors until T is reduced to one processor, which is denoted by w_i^{eq} . The Gantt chart-like timing diagram for multi-level tree divisible load is depicted in Figure 2(a).

3.2.3. Other Topologies. Up until 1999, the closed form for the DLT with various topologies including hypercubes [14], arbitrary graphs [99], daisy chain networks [6, 100], two-dimensional meshes [24], and three-dimensional meshes [25] were proposed. Subsequently, a closed form for k -dimensional meshes was proposed [50].

4. Strategies

In this section, we explain some strategies that have been applied to the divisible load in order to improve the performance. According to the literature, there are three main strategies including multi-installment, adaptive, and multi-source.

4.1. Multi-Installment Divisible Load. Based on multi-installment (i.e., multi-round) processing, the load is sent to a processor in more than one chunk. Therefore, the processor p_1 will start execution earlier, and the whole processing time will be shorter. Multi-installment processing may reduce the scheduled length to 0.632 of the initial length [56, 92]. It also depends on the proper number of installments [60]. The first paper about multi-installment divisible load was published in 1995 [8, 13]. Subsequently, a closed form formula for multi-installment divisible load was proposed in [8, 13]. A few years later, the multi-installment DLT was developed by considering the communication start-up cost function [38]. A year later, the memory limitation in multi-installment was investigated [93]. Subsequently, the multi-installment divisible load method was improved on k -dimensional meshes [95]. According to researchers the multi-installment divisible load consists of two main forms including uniform multi-round (UMR) and robust-UMR (RUMR) [54]. In the recent research related to multi-installment divisible load a number of heuristics for the scheduling of multi-installment divisible loads were proposed [64]. The time Gantt chart-like diagram for multi-installment divisible load is shown in Figure 2(b). Moreover, Table 2 summarizes the research related to the multi-installment divisible load theory.

4.2. Adaptive Divisible Load Model. Initially, the adaptive strategy was introduced under the name of feedback strategy for divisible load allocation [119]. Adaptive divisible load scheduling requires less finish time than the other methods [55]. The basic adaptive load distribution strategy consists of

TABLE 2: Important events in multi-installment divisible load scheduling.

Author(s)	Description/issues
Veeravalli et al. [8]	Multi-installment DLT originated
Veeravali et al. [13]	Closed form for multi-installment divisible load was propped
Wolniewicz [56]	Effects of multi-installment processing on the total finish time in divisible load scheduling
Yang and Casanova [54]	UMR and RUMR were classified
Drozdowski and Wolniewicz [33], Yang et al. [76]	Complexity of multi-installment divisible load was studied
Drozdowski and Lawenda [93]	Multi-installment DLT with limited memory
Berlińska and Drozdowski [64]	Heuristics for multi-installment divisible load

two phases. In the first phase (probe phase), a small part of the load (probe installment) is partitioned and communicated to individual processing nodes. Each node sends back a communication task completion message when it finishes receiving its load fraction. The second phase is called the optimal load distribution phase [55]. Furthermore, three adaptive strategies for divisible load were proposed. The three proposed strategies are as follows:

- (i) probing and delayed distribution (PDD) strategy;
- (ii) probing and continuous distribution (PCD) strategy;
- (iii) probing and selective distribution (PSD) strategy.

Furthermore, closed form formulations for PDD, PCD, and PSD have been calculated and can be found in [55]. In addition, some useful conclusions concerning the adaptive divisible load can be found in [65, 66, 68].

4.3. Divisible Load with Multiple Sources/Sites. The basic idea about the divisible load scheduling was based on the fact that the load originated from a single (root) processor. In 1994, a multi-job divisible job on bus networks was proposed [9]. The results of that work indicated that a multi-job scheme outperforms a single-job scheme in terms of the total solution finish time. After that, the divisible load with multiple sources (i.e., multiple sites) was studied in [124] for the first time and developed in [52]. The last paper focused on multiple source grid scheduling with capacity limitations. A few years later, grid scheduling divisible loads from multiple sources using the linear programming method was investigated [115]. Subsequently, a close form for divisible load with multiple source was proposed [114]. Lastly, a closed form of divisible load with two sources was calculated [70]. That paper contains several open problems and new ideas for further research in the field of divisible load theory with multiple sources. More recently, the divisible load with multiple sites in multi-tree network was investigated [67]. Although the results of the related research show that multiple source approach in divisible load

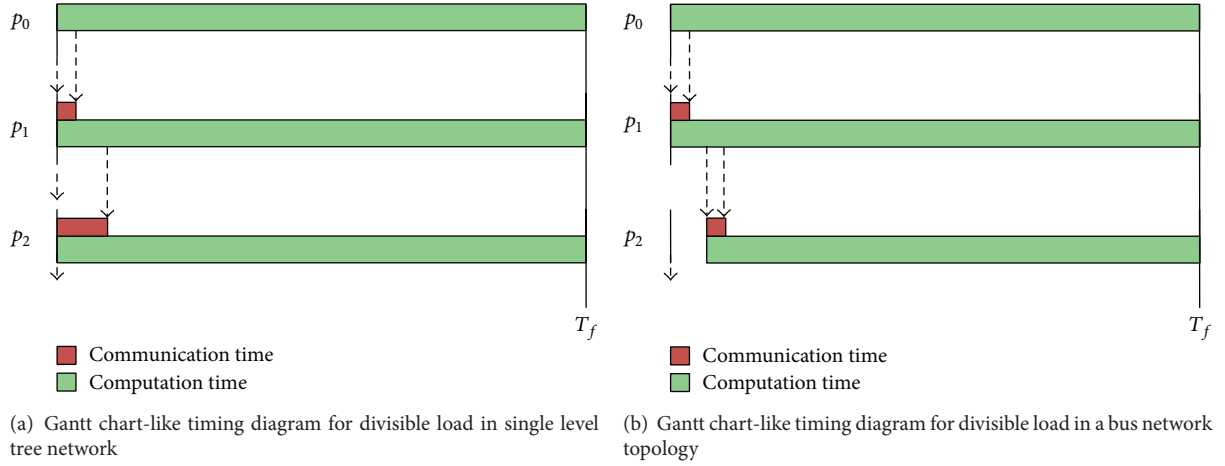


FIGURE 1: Gantt chart-like timing diagrams for divisible load.

may reduce scheduling time [67], a formal proof has not yet been published about this problem. Hence, investigating a formal proof for the effects of multi-sources divisible load can be considered as an open problem in this field.

5. Approaches

5.1. Divisible Load Model and Markov Chain. The equivalence between various divisible loads and continuous time Markov chain models helps in understanding the behavior of divisible load. These equivalences were investigated by Moges and Robertazzi [51, 118]. In [118] the closed form for divisible load with various topologies using the Markov chain model was demonstrated. Figure 3 shows the equivalence of Markov model and DLS with single tree level topology.

