A Variation-Aware Approach for Task Allocation in Wireless Distributed Computing Systems

Xiaofu Ma*1, Haris I. Volos†2, Xiangwei Zheng*3, Jeffrey H. Reed*4, and Tamal Bose†5

* *Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, Virginia 24061, USA*
{xfma1, xiangwei3, reedjh4}@vt.edu

† *Department of Electrical and Computer Engineering, The University of Arizona, Tucson, Arizona 85721, USA*
{hvolos2, tbose5}@arizona.edu

***Abstract*—Wireless distributed computing (WDC) enables the radio nodes with reduced computing abilities to cooperate in processing complex computational tasks for minimizing the overall processing latency (makespan). However, the uncertainty of the dynamic mobile wireless environment, which is not an issue for the traditional distributed computing, poses a challenge for WDC. In this paper, a variation-aware approach for WDC is proposed to determine the task allocation by considering the heterogeneous computing capability of the radio nodes as well as the impact of the radio environment. To this end, the transmission latency is characterized as a random variable that depends on the channel fading and the transport protocol. A variation-aware task graph analysis is proposed for the estimation of the makespan's distribution. The evolutionary algorithms are employed for the allocation mapping. We use simulation results to affirm makespan estimation improvement of the proposed approach compared with the traditional deterministic approach, and give insights on the dominating factors for the improvement[1].**

I. Introduction

Wireless distributed computing (WDC) systems will become the next frontier of communication and computation research [1]. The goal of WDC is to exploit the computation capability of distributed computing systems in a cable-free environment, which will bring a new level of flexibility for users who need to work in rapidly changing wireless and mobile environments [2]. WDC can fit well into software defined radio networks, cognitive radio networks, wireless sensor networks, smart phones, wireless cloud computing, wireless grids, and ubiquitous mobile computing systems [3]. In cognitive radio networks, for instance, the statistics of the channel conditions sensed and learned by the radio nodes can be properly exploited by WDC for better execution performance of computational tasks.

The key objective of distributed computing is to divide a complex computational task into subtasks, and to assign each subtask to one or more computing devices. The task allocation strategies are made based on the computation and transmission performance of each computing device. For traditional distributed computing, thanks to the high speed and stability nature of the wired communication links among the computing devices, the fluctuation of the transmission can be considered negligible for the makespan of task assignment. However, the radio environment, which is characterized by uncertainty, high outage probabilities and bit-error-rates, and random delays due to retransmissions [2], makes it difficult to design task allocation strategies for WDC systems. Because of the uncertainty of the channel, the variation of packet retransmissions and the uncertainty of the node failures, the transmission latency cannot be considered as a constant number.

Addressing these challenges, we focus on the challenges arising due to the effect of the wireless channel, take the statistic information of the wireless channel into consideration for task allocation and provide not only the makespan of task allocation, but also the confidence estimation of the result. A variation-aware approach for task allocation in WDC system is proposed with the consideration of channel fading and the transport protocol, stop-and-wait automatic repeat request (ARQ). The major contributions of this work are:

1. The WDC scheduling issue is formulated based on the variation-aware approach.

2. The impact of the channel on the transmission latency of different modulation schemes and the transport layer protocol is analyzed to obtain the probability distribution function (PDF) of the transmission latency.

3. Variation-aware task graph analysis is used to get the PDF of the makespan.

4. An evaluation metric of the makespan PDF is proposed for wireless distributed computing.

5. Evolutionary algorithms are used for calculating the allocation mapping. The proposed approach's improvement vs. the traditional deterministic worst case analysis is evaluated.

The remainder of the paper is organized as follows: Section II presents an overview of related research and our motivation. In Section III, we formulate the task allocation and scheduling problem based on the variation-aware approach. In Section IV, we analyze the impact of wireless channel on the transmission latency of different modulation schemes and stop-and-wait ARQ protocol. In Section V, variation-aware task graph is analyzed. The evolutionary-algorithm-based methods for task allocation are described in Section VI. Performance evaluation results are presented in Section VII. Finally, conclusions are presented in Section VIII.