5.2. Linear Programming Approach in Divisible Load Scheduling. Linear programming is a traditional approach to solving optimization problems. It is easy to understand and simple in implementation. A linear programming model can be solved by various methods such as simplex and duality methods. There is a wide variety of software to solve a linear programming problem. Moreover, linear programming problem can be solved very fast. Fortunately, divisible load scheduling can be modeled as a linear programming problem. A simple form of divisible load scheduling has been modeled as linear programming in [112]. The complexity of this model in a simple form is $\Omega(n)$. Generally, divisible load scheduling may be modeled as a more complex form of linear programming; for example, a linear programming model for multi-round divisible load scheduling with limited memory is very complicated. It can be solved by branch and bound method which is an NP-hard problem [64]. The various linear programming model for multi-installment divisible load can be found in [92, 131].

5.3. Nonlinear Divisible Load. The general assumption is that both the computation and communication in divisible load model are linear. A nonlinear model is also possible for a

divisible load. The first nonlinear model was proposed in [112]. Subsequently, the nonlinear model of divisible load was developed by the researchers [53, 71]. The basic assumptions and definition of nonlinear divisible load, closed form for nonlinear divisible load, and Gantt chart-like timing diagram for nonlinear divisible load can be found in [71]. In general, a closed form for divisible load with nonlinear cost can be written as the following equation:

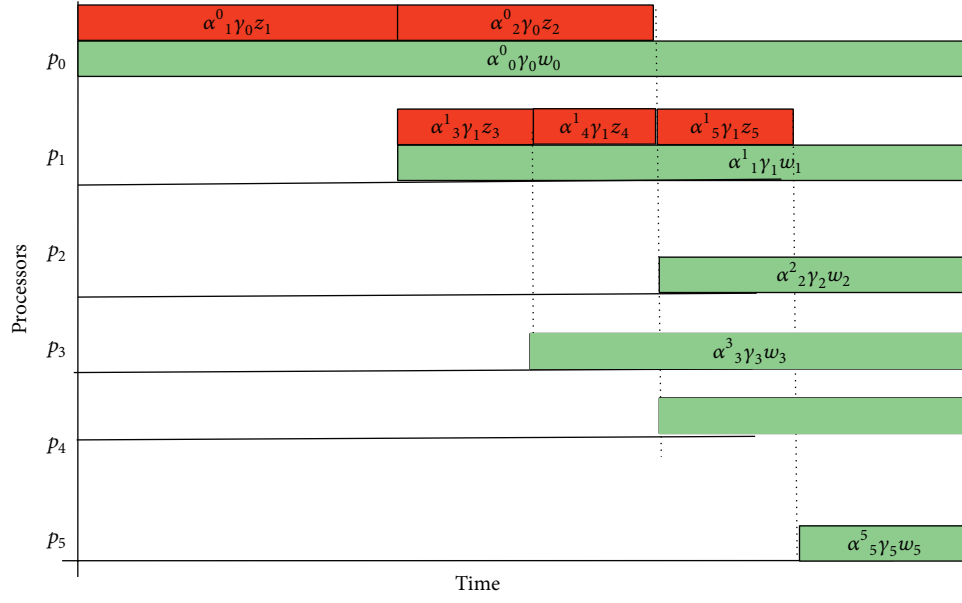
$$T_i(\alpha) = \begin{cases} \alpha_0 w_0 & i = 0 \\ \sum_{j=1}^i \alpha_j z_j^\gamma + \alpha_i w_i^\theta & i = 1, 2, \dots, m, \end{cases} \quad (3)$$

where $\alpha_0 + \alpha_1 + \dots + \alpha_m = V$.

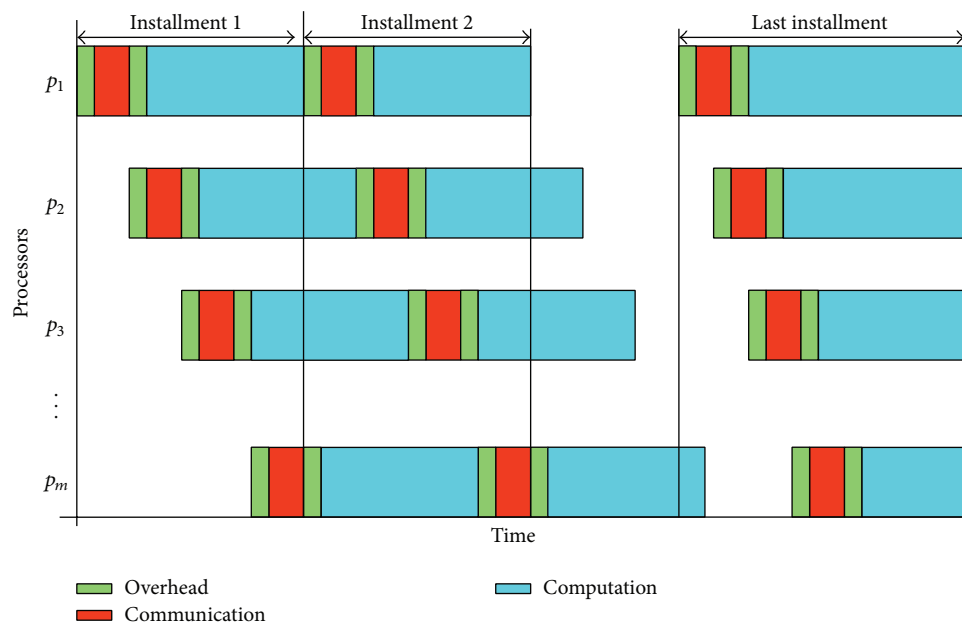
6. Issues and Challenges in Special Studies

This section mainly focuses on certain problems related to the features or limitations of processors or processing in divisible load scheduling.

6.1. Cheat Processors in the Divisible Load Model. The main idea of the processor cheating refers to misreporting and time varying problems, which were investigated in respect of divisible load scheduling in 1998 [63]. Subsequently, Carroll and Grosu focused on the application case of misreporting in the divisible load scheduling [62, 63]. They proposed a strategy proof mechanism for the divisible load scheduling under various topologies including the bus and multi-level tree network [62, 63, 102, 103]. However, the cheating problem may occur, if the processors execute their fraction of loads with different rate. Assume that p_0, p_1, \dots, p_m are $m + 1$ processors in a divisible load scheduling model. We also assume that p_0 is the root processor and p_1, p_2, \dots, p_m are the worker processors. Consider that $w = (w_0, w_1, \dots, w_m)$ and $z = (z_0, z_1, \dots, z_m)$ denote the computation and communication rates for processors, respectively. In the first stage, the computation rates of processors are sent to the root processor. The root processor allocates $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_m)$



(a) Gantt chart-like timing diagram for a multi-level tree network



(b) Gantt chart-like diagram for multi-round divisible load [60]

FIGURE 2: Gantt chart-like timing diagrams for divisible load.

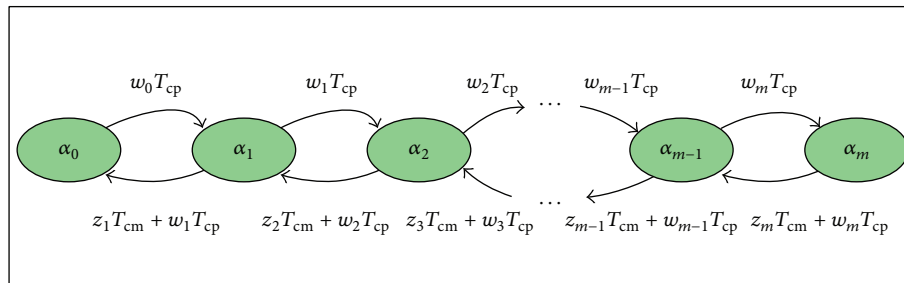


FIGURE 3: Markov model for single level tree divisible load [118].

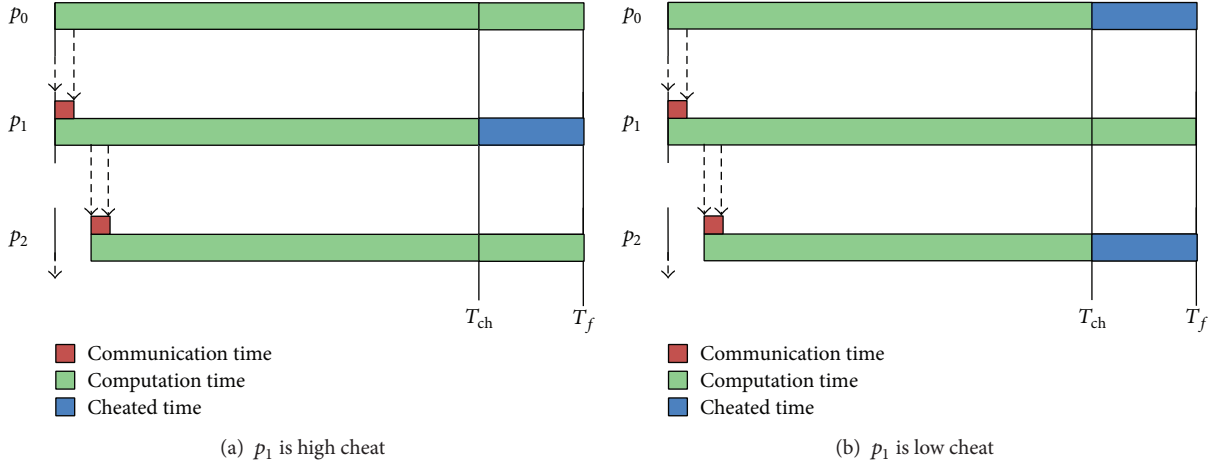


FIGURE 4: The effects of a cheat processor on the makespan.

fraction of load for the processors. The DLT with cheat processors has been investigated in several works [62, 63, 102, 103] by Carroll and Grosu. They examined various network topologies including bus [63], chain [102, 103], and multi-level tree [62]. They considered p_1 as the cheating processor, since its position gives a great influence over the allocations. They investigated the effects of cheating on the makespan, system utility, and verification. They proved that when the worker processors in a divisible load model do not report their true computation rates, they cannot obtain optimal performance. Figures 4(a) and 4(b) indicate the effects of the cheat processor on the makespan.

6.2. Time Varying. Time varying models are the most challenging problems in the recent research on the divisible load theory. It is also an open problem in the field of divisible load theory. The time varying problem means that communication and computation speeds may change in time. Time varying was discussed for the first time in [19]. Analysis of time varying in a continuous and discrete time can be found in [16, 19, 92, 112]. A closed form formula for divisible load with a time varying system can be formulated by changing the DLT model. In this case, *time* may change from 0 to T , while T is considered as the total scheduling time. A simple form of time varying models in divisible load can be formulated as the following equation:

$$T_i(\alpha) = \begin{cases} \alpha_0 w_0(t) & i = 0 \\ \sum_{j=1}^i \alpha_j z_j(t) + \alpha_i w_i(t) & i = 1, 2, \dots, m, \end{cases} \quad (4)$$

where $\alpha_0 + \alpha_1 + \dots + \alpha_m = V$. Furthermore, some useful conclusions about time varying models concerning the DLT can be found in [112].

6.3. Memory Limitation in the Divisible Load Scheduling. Scheduling divisible load with limited memory buffers was

first analyzed in [35]. Subsequently, a multi-installment processing of divisible loads in systems with limited memory was analyzed [64]. Some details about divisible load with the memory model and hierarchical memory model can be found in [93, 100]. Furthermore, some algorithms and heuristics about divisible load with limited memory were investigated in [64].

6.4. Multi-Criteria Approach. The existing divisible load scheduling algorithms do not consider any priority for processors to allocate a fraction of the load. In some situation, the fractions of load must be allocated based on some certain priorities. For example, when we have some limitations in resources and so on, we need to allocate fractions of load based on some priority. In this case, a multi-criteria divisible load scheduling algorithm would be very useful. In [77] a multi-criteria divisible load model has been proposed for the first time. However, a closed form formula under multi-criteria has not appeared and it is still considered to be an open problem.

7. Algorithms

There are various algorithms for solving the divisible load scheduling problem. The algorithms have different behaviors. It is clear that using a suitable algorithm helps us to solve the problem without wasting time. Table 3 summarizes the algorithm applied for divisible load scheduling. In this section, we discuss the two main problems concerning the algorithms, including complexity and convergence.

7.1. Complexity. The complexity of divisible load algorithms has been investigated by several researchers [64, 76, 92, 100]. In a general form of divisible load scheduling the problem can be formulated as a linear programming problem. A simple form of linear programming can be solved by classical methods, such as the simplex method. Although the simplex method is a linear method in nature [131], a divisible load scheduling algorithm would be NP-hard even

TABLE 3: Comparing the algorithm applied for divisible load scheduling.

Algorithm	Advantages	Disadvantages	Approach	Reference
CDLT, IDLT, and TDP	Flexible	Complexity	Programming	[124]
Tractable recursive linear equation	Convergence	Complexity	Mathematical	[112, 131]
Linear programming	Convergence	NP-hard	Mathematical	[112]
Branch and bound	No limitation	NP-hard	Mathematical	[64]
Greedy algorithm	Certainty	Np-hard	Heuristics	[20]
Genetic algorithm	Fast	Divergence	Evolutionary heuristics	[68, 76]
Game theory	Truthful	Divergence	Heuristics	[62, 63]
Mixed integer linear programming	Certainty	Complexity	Mathematical	[76]

in the one-round case [76]. However, according to [76] multi-installment divisible load scheduling is also NP-hard. In fact, divisible load algorithms by considering some limitations, such as memory, have more complexity than a normal case. For instance, using the branch and bound method for solving multi-installment with limited memory would be an NP-hard problem [118]. As a result, the complexity is still one of the most important open issues of divisible load scheduling algorithms.