II. Related Work

The task allocation issue in distributed computing has been found to be an NP-hard problem [4]. Generally, there are two main categories among all its solutions. One of them is the set of mathematical programming approaches which include the graph theoretic method [5] and the branch-and-bound method [6]. The mathematical programming approaches

[1]This project was funded in part by the affiliates of Wireless@Virginia Tech and Broadband Wireless Access & Applications Center (BWAC).

use systematical mathematic tools to find the optimal solution. Those approaches are seeking the exact solutions which make them computationally intensive, if the problem size is large. Only for some specific task graphs and communication graphs, the optimal solution can be found in polynomial time. The other category is to use a heuristic approach. Heuristic approaches cannot guarantee the optimal solution, but they are likely to lead to a solution that is approximately optimal in a reasonable time. The typical heuristic methods used for task allocation problem are the genetic algorithm [7], the differential evolution algorithm [8], and heterogeneous earliest-finish-time [9]. However, due to the unique challenges of wireless systems, the task allocation approaches for traditional distributed computing cannot directly work for WDC systems.

For wireless systems, the current work can be generally classified as task allocation problems and workload distribution problems, which assume all the tasks are independent and each node executes only one task at a time.

For task allocation, the authors of [10] considered the energy and time costs of both computation and communication activities for single-hop cluster of homogeneous sensor nodes. The objective of [10] is to maximize the lifetime of the sensor network. In [11], the authors studied the task allocation issues in multi-hop wireless network. Genetic algorithm is used to perform task allocation and balance the energy consumption over collaborative nodes in order to extend the overall network lifetime. Article [12] presents a task allocation algorithm that aims to balance the energy consumption and makespan in a heuristic manner. This algorithm's objective is computed as the weighted sum of the energy consumption and latency. However, the impact of channel variations on execution and transmission latency is not considered in the above analyses.

The impact of channel variations on wireless distributed computing systems is considered in [1] for workload distribution. The authors derive the computation latency in terms of the channel variation and find that the latency performance is influenced by both the average channel condition and the variation of channels. Article [13] provides the power consumption models for the computation and communication subsystems. The authors present a fundamental power efficiency analysis of WDC for uniform workload distribution. However, those efforts cannot apply for the task allocation problems which consider the dependency constraints among the tasks.

In this paper, our focus is the task allocation approach in WDC systems. One major consideration is the probability distribution of the transmission latency. If the PDFs of the transmission latency among the links are known by cognitive radios, the analysis for task allocation is not only based on the operation of constant numbers, but also the PDF operations. Therefore, a variation-aware approach is preferable in WDC systems.

Parameter variation problems also exist in the field of multiprocessor system-on-chip (MPSoC). A heuristic latency variation-aware task allocation and scheduling algorithm is proposed for MPSoC in [14], and the power variation for task allocation in MPSoC is considered in [15]. The computing latency in MPSoC systems is the key factor, but the transmission latency is ignored. This is because all the processors share the memory on the same chip. However, the transmission latency is the major concern of WDC systems. Thus, the approaches for MPSoC cannot apply for wireless system directly. In addition, the probability distributions of the parameters in those two systems are not the same.

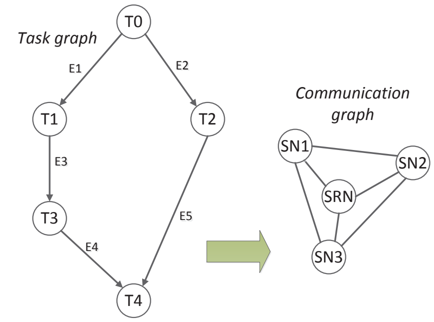


Fig. 1: A directed acyclic task graph and a communication graph in the WDC system

To the best of our knowledge, there's currently no framework to analyze the variation of the transmission latency for task allocation in WDC system. Thus, our goal is to present a paradigm shift from the deterministic approach to the variation-aware approach, which is preferable in the wireless environment.

III. Formulation of Variation-aware Approach

In this section, we first describe the network and radio node configuration considered in the variation-aware approach, and then we present the problem formulation.

A. Network and Node Configuration

In the WDC system, as shown in Fig. 1, the computing resources are notated as service nodes (SNs), which are interconnected via a mesh network to perform distributed computing. The service request node (SRN) is in charge of task allocation and access control. The SNs are heterogeneous and capacitated with various units of memory and processing resources. That is to say, if the same task is executed on different SNs, the computation latency may be different. Data transmission between successive tasks on different SNs causes the transmission latency, which is a random variable with a PDF due to the impact of channel variations.

The WDC application is decomposed into M dependent computational tasks, which can be represented as a directed acyclic task graph (DAG), $G = (T, E)$. The vertices represent the tasks, and the edges together with the weights represent the dependencies between the successive tasks and the data transmission between them. In this work, for a particular allocation decision of the tasks, each tasks computation latency on the allocated SN can be known as a fixed number, and the transmission latency between SNs with successive tasks can be known as a variable with a fixed PDF from cognitive radios. The PDF of the transmission latency in the wireless environment will be discussed in the next section. The notations that are used for the problem formulation are listed in Table I.

In this paper, we assume: (a) the frequency channels used for different links are orthogonal; (b) the packet sizes are fixed for all the data transmissions; (c) TDMA is used for media access control (MAC); (d) the bandwidth for each channel is fixed; (e) only one channel can be used for one communication activity of each link; and (f) data transmission latency within one service node is negligible.

TABLE I: Symbols Used for Problem Formulation

Symbol	Description
T_i	*Task i in DAG*
(T_i, T_k)	*Edge in DAG with the starting vertex T_i and ending vertex T_k*
M	*Number of Tasks in DAG*
$Cmp[i]$	*Amount of Calculation needed for T_i*
$Prec(i)$	*Set of predecessors for T_i*
$Succ(i)$	*Set of successors for T_i*
$Trans[M, M]$	*Transmission Latency Variable matrix for M Tasks*
N_j	*SN_j (service node j)*
N	*Number of SNs*
$S[j]$	*Computing Speed of N_j*
$ECT[i, j]$	*Estimated computation time for T_i executed on N_j*
MS	*Makespan, finishing time for all tasks*

B. Problem Formulation

Given (1) a wireless distributed computing network which comprises heterogeneous service nodes and (2) a DAG for a WDC application in which the tasks are executed in the wireless network, determine the task allocation strategy that maps the tasks to the service nodes, and find the scheduling strategy that minimizes the makespan metric while satisfying (1) task dependency and (2) resource availability.

The constrains for a feasible solution also include: (1) each task is only executed on one SN; (2) no SNs have multiple processor, and no multi-threading technique is used, i.e. only one task can be executed on one SN at one time.

C. Benefit of Variation-aware Approach

The traditional approach takes the deterministic worst case time for the transmission latency. We call them the deterministic worst-case analysis (DWCA) in this paper. People may naturally think of converting this variation-aware task allocation problem into a deterministic one by using the worst-case analysis. In this way, the conventional task allocation approaches can be applied. However, an overly pessimistic estimation of the transmission latency may result from DWCA. For example, consider two variables T_1 and T_2 with the normal distributions $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$. The worst-case time is calculated as the time value associated with $\mu + n\sigma$ (typical value of n is 3). The summation of the two latencies using DWCA equals to $\mu_1 + \mu_2 + n\sigma_1 + n\sigma_2$, which is the summation of the two worst-case time. Meanwhile, for the variation aware (VA) approach, $SUM\,(T_1, T_2)$ follows $N\left(\mu_1 + \mu_2, \sqrt{{\sigma_1}^2 + {\sigma_2}^2}\right)$, and thus the corresponding worst-case time is $\mu_1 + \mu_2 + n\sqrt{{\sigma_1}^2 + {\sigma_2}^2}$. Since $\mu_1 + \mu_2 + n\sigma_1 + n\sigma_2 - (\mu_1 + \mu_2 + n\sqrt{{\sigma_1}^2 + {\sigma_2}^2}) = n(\sqrt{(\sigma_1 + \sigma_2)^2} - \sqrt{(\sigma_1 + \sigma_2)^2 - 2\sigma_1\sigma_2} > 0$, VA approach reduces the pessimistic estimation.

IV. The PDF of the Transmission Latency

In this section, the impact of wireless channel variations on the transmission latency is investigated. Different modulation schemes and stop-and-wait ARQ scheme are jointly studied.