7.2. Convergence. Convergence is mainly considered in random heuristic based algorithms, for example, the genetic algorithm. However, the convergence issue in the genetic algorithm means that if too many operations are performed on a primary offspring, some offspring of the population cannot be produced. In general, a genetic algorithm based method may not be able to find optimum value in the DLT. There is a wide range of research that has been applied in the genetic algorithm concerning the DLT [68, 100, 125]. However, the existing research has not considered the convergence of the problem. Hence, it can be considered as an open problem in the area of genetic algorithms and divisible load scheduling.

8. Applications and Open Problem

This section focuses on applications and open problems in the field of divisible load theory.

8.1. Applications of Divisible Load Theory. A wide range of applications has been reported for the DLT in the literature. In the following list, we have addressed the most important applications of DLT according to the research: large size matrix-vector [27], image and vision processing [37], pipelined communication [87, 122], wireless sensor networks and processing measurement data [10, 11, 120], sequence alignment [59], video and multimedia applications [29, 81], computer vision [29], large scale data file processing [116], data intensive applications [78], query processing in database systems [22], monetary cost optimization [121], efficient movie retrieval for network-based multimedia [42],

grid computing [70, 75, 89], cloud computing [118], real-time computing [15, 39, 58, 61, 108, 110], flat file database [79], radar and infrared tracking [84], and multimedia and broadcasting [123]. More references for applications of divisible load scheduling can be found in [92, 105, 112].

8.2. Open Problems in Divisible Load Theory. The divisible load theory has high capabilities for application in homogeneous or heterogeneous environments. Our study shows that there are several important issues that still remain as open problems in the field of the divisible load theory. According to the literature, the following list refers to the main topics of open problems in the divisible load field study: reducing the number of variables in the linear programming model of divisible load [112]; analyzing the convergence of evolutionary algorithms such as genetic algorithms in divisible load [68, 100, 125]; investigating the complexity of algorithm [33, 76]; investigating the effects of cheat processors in the adaptive divisible load model [102, 103]; calculating a close form nonlinear divisible load model [53, 71]; applying the divisible load on the high performance computing, that is, cloud computing; investigating the effects of time varying problem on the divisible load model [16, 19, 92, 112]; investigating the relationship between the DLT and Markov chain [51, 118]; and investigating the closed form of the divisible load model with memory limitation [93, 100].

9. Comparison and Typology of the Existing Research

In this section, we have focused on the typology of the related research. According to ISI, IEEE, Springer, and Google Scholar about 200 scientific documents, including journal papers, conference papers, PhD theses, and books, are available in this field. However, we compared the number of citations based on the articles in [1–131]. The related documents cover over 5,812 citations from 1988 to 2013. We found that more than 20 subjects have the most citations in this field. We have classified these subjects into five main categories, including “development,” “strategies,” “approaches,” “applications,” and “others.” The five main groups and related subjects

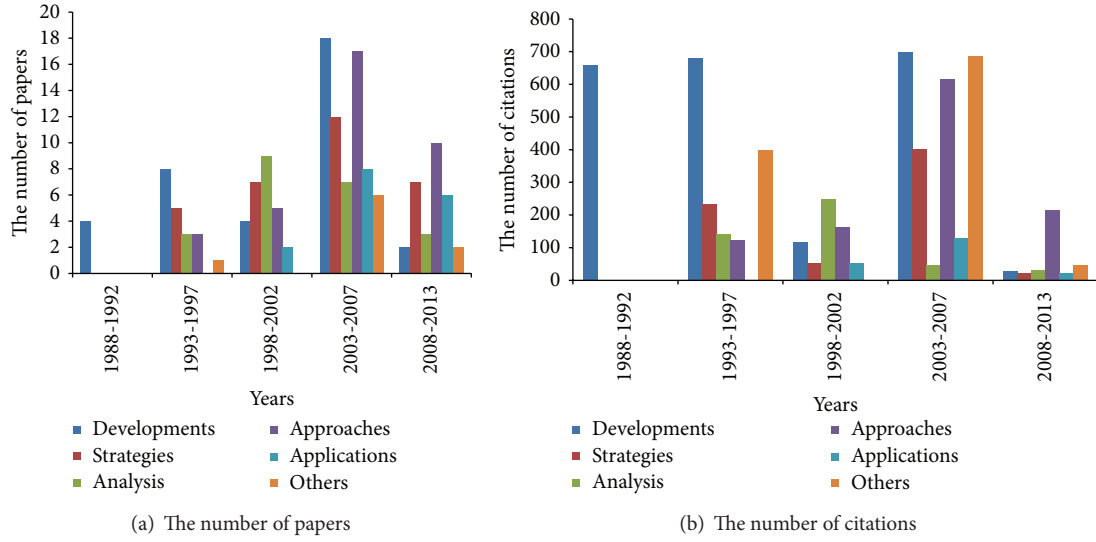


FIGURE 5: Comparison and typology of the researches.

TABLE 4: Typology of researches in divisible load scheduling.

Groups	Related topic (subjects)	The number of	
		papers	citations
Development	Foundations, closed form, topologies, ...	32	2181
Strategies	Multi-installment, multi-site, and adaptive	27	709
Applications	Image processing, grid computing, real time, ...	19	467
Approaches	Markov chain model, memory limitation, convergence, ...	30	1117
Analysis	Algorithm, complexity, ...	14	206
Others	Books, review papers, surveys, ...	9	1132

are listed in Table 4. We also compared the total numbers of papers based on the five research groups from 1988 to 2013. This is depicted in Figure 5(a). Moreover, the figure shows that “Development” has the highest number of papers from 1988 to 1997 and 2003 to 2007, while “Approaches” has the highest number of papers from 2008 to 2013. The figure also indicates that “Analysis” has the highest number of papers from 1998 to 2002. In addition, the figure illustrates that the highest number of papers was published between 2003 and 2007. Similarly, Figure 5(b) gives some information about the number of citations to the documents from 1988 to 2013. The figure indicates that the papers published from 1988 to 1997 and 2002 to 2007 in the category of “Developments” have the highest number of citations.

10. Conclusion

The divisible load theory has found a wide range of applications in the area of distributed computing. There are several intensive researches in the field of divisible load scheduling that employ various strategies in their approach to the problem. This paper has provided a comprehensive review of the research related to the divisible load theory. Furthermore,

we classified the papers according to the concepts, strategies, approaches, algorithms, and open problems. We also provided a list of applications and open problems in the related area. Finally, we compared the related research based on the typology of papers in the field of the divisible load theory.

Notation

V :	Total size of data
m :	Number of processors
α_i :	Size of load fraction allocated to processor p_i where $\sum_{i=0}^m \alpha_i = V$
p_i :	The i th processor
w_i :	The inverse of computing speed of processor p_i
z_i :	The inverse of transferring speed of processor p_i
w_i^{eq} :	Equivalent inverse processing speed of the i th subtree
T_{comp} :	Time of computation for processor p_i and equal to $\alpha_i w_i T_{\text{cp}}$
T_{comm} :	Time of communication for processor p_i and equal to $\alpha_i z_i T_{\text{cp}}$

T_i : Finish time for processor p_i and equal to $T_{\text{comm}} + T_{\text{comp}}$
 T : The time of processing all the data and equal to $\max\{T_0, T_1, \dots, T_m\}$.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been supported by the Malaysian Ministry of Education Fundamental Research Grant Scheme FRGS/02/01/12/1143/FR.