Stop-and-wait ARQ is a basic approach to conduct error control in digital communications. The sender waits for an acknowledgement (ACK) after sending a packet, and the sender will not send any sequential packet until ACK is received or there's a timeout. The receiver will send an ACK after receiving a packet successfully. If the received packet is lost or damaged due to the non-ideal behavior of the channel, the receiver will simply discard the packet.

In this paper, we define T_s as the transmission success delay, which is the duration of a successful packet transmission. We assume that packet sizes are fixed for all the data transmissions, and thus the transmission success delay is a constant. The queue delay and propagation delay are ignored for the simplicity of the analysis.

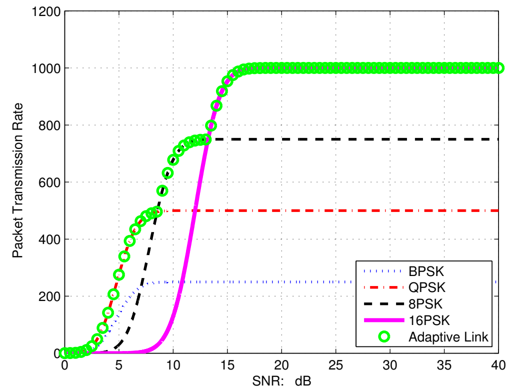


Fig. 2: Packet transmission rates for different modulation schemes

Once the channel causes errors, the total time required to deliver a correct packet needs to be recalculated. Let n be the number of transmissions required to deliver a packet successfully, and P_p be the packet error rate. Then $n = k$ transmissions are required if the first $k - 1$ transmissions are in error and the kth transmission is error free, thus

$$P[n = k] = (1 - P_p)P_p^{(k-1)} \quad \forall \quad k = 1, 2, 3, \cdots \tag{1}$$

We set the timeout period to be $T_{out} = aT_s$, (a > 1). Then, the expected transmission latency under stop-and-wait ARQ protocol can be express as

$$\begin{aligned} t &= T_s + T_{out} \sum_{k=1}^{\infty} (k-1)\, P[n = k] \\ &= T_s + aT_s \sum_{k=1}^{\infty} (k-1)\,(1 - P_p)P_p^{k-1} \\ &= T_s + (aT_sP_p)/(1 - P_p) \\ &= T_s[1 + (a-1)P_p]/(1 - P_p) \end{aligned} \tag{2}$$

Consider a certain modulation scheme, such as BPSK. The probability of bit error for non-coherent reception is $P_b = \frac{1}{2}erfc(\sqrt{\gamma})$. And we assume that during the transmission of each packet, the SNR is constant. If P_p is very small, we know that $P_p \doteq NP_b$, where N is the packet length. Thus, the PDF of transmission latency under stop-and-wait ARQ and BPSK is

$$\begin{aligned} f_t\,(t) = &((erfc^{-1}(\frac{2(t - T_s)}{(a-1)T_sN + tN}))^2)' \times \\ &f_\gamma((erfc^{-1}(\frac{2(t - T_s)}{(a-1)T_sN + tN}))^2) \end{aligned} \tag{3}$$

where, $f_\gamma(\gamma)$ is the PDF of SNR.

Similarly, the transmission latency distribution can be obtained for different schemes, such as QPSK, 8PSK and 16PSK. We assume that the SNR has a lognormal distribution. We set the mean SNR to be 8 dB, standard deviation to be 2 dB, packet length to be 100 bits, and a to be 2. We also assume

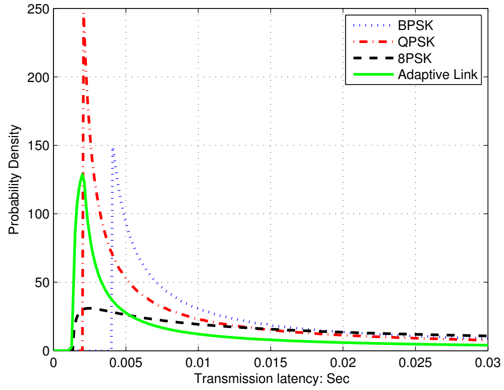
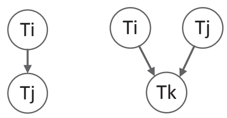
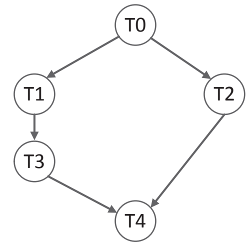


Fig. 3: Probability distributions of the transmission latency for different modulation schemes



(a) Serial *(b)* Parallel

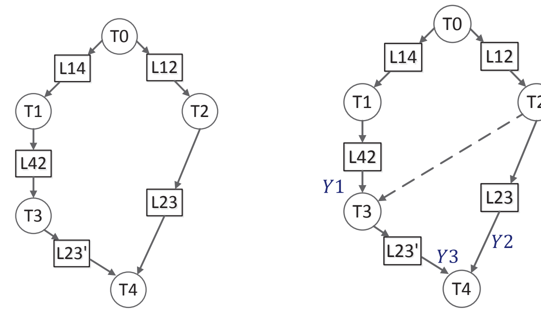
Fig. 4: Serial sub-graph and parallel sub-graph



(a) Original DAG

Task No.	T0	T1	T2	T3	T4
Service Node	*SN1*	*SN4*	*SN2*	*SN2*	*SN3*

(b) Allocation mapping



(c) DAG with transmission tokens

(d) DAG graph with transmission node and dependency edge

Fig. 5: Variation-aware task graph analysis

the channels are of fixed bandwidth, 30 kHz. The packet transmission rate curves for different modulation schemes are shown in Fig. 2. If perfect link adaptation of a cognitive radio node is used, its transmission rate performance corresponds to the green circle curve in Fig. 2. And Fig. 3 shows the PDF for the transmission latency under different modulation schemes and stop-and-wait ARQ. As shown, QPSK transmits faster than BPSK if SNR is around 8 dB, because QPSK has high spectrum efficiency. 8PSK transmits slower, because the SNR is not sufficient enough to achieve its maximum spectrum efficiency.

Then, the PDF of transmission latency for multiple packets can be derived correspondingly by the convolution operation of the PDF of the single packet transmission latency.

In this paper, TDMA is used for MAC, and the throughput for each receiver will be reduced by the number of the receiver. And the distribution of the transmission latency can be obtained in the same way. Note that if error control is also adopted in the physical layer, the transmission latency can also be obtained accordingly.

In reality, the packet transmission latency distribution can be achieved by a curve fitting of the cognitive radio nodes based on the measurement data. Thus, the variation aware approach should be able to deal with arbitrary form of distribution.

V. Variation-aware Task Graph Analysis

In this section, we go into the details about how to analyze the statistical task graph. In statistical timing analysis for task graphs, the timing quantity is computed by using two atomic functions SUM and MAX [14]. For example, in Fig. 4, T_i, T_j and T_k stand for the latency random variables for three events (either computation event, or transmission event). The output result of the serial sub-graph, is $SUM(T_i, T_j)$, and the output result of the parallel sub-graph is $MAX(T_i, T_j)$.

We analyze the two PDF operations, design the variation-aware framework to get the PDF of the makespan, and propose a new metric to evaluate the PDF of the makespan.

A. PDF Operations

(1) PDF of SUM of the two random variables

Let Y_1, Y_2 be two independent random variables with the PDF $f_{Y_i}(y)$, where $i = 1, 2$. The PDF of $Y_{sum} = Y_1 + Y_2$ is

$$f_{Y_{sum}}(y) = \int_{-\infty}^{\infty} f_{Y_1}(x) f_{Y_2}(y - x)\, dx \tag{4}$$

(2) PDF of MAX of the two random variables

Let Y_1, Y_2 be two independent random variables with the PDF $f_{Y_i}(y)$ and the cumulative distribution function (CDF) $F_{Y_i}(y)$, where $i = 1, 2$. Let $Y_{max} = \max\{Y_1, Y_2\}$. Then, the PDF of Y_{max} is

$$f_{Y_{max}}(y) = f_{Y_1}(y) F_{Y_2}(y) + F_{Y_1}(y) f_{Y_2}(y) \tag{5}$$

B. Makespan of Variation-aware Approach

Given a particular task allocation mapping as well as the priorities of the tasks, three stages are designed to get the makespan.