References

- [1] Y. Cheng and T. G. Robertazzi, "Distributed computation with communication delay," *IEEE Transactions Aerospace and Electronic Systems Magazine*, vol. 24, no. 6, pp. 700–712, 1988.
- [2] R. Agrawal and H. V. Jagadish, "Partitioning techniques for large-grained parallelism," *IEEE Transactions on Computers*, vol. 37, no. 12, pp. 1627–1634, 1988.
- [3] J. Sohn and T. G. Robertazzi, "Optimal load sharing for a divisible job on a bus network," in *Proceedings of the 1993 Conference on Information Sciences and Systems*, 1993.
- [4] T. G. Robertazzi, "Ten reasons to use divisible load theory," *Computer*, vol. 36, no. 5, pp. 63–68, 2003.
- [5] Y. Cheng and T. G. Robertazzi, "Distributed computation for a tree network with communication delays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 3, pp. 511–516, 1990.
- [6] T. G. Robertazzi, "Processor equivalence for daisy chain load sharing processors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pp. 1216–1221, 1993.
- [7] S. Bataineh, T. Hsiung, and T. G. Robertazzi, "Closed form solutions for bus and tree networks of processors load sharing a divisible job," *IEEE Transactions on Computers*, vol. 43, no. 10, pp. 1184–1196, 1994.
- [8] B. Veeravalli, D. Ghose, and V. Mani, "Multi-installment load distribution strategy in linear networks with communication delays," in *Proceedings of the 1st International Workshop on Parallel Processing (IWPP'94)*, pp. 563–568, Bangalore, India, December 1994.
- [9] J. Sohn and T. G. Robertazzi, *A Multi-Job Load Sharing Strategy for Divisible Jobs on Bus Networks*, College of Engineering, State University of New York at Stony Brook, New York, NY, USA, 1994.
- [10] H. Shi, W. Wang, N. M. Kwok, and S. Chen, "Adaptive indexed divisible load theory for wireless sensor network workload allocation," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 484796, 18 pages, 2013.
- [11] H. Shi, W. Wang, and N. Kwok, "Energy dependent divisible load theory for wireless sensor network workload allocation," *Mathematical Problems in Engineering*, vol. 2012, Article ID 235289, 16 pages, 2012.
- [12] C. Lee and M. Hamdi, "Parallel image processing applications on a network of workstations," *Parallel Computing*, vol. 21, no. 1, pp. 137–160, 1995.
- [13] B. Veeravalli, D. Ghose, and V. Mani, "Multi-installment load distribution in tree networks with delays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 2, pp. 555–567, 1995.
- [14] J. Blazewicz and M. Drozdowski, "Scheduling divisible jobson hypercubes," *Parallel Computing: Theory and Applications*, vol. 21, no. 12, pp. 1945–1956, 1995.
- [15] H. Emile, "Runtime reallocation of divisible load under processor execution deadlines," in *Proceedings of the Third IEEE Workshop on Parallel and Distributed Real-Time Systems*, 1995.
- [16] J. Sohn and T. G. Robertazzi, *An Optimum Load Sharing Strategy for Divisible Jobs with Time-Varying Processor and Channel Speed*, State University of New York at Stony Brook, College of Engineering, Stony Brook, NY, USA, 1995.
- [17] B. Veeravalli, *Scheduling Divisible Loads in Parallel and Distributed Systems*, vol. 8, John Wiley & Sons, 1996.
- [18] B. Veeravalli, H. F. Li, and T. Radhakrishnan, "Scheduling divisible loads in bus networks with arbitrary processor release times," *Computers & Mathematics with Applications*, vol. 32, no. 7, pp. 57–77, 1996.
- [19] J. Sohn and T. G. Robertazzi, "Optimal time-varying load sharing for divisible loads," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 907–923, 1998.
- [20] W. Glazek, "A greedy algorithm for processing a divisible load on a hypercube," in *Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC'98)*, pp. 185–188, Bialystok, Poland, September 1998.
- [21] B. Veeravalli, X. Li, and C. C. Ko, "On the influence of start-up costs in scheduling divisible loads on bus networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 12, pp. 1288–1305, 2000.
- [22] K. Ko and T. G. Robertazzi, "Record search time evaluation using divisible load analysis," Tech. Rep. 765, SUNY at Stony Brook College of Engineering and Applied Science, 1998.
- [23] J. Błazewicz, M. Drozdowski, and M. Markiewicz, "Divisible task scheduling—concept and verification," *Parallel Computing*, vol. 25, no. 1, pp. 87–98, 1999.
- [24] J. Blazewicz, M. Drozdowski, F. Guinand, and D. Trystram, "Scheduling a divisible task in a two-dimensional toroidal mesh," *Discrete Applied Mathematics*, vol. 94, no. 1–3, pp. 35–50, 1999.
- [25] M. Drozdowski and W. Glazek, "Scheduling divisible loads in a three-dimensional mesh of processors," *Parallel Computing: Theory and Applications*, vol. 25, no. 4, pp. 381–404, 1999.
- [26] B. Veeravalli and N. Viswanatham, "Sub-optimal solutions using integer approximation techniques for scheduling divisible loads on distributed Bus Networks," Tech. Rep. VB/DLT/002/1999, Department of Electrical and Computer Engineering, The National University of Singapore, Singapore, January 1999.
- [27] S. K. Chan, B. Veeravalli, and D. Ghose, "Experimental study on large size matrix-vector product computations using divisible load paradigm on distributed bus networks," Tech. Rep. VB/DLT/003/1999, Department of Electrical and Computer Engineering, Singapore, Singapore, 1999.
- [28] B. Veeravalli, X. Li, and C. C. Ko, "Design and analysis of load distribution strategies with start-up costs in scheduling divisible loads on distributed networks," *Mathematical and Computer Modelling*, vol. 32, no. 7–8, pp. 901–932, 2000.
- [29] B. Veeravalli, X. Li, and C. C. Ko, "Efficient partitioning and scheduling of computer vision and image processing data on