Stage 1: Add transmission tokens on the task graph, and associate the corresponding statistical parameters to each of the transmission tokens. For example, consider the DAG in Fig. 5 (a) with the allocation mapping shown in Fig. 5 (b), and suppose the task graph calculated task priority order is T0, T1, T2, T3, T4. After adding transmission tokens, which stand for the transmission latency, we got the graph shown in Fig. 5 (c). The circle stands for the computation latency value, and the square stands for the transmission latency variable. For example, L14 between T0 and T1 corresponds to the transmission latency from SN1 which executes T0 to SN4 which executes T1.

Stage 2: Update the task graph of S(G) by adding the scheduling constraint according to the allocation mapping and the task priorities. For the tasks that are allocated on the same SN, the execution order is determined by their priorities. For example, T2 and T3 are both allocated to SN2, and T2 should be executed first since its priority is higher than T3. Thus, we add a dependency edge between them, and the resulting graph is shown in Fig. 5 (d).

Stage 3: The makespan is expressed by SUM and MAX operations. In this example,
$MS = SUM(MAX\,(Y2, Y3)\,, T4)$, where,
$Y1 = MAX\,(SUM\,(T0, L14, T1, L42)\,, SUM\,(T0, L12, T2))$
$Y2 = SUM(T0, L12, T2, L23)$
$Y3 = SUM(Y1, T3, L23')$

Note that this method also works well for deterministic allocation method, and SUM and MAX operations are then the direct calculations of the fixed numbers.

In this paper, the priority of each task T_i is assigned based on its estimated level, which is recursively computed by traversing the DAG downward from the entry node as follows

$$prior\,[T_i] = avg\,(T_i) + \max_{T_j \in succ(i)} (E(Trans\,[i,j]) + prior[T_j] \tag{6}$$

where, $avg\,(T_i) = \sum_{j=1}^{N} ECT\,[i,j]\,/N$, $succ(i)$ is the set of successors for T_i, and $E(Trans\,[i,j])$ is the expected value of the transmission latency variable between task i and task j.

C. PDF Evaluation Metric

To evaluate the scheduling, a new design metric, reliable makespan (RM), is defined as the expected finishing time given the reliability index:

$$RM = F^{-1}(\eta) \tag{7}$$

where, F is the CDF of completion time, and η is the reliability index.

Reliable makespan basically describes how confident we are sure about the tasks can be finished by this reliable makespan. It is used to guide the task allocation and scheduling procedure.

VI. Task Allocation Methods

Genetic algorithm (GA) and different evolution algorithm (DE) are commonly used iterative probabilistic search methods for determining the allocation mapping. The candidate solutions of the problem are encoded as chromosomes. Each chromosome is made up of a sequence of genes. In each iteration, the chromosomes are updated by three operations, selection, mutation and crossover. The updating process continues until certain convergence indicators are met. Since the allocation algorithm is not the main focus of this paper, only the standard GA and DE are used to compare the estimation performance between the variation aware approach and the traditional deterministic approach.

For a DAG G with M task nodes, and the wireless system with N service nodes, the chromosome is designed as M integer values, which are all ranged from 1 to N. Each integer value stands for the index of the service node which the corresponding task is assigned to. Thus, the allocation strategy, which is updated during each iteration, can be described as these M integer values. Fig. 6 shows the chromosome example for 10 computational tasks and 4 service nodes.

Task No.	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
Service Node	*SN1*	*SN4*	*SN2*	*SN2*	*SN3*	*SN4*	*SN2*	*SN1*	*SN4*	*SN3*
Chromosome	*1*	*4*	*2*	*2*	*3*	*4*	*2*	*1*	*4*	*3*

Fig. 6: A simple chromosome with 10 tasks

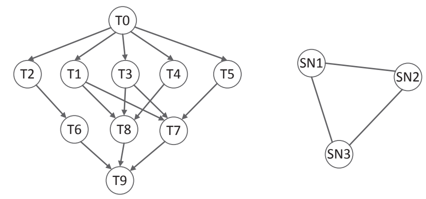


Fig. 7: Testing DAG and wireless network topology

VII. Performance Evaluation

A. Simulation setup

In the section, simulations are performed to evaluate the performance of the proposed task allocation approach. We consider the situation when the heterogeneous computation latency for different tasks on different service nodes, and the transmission latency distributions among different radio nodes are given in advance based on the prior testing or experience.