- bus networks using divisible load analysis," *Image and Vision Computing*, vol. 18, no. 11, pp. 919–938, 2000.
- [30] X. Li, B. Veeravalli, and C. C. Ko, "Divisible load scheduling on single-level tree networks with buffer constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 4, pp. 1298–1308, 2000.
- [31] B. Veeravalli and N. Viswanadham, "Suboptimal solutions using integer approximation techniques for scheduling divisible loads on distributed bus networks," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 30, no. 6, pp. 680–691, 2000.
- [32] X. Li, B. Veeravalli, and C. C. Ko, "Divisible load scheduling on a hypercube cluster with finite-size buffers and granularity constraints," in *Proceedings 1st IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2001.
- [33] M. Drozdowski and P. Wolniewicz, "On the complexity of divisible job scheduling," Tech. Rep. RA-001/2001, Poznan University of Technology, 2001.
- [34] X. Li, *Studies on divisible load scheduling strategies in distributed computing systems: design, analysis and experiment [Ph.D. thesis]*, National University of Singapore, Singapore, 2001.
- [35] P. Wolniewicz and M. Drozdowski, "Processing time and memory requirements for multi-installment divisible job processing," in *Proceedings of the 4th International Conference on Parallel Processing and Applied Mathematics (PPAM '01)*, vol. 2328 of *Lecture Notes in Computer Science*, pp. 125–133, 2002.
- [36] B. Veeravalli and G. Barlas, "Efficient scheduling strategies for processing multiple divisible loads on bus networks," *Journal of Parallel and Distributed Computing*, vol. 62, no. 1, pp. 132–151, 2002.
- [37] B. Veeravalli and S. Ranganath, "Theoretical and experimental study on large size image processing applications using divisible load paradigm on distributed bus networks," *Image and Vision Computing*, vol. 20, no. 13-14, pp. 917–935, 2002.
- [38] P. Wolniewicz, "Multi-installment divisible job processing with communication start up cost," *Foundations of Computing and Decision Sciences*, vol. 27, no. 1, pp. 43–57, 2002.
- [39] B. Veeravalli and W. H. Min, "Scheduling divisible loads on heterogeneous linear daisy chain networks with arbitrary processor release times," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 3, pp. 273–288, 2004.
- [40] B. Veeravalli, "Design, analysis, and simulation of multiple divisible load scheduling strategies in distributed computing networks," Tech. Rep. RP-263-000-073-12, Open Source Software Laboratory, Department of Electrical and Computer Engineering, The National University of Singapore, Singapore, 2002.
- [41] O. Beaumont, A. Legrand, and Y. Robert, "Scheduling divisible workloads on heterogeneous platforms," *Parallel Computing: Theory and Applications*, vol. 29, no. 9, pp. 1121–1152, 2003.
- [42] B. Veeravalli, D. Ghose, and T. G. Robertazzi, "Divisible load theory: a new paradigm for load scheduling in distributed systems," *Cluster Computing*, vol. 6, no. 1, pp. 7–17, 2003.
- [43] D. Ghose and T. G. Robertazzi, "Foreword (special issue of cluster computing on divisible load scheduling)," *Cluster Computing*, vol. 6, no. 1, p. 5, 2003.
- [44] H.-J. Kim, "A novel optimal load distribution algorithm for divisible loads," *Cluster Computing*, vol. 6, no. 1, pp. 41–46, 2003.
- [45] K. Li, "Parallel processing of divisible loads on partition-able static interconnection networks," *Cluster Computing*, vol. 6, no. 1, pp. 47–55, 2003.
- [46] V. Mani, "An equivalent tree network methodology for efficient utilization of front-ends in linear network," *Cluster Computing*, vol. 6, no. 1, pp. 57–62, 2003.
- [47] B. Veeravalli and B. Gerassimos, "Scheduling divisible loads with processor release times and finite size buffer capacity constraints in bus networks," *Cluster Computing*, vol. 6, no. 1, pp. 63–74, 2003.
- [48] X. Li, B. Veeravalli, and C. C. Ko, "Distributed image processing on a network of workstations," *International Journal of Computers and Applications*, vol. 25, no. 2, pp. 1–10, 2003.
- [49] K. Li, "Speed-up of parallel processing of divisible loads on k-dimensional meshes and tori," *The Computer Journal*, vol. 46, no. 6, pp. 625–631, 2003.
- [50] K. Li, "Improved methods for divisible load distribution on k-dimensional meshes using pipelined communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 12, pp. 1250–1261, 2003.
- [51] M. Moges and T. G. Robertazzi, "Optimal divisible load scheduling and Markov chain models," in *Proceedings of the Conference on Information Sciences and Systems*, 2003.
- [52] H. M. Wong, B. Veeravalli, D. Yu, and T. G. Robertazzi, "Data intensive grid scheduling: multiple sources with capacity constraints," in *Proceedings of the 15th IASTED International Conference on Parallel and Distributed Computing and Systems*, pp. 7–11, November 2003.
- [53] J. T. Hung and T. G. Robertazzi, "Scheduling nonlinear computational loads," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1169–1182, 2008.
- [54] Y. Yang and H. Casanova, "Rumr: robust scheduling for divisible workloads," in *Proceedings of 12th IEEE International Symposium on High Performance Distributed Computing*, 2003.
- [55] D. Ghose, H. J. Kim, and T. H. Kim, "Adaptive divisible load scheduling strategies for workstation clusters with unknown network resources," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, pp. 897–907, 2005.
- [56] P. Wolniewicz, *Divisible job scheduling in systems with limited memory [Ph.D. thesis]*, Poznan University of Technology, Poznań, Poland, 2003.
- [57] C. Yu and D. C. Marinescu, "Algorithms for divisible load scheduling of data-intensive applications," *Journal of Grid Computing*, vol. 8, no. 1, pp. 133–155, 2010.
- [58] X. Lin, Y. Lu, J. Deogun, and S. Goddard, "Enhanced real-time divisible load scheduling with different processor available times," in *High Performance Computing—HiPC 2007*, vol. 4873 of *Lecture Notes in Computer Science*, pp. 308–319, Springer, Berlin, Germany, 2007.
- [59] W. H. Min and B. Veeravalli, "Aligning biological sequences on distributed bus networks: a divisible load scheduling approach," *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 4, pp. 489–501, 2005.
- [60] A. Shokripour, M. Othman, H. Ibrahim, and S. Subramaniam, "New method for scheduling heterogeneous multi-installment systems," *Future Generation Computer Systems*, vol. 28, no. 8, pp. 1205–1216, 2012.
- [61] A. Mamat, Y. Lu, J. Deogun, and S. Goddard, "Scheduling real-time divisible loads with advance reservations," *Real-Time Systems*, vol. 48, no. 3, pp. 264–293, 2012.
- [62] T. E. Carroll and D. Grosu, "An incentive-based distributed mechanism for scheduling divisible loads in tree networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 3, pp. 389–401, 2012.