We use the classical task graph in [9] to test the performance of the evolutionary algorithms as well as the performance of VA against traditional DWCA. The testing DAG and wireless network topology are shown in Fig. 7. We assume that SN1 also serves as the SRN in this case. The DAG setting of task computation time on different service nodes is set equal to the computation cost in [9], and the unit is set to be second. For the PDF of the transmission latency, we consider the analysis in Section IV, and assume that the link adaptation from the cognitive radio node is used for modulation. TDMA is used for MAC, and the packet length is 100 bits. The mean and standard deviation of the SNR are set to be 8 dB and 2 dB, respectively. We assume that the channels are of fixed bandwidth, 30 kHz. For each task, the number of the output packets is 1000, and those packets are equally divided for the sequential task(s). The parameters of GA and DE are listed in Table II.

B. Simulation results

As discussed before, the traditional DWCA initially takes the worst-case time according to the reliability index, and then conducts deterministic operations to get the makespan. VA conducts the PDF operation initially, and from the PDF of makespan, VA will take the worst-case time according to the same reliability index. DWCA and VA are different makespan estimation approaches, while GA and DE are different ways to search for the allocation results. Thus, there are four different combinations to be compared.

Fig. 8 shows the searching performance of four combinations when reliable makespan (reliability index $\eta = 99\%$) is used as the metric. As expected, VA is a better estimator of the makespan than traditional DWCA no matter which evolutionary algorithm is used for making allocation decisions. The improvement is about 15 seconds when the results are steady around 500 iterations. In addition, DE always performs better than GA when the same makespan estimation approach is adopted, but the improvement is relatively small. In other words, the choice of VA dominates the improvement of the

TABLE II: Parameters of GA vs. DE

GA Parameters	Values	DE Parameters	Values
Population Size	20	Population Size	20
Variable Size (N)	10	Variable Size (N)	10
Cross Rate	0.8	Cross Rate	0.7
Mutation Rate	0.05	Mutation Rate	0.5
Max Iteration	500	Max Iteration	500

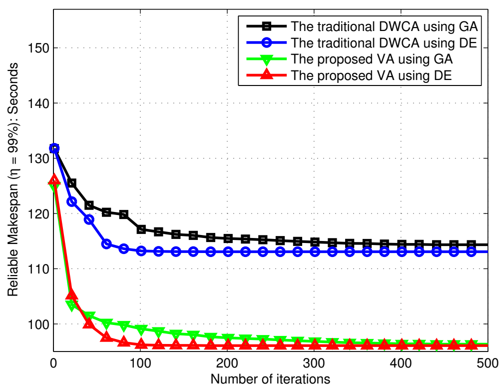


Fig. 8: Searching performance comparison

makespan estimation.

The comparison among the four combinations when the reliability index changes is shown in Fig. 9. With the same makespan estimation approach, the results are very close for either of the evolutionary algorithms. It is the choice of makespan estimation approach that makes the performance remarkably different. As can be seen, the makespan improvement from VA against DWCA is around 4 seconds up to more than 23 seconds when the reliability index increases from 84% to 99.8%. And this means that VA can reduce the pessimistic estimation effectively, especially when the reliability index is closer to 1, i.e. higher confidence of result is expected.

VIII. Conclusions

Wireless distributed computing (WDC) distributes tasks over wireless networks taking into consideration the wireless channel's fluctuations. It is important for the proper task allocation to have estimates of the makespan of a distributed task. The current methods make estimates based only on a single number without any confidence indicator. Ideally, one should have the probability distribution function (PDF) of the makespan.