- [63] T. E. Carroll and D. Grosu, "A strategy proof mechanism for scheduling divisible loads in bus networks without control processors," in *Proceedings of the 20th International IEEE Parallel and Distributed Processing Symposium*, 2006.
- [64] J. Berlińska and M. Drozdowski, "Heuristics for multi-round divisible loads scheduling with limited memory," *Parallel Computing*, vol. 36, no. 4, pp. 199–211, 2010.
- [65] M. Othman, M. Abdullah, H. Ibrahim, and S. Subramaniam, "Adaptive divisible load model for scheduling data-intensive grid applications," in *Computational Science—ICCS 2007*, vol. 4487, pp. 446–453, Springer, Berlin, Germany, 2007.
- [66] M. Abdullah, M. Othman, H. Ibrahim, and S. Subramaniam, "An integrated approach for scheduling divisible load on large scale data grids," in *Computational Science and Its Applications—ICCSA 2007*, vol. 4705 of *Lecture Notes in Computer Science*, pp. 748–757, Springer, Berlin, Germany, 2007.
- [67] X. Li and B. Veeravalli, "PPDD: scheduling multi-site divisible loads in single-level tree networks," *Cluster Computing*, vol. 13, no. 1, pp. 31–46, 2010.
- [68] M. Abdullah, M. Othman, H. Ibrahim, and S. Subramaniam, "Optimal workload allocation model for scheduling divisible data grid applications," *Future Generation Computer Systems*, vol. 26, no. 7, pp. 971–978, 2010.
- [69] J. Berlińska, "Fully polynomial time approximation schemes for scheduling divisible loads," in *Parallel Processing and Applied Mathematics*, pp. 1–10, Springer, Berlin, Germany, 2010.
- [70] M. A. Moges, D. Yu, and T. G. Robertazzi, "Grid scheduling divisible loads from two sources," *Computers & Mathematics with Applications*, vol. 58, no. 6, pp. 1081–1092, 2009.
- [71] S. Suresh, H. J. Kim, C. Run, and T. G. Robertazzi, "Scheduling nonlinear divisible loads in a single level tree network," *The Journal of Supercomputing*, vol. 61, no. 3, pp. 1068–1088, 2012.
- [72] T. E. Carroll and D. Grosu, "Divisible load scheduling: an approach using coalitional games," in *Proceedings of the 6th International Symposium on Parallel and Distributed Computing (ISPD'07)*, July 2007.
- [73] D. Ghose, "A feedback strategy for load allocation in workstation clusters with unknown network resource capabilities using the DLT paradigm," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, vol. 1, CSREA Press, 2002.
- [74] W. H. Min, B. Veeravalli, and G. Barlas, "Design and performance evaluation of load distribution strategies for multiple divisible loads on heterogeneous linear daisy chain networks," *Journal of Parallel and Distributed Computing*, vol. 65, no. 12, pp. 1558–1577, 2005.
- [75] T. G. Robertazzi and D. Yu, "Multi-source grid scheduling for divisible loads," in *Proceedings of the 40th Annual Conference on Information Sciences and Systems (CISS'06)*, pp. 188–191, Princeton, NJ, USA, March 2006.
- [76] Y. Yang, H. Casanova, M. Drozdowski, M. Lawenda, and A. Arnaud, *On the Complexity of Multi-Round Divisible Load Scheduling*, INRIA, 2007, <http://hal.inria.fr/inria-00123711>.
- [77] S. Ghanbari, M. Othman, W. J. Leong, and M. R. Abu Bakar, "Multi-criteria based algorithm for scheduling divisible load," in *Proceedings of the 1st International Conference on Advanced Data and Information Engineering (DaEng '13)*, Lecture Notes in Electrical Engineering, pp. 547–554, Springer, Singapore, 2014.
- [78] K. Ko and T. G. Robertazzi, "Equal allocation scheduling for data intensive applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 695–705, 2004.
- [79] K. Ko and T. G. Robertazzi, "Signature search time evaluation in flat file databases," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 493–502, 2008.
- [80] S. K. Chan, B. Veeravalli, and D. Ghose, "Large matrix-vector products on distributed bus networks with communication delays using the divisible load paradigm: performance analysis and simulation," *Mathematics and Computers in Simulation*, vol. 58, no. 1, pp. 71–92, 2001.
- [81] P. Li, B. Veeravalli, and A. A. Kassim, "Design and implementation of parallel video encoding strategies using divisible load analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 9, pp. 1098–1112, 2005.
- [82] S. Suresh, S. N. Omkar, and V. Mani, "Parallel implementation of back-propagation algorithm in networks of workstations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 1, pp. 24–34, 2005.
- [83] H.-J. Kim, "A novel optimal load distribution algorithm for divisible loads," *Cluster Computing*, vol. 6, no. 1, pp. 41–46, 2003.
- [84] J. T. Hung and T. G. Robertazzi, "Divisible load cut through switching in sequential tree networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pp. 968–982, 2004.
- [85] M. Moges and T. G. Robertazzi, "Wireless sensor networks: Scheduling for measurement and data reporting," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 327–340, 2006.
- [86] B. Veeravalli, *Scheduling Divisible Loads in Parallel and Distributed Systems*, vol. 8, John Wiley & Sons, New York, NY, USA, 1996.
- [87] K. Li, "New divisible load distribution methods using pipelined communication techniques on tree and pyramid networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 806–819, 2011.
- [88] O. Beaumont, H. Casanova, A. Legrand, Y. Robert, and Y. Yang, "Scheduling divisible loads on star and tree networks: Results and open problems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 3, pp. 207–218, 2005.
- [89] T. Zhu, Y. Wu, and G. Yang, "Scheduling divisible loads in the dynamic heterogeneous grid environment," in *Proceedings of the 1st ACM International Conference on Scalable Information Systems (InfoScale '06)*, Hong Kong, June 2006.
- [90] Y. C. Cheng and T. G. Robertazzi, "Distributed computation with communication delay," *IEEE Aerospace and Electronic Systems Magazine*, vol. 24, no. 6, pp. 700–712, 1988.
- [91] J. Baewicz and M. Drozdowski, "The performance limits of a two dimensional network of load-sharing processors," *Foundations of Computing and Decision Sciences*, vol. 21, no. 1, pp. 3–15, 1996.
- [92] M. Drozdowski, *Scheduling for Parallel Processing*, Springer, 2009.
- [93] M. Drozdowski and M. Lawenda, "Multi-installment divisible load processing in heterogeneous systems with limited memory," in *Parallel Processing and Applied Mathematics*, pp. 847–854, Springer, Berlin, Germany, 2006.
- [94] B. Veeravalli, D. Ghose, and V. Mani, "Optimal sequencing and arrangement in distributed single-level tree networks with communication delays," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 9, pp. 968–976, 1994.
- [95] Y.-K. Chang, J.-H. Wu, C.-Y. Chen, and C.-P. Chu, "Improved methods for divisible load distribution on k-dimensional meshes using multi-installment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 11, pp. 1618–1629, 2007.

- [96] S. Suresh, V. Mani, and S. N. Omkar, "The effect of start-up delays in scheduling divisible loads on bus networks: an alternate approach," *Computers & Mathematics with Applications*, vol. 46, no. 10-11, pp. 1545-1557, 2003.
- [97] H. J. Kim and V. Mani, "Divisible load scheduling in single-level tree networks: optimal sequencing and arrangement in the non-blocking mode of communication," *Computers & Mathematics with Applications*, vol. 46, no. 10-11, pp. 1611-1623, 2003.
- [98] B. Veeravalli and J. Yao, "Divisible load scheduling strategies on distributed multi-level tree networks with communication delays and buffer constraints," *Computer Communications*, vol. 27, no. 1, pp. 93-110, 2004.
- [99] J. Yao and B. Veeravalli, "Design and performance analysis of divisible load scheduling strategies on arbitrary graphs," *Cluster Computing*, vol. 7, no. 2, pp. 191-207, 2004.
- [100] M. Drozdowski and P. Wolniewicz, "Optimum divisible load scheduling on heterogeneous stars with limited memory," *European Journal of Operational Research*, vol. 172, no. 2, pp. 545-559, 2006.
- [101] D. Grosu and T. E. Carroll, "A strategyproof mechanism for scheduling divisible loads in distributed systems," in *Proceedings of the 4th International Symposium on Parallel and Distributed Computing (ISPD '05)*, pp. 83-90, July 2005.
- [102] T. E. Carroll and D. Grosu, "A strategy proof mechanism for scheduling divisible loads in bus networks without control processors," in *Proceedings of the 20th IEEE International Symposium on Parallel and Distributed Processing*, 2006.
- [103] T. E. Carroll and D. Grosu, "A strategyproof mechanism for scheduling divisible loads in linear networks," in *Proceedings of the 21st International Parallel and Distributed Processing Symposium (IPDPS '07)*, March 2007.
- [104] M. Abdullah, M. Othman, H. Ibrahim, and S. Subramaniam, "An integrated approach for scheduling divisible load on large scale data grids," in *Computational Science and Its Applications—ICCSA*, vol. 4705, pp. 748-757, Springer, Berlin, Germany, 2007.
- [105] A. Shokripour and M. Othman, "Survey on divisible load theory," in *Proceedings of the IEEE International Association of Computer Science and Information Technology—Spring Conference (IACSITSC '09)*, pp. 9-13, Singapore, 2009.
- [106] A. Shokripour and M. Othman, "Categorizing DLT researches and its applications," *European Journal of Scientific Research*, vol. 37, no. 3, pp. 496-515, 2009.
- [107] J. Jia, B. Veeravalli, and J. Weissman, "Scheduling multisource divisible loads on arbitrary networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 4, pp. 520-531, 2010.
- [108] A. Mamat, Y. Lu, J. Deogun, and S. Goddard, "Scheduling real-time divisible loads with advance reservations," *Real-Time Systems*, vol. 48, no. 3, pp. 264-293, 2012.
- [109] X. Lin, A. Mamat, Y. Lu, J. Deogun, and S. Goddard, "Real-time scheduling of divisible loads in cluster computing environments," *Journal of Parallel and Distributed Computing*, vol. 70, no. 3, pp. 296-308, 2010.
- [110] S. Chuprat, "Divisible Load scheduling of real-time task on heterogeneous clusters," in *Proceedings of the IEEE International Symposium on Information Technology (ITSim '10)*, vol. 2, pp. 721-726, Kuala Lumpur, Malaysia, June 2010.
- [111] A. Ghatpande, H. Nakazato, H. Watanabe, and O. Beaumont, "Divisible load scheduling with result collection on heterogeneous systems," in *Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS '08)*, pp. 1-8, April 2008.
- [112] T. G. Robertazzi, *Networks and Grids: Technology and Theory*, Springer, New York, NY, USA, 2007.
- [113] O. Beaumont, H. Casanova, A. Legrand, Y. Robert, and Y. Yang, "Scheduling divisible loads on star and tree networks: results and open problems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 3, pp. 207-218, 2005.
- [114] M. A. Moges, D. Yu, and T. G. Robertazzi, "Grid scheduling divisible loads from multiple sources via linear programming," in *Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems (PDOS '04)*, pp. 548-553, Cambridge, Mass, USA, November 2004.
- [115] T. G. Robertazzi, "Divisible load scheduling with multiple sources: closed form solutions," in *Proceedings of the Conference on Information Sciences and Systems*, 2005.
- [116] L. Marchal, Y. Yang, H. Casanova, and Y. Robert, "A realistic network/application model for scheduling divisible loads on large-scale platforms," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, April 2005.
- [117] M. Abdullah and M. Othman, "An improved genetic algorithm for job scheduling in cloud computing environment," *Procedia Information Technology and Computer Science*, vol. 2, 2013.
- [118] M. A. Moges and T. G. Robertazzi, "Divisible load scheduling and Markov chain models," *Computers & Mathematics with Applications*, vol. 52, no. 10-11, pp. 1529-1542, 2006.
- [119] D. Ghose, "A feedback strategy for load allocation in workstation clusters with unknown network resource capabilities using the DLT paradigm," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, vol. 1, pp. 425-428, CSREA Press, 2002.
- [120] M. A. Moges and T. G. Robertazzi, "Wireless sensor networks: scheduling for measurement and data reporting," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 327-340, 2006.
- [121] J. Sohn, T. G. Robertazzi, and S. Luryi, "Optimizing computing costs using divisible load analysis," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 3, pp. 225-234, 1998.
- [122] K. Li, "Accelerating divisible load distribution on tree and pyramid networks using pipelined communications," in *Proceedings 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 3131-3138, Santa Fe, New Mexico, April 2004.
- [123] B. Veeravalli and G. Barlas, "Access time minimization for distributed multimedia applications," *Multimedia Tools and Applications*, vol. 12, no. 2-3, pp. 235-256, 2000.
- [124] K. Ko and T. G. Robertazzi, "Scheduling in an environment of multiple job submissions," in *Proceedings of the 2002 Conference on Information Sciences and Systems*, January 2002.
- [125] S. Kim and J. B. Weissman, "A genetic algorithm based approach for scheduling decomposable Data Grid applications," in *Proceedings of the International Conference on Parallel Processing (ICPP '04)*, pp. 406-413, August 2004.
- [126] J. Berlińska and M. Drozdowski, "Scheduling divisible MapReduce computations," *Journal of Parallel and Distributed Computing*, vol. 71, no. 3, pp. 450-459, 2011.
- [127] M. Drozdowski, M. Lawenda, and F. Guinand, "Scheduling multiple divisible loads," *International Journal of High Performance Computing Applications*, vol. 20, no. 1, pp. 19-30, 2006.
- [128] M. Drozdowski and M. Lawenda, "Scheduling multiple divisible loads in homogeneous star systems," *Journal of Scheduling*, vol. 11, no. 5, pp. 347-356, 2008.

- [129] A. Ghatpande, H. Nakazato, O. Beaumont, and H. Watanabe, "SPORT: An algorithm for Divisible Load Scheduling with result collection on heterogeneous Systems," *IEICE Transactions on Communications*, vol. 91, no. 8, pp. 2571–2588, 2008.
- [130] M. Drozdowski and P. Wolniewicz, "Performance limits of divisible load processing in systems with limited communication buffers," *Journal of Parallel and Distributed Computing*, vol. 64, no. 8, pp. 960–973, 2004.
- [131] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, McGraw-Hill, New York, NY, USA, 8th edition, 2005.