In this paper, we introduce a variation-aware based framework approach for task allocation in WDC. The transmission latency is characterized as a random variable that depends on the fading of the wireless channel and the stop-and-wait ARQ protocol. We show how PDF operations can be used to for the estimation of the makespans PDF. Finally, we use the resulting allocation PDF for improving makespan. We perform the optimization using evolutionary algorithms, which are adapted with a novel PDF evaluation metric, the reliable makespan. From the simulation, we show that the VA approach provides significant improvement on the makespan estimation over the existing methods. From our findings, the makespan calculation method dominates the results irrespective of the task allocation method. Therefore, it is important to have the complete picture about the estimated makespan as it is provided by its PDF.

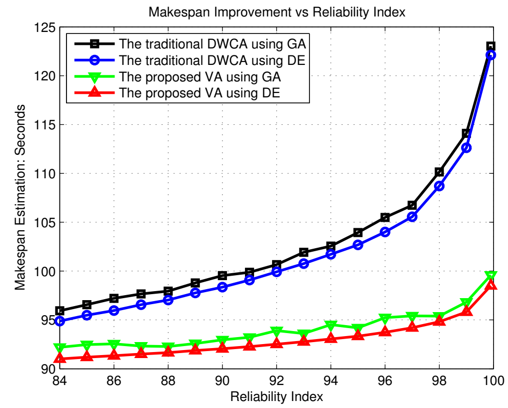


Fig. 9: Makespan estimation comparison

References

[1] X. Chen, T. R. Newman, D. Datla, T. Bose, and J. H. Reed, "The impact of channel variations on wireless distributed computing networks," *IEEE Global Telecommunications Conference*, pp. 1–6, 2009.

[2] D. Datla, X. Chen, T. Tsou, S. Raghunandan, S. M. S. Hasan, J. H. Reed, C. B. Dietrich, T. Bose, B. Fette, and J. Kim, "Wireless distributed computing: a survey of research challenges," *IEEE Communications Magazine*, vol. 50, no. 1, pp. 144–152, 2012.

[3] D. Datla, H. I. Volos, S. M. Hasan, J. H. Reed, and T. Bose, "Wireless distributed computing in cognitive radio networks," *Ad Hoc Networks*, vol. 10, no. 5, 2011.

[4] V. M. Lo, "Task assignment in distributed systems," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1983.

[5] V. Chaudhary and J. K. Aggarwal, "A generalized scheme for mapping parallel algorithms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 3, pp. 328–346, 1993.

[6] S. Kartik and C. Siva Ram Murthy, "Task allocation algorithms for maximizing reliability of distributed computing systems," *IEEE Transactions on Computers*, vol. 46, no. 6, pp. 719–724, 1997.

[7] R. C. Correa, A. Ferreira, and P. Rebreyend, "Scheduling multiprocessor tasks with genetic algorithms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 8, pp. 825–837, 1999.

[8] D. Zou, H. Liu, L. Gao, and S. Li, "An improved differential evolution algorithm for the task assignment problem," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 4, pp. 616–624, 2011.

[9] H. Topcuoglu, S. Hariri, and W. Min-You, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.

[10] Y. Yu and V. K. Prasanna, "Energy-balanced task allocation for collaborative processing in wireless sensor networks," *Mob. Netw. Appl.*, vol. 10, no. 1-2, pp. 115–131, 2005.

[11] Y. Jin, J. Jin, A. Gluhak, K. Moessner, and M. Palaniswami, "An intelligent task allocation scheme for multihop wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 444–451, 2012.

[12] D. Datla, H. I. Volos, S. Hasan, J. Reed, and T. Bose, "Task allocation and scheduling in wireless distributed computing networks," *Analog Integrated Circuits and Signal Processing*, vol. 69, no. 2, pp. 341–353, 2011.

[13] D. Datla, X. Chen, T. R. Newman, J. H. Reed, and T. Bose, "Power efficiency in wireless network distributed computing," *IEEE 70th Vehicular Technology Conference*, pp. 1–5, 2009.

[14] F. Wang, C. Nicopoulos, X. Wu, Y. Xie, and N. Vijaykrishnan, "Variation-aware task allocation and scheduling for MPSoC," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 598–603, 2007.

[15] M. Momtazpour, E. Sanaei, and M. Goudarzi, "Power-yield optimization in mpsoc task scheduling under process variation," *11th International Symposium on Quality Electronic Design*, pp. 747–754, 2010.