Hong Quy Le, Hussein Al-Shatri, Anja Klein, ”Optimal Joint Power Allocation and Task Splitting in Wireless Distributed Computing,” in *Proc. 11th International ITG Conference on Systems, Communications and Coding*, Hamburg, Germany, 6-9 February 2017.

Optimal Joint Power Allocation and Task Splitting in Wireless Distributed Computing

Hong Quy Le, Hussein Al-Shatri, Anja Klein
Communications Engineering Lab
Technische Universität Darmstadt, Merckstrasse 25, 64283 Darmstadt, Germany
{h.le, h.shatri, a.klein}@nt.tu-darmstadt.de

***Abstract*—The mobile application market is growing very fast. More and more applications require intensive computations. However, computation power of mobile devices is limited and does not catch up with the growth of computation demand of applications. Computation offloading is a promising approach for reducing the computation load and reducing execution time and energy consumption. In this work, we investigate a wireless distributed computing scenario where a mobile device exploits the computation resources of the nearby devices to reduce execution time. The main goal is to study the gain of computation offloading among heterogeneous devices. We formulate the problem of optimal offloading as a joint resource allocation problem consisting of power allocation and task splitting. The resulting problem is non-linear and non-convex. By exploiting the unique characteristics of the problem, we can transform it into a convex problem. The optimal solution of the original problem can then be determined by performing a bisection search over one parameter. Using numerical simulation, we can show that by offloading computation load to nearby devices, the execution time is reduced substantially. Moreover, the gain of offloading in comparison with computing locally increases with the ratio of the computation load to the size of input data of the task as well as with increasing number of nearby devices.**

I. INTRODUCTION

The last years have witnessed significant advancement in the development of wireless mobile devices, e.g., smart phones and smart wearable devices. Todays' mobile devices are equipped with advanced technologies, for example, high resolution cameras or integrated sensors. With the improving capabilities of the devices and the increasing interest in mobile applications and services for our daily purposes, the functionality of mobile devices has recently been progressively extended to data collection and data processing. This results in exponential growth in the usage of mobile devices in many different areas. More and more applications require high computation power, e.g., augmented reality, speech-to-text, gaming, video processing, face recognition, 3D localization/mapping or processing of bulky data. However, the efficiency of the applications is limited by the computation capability of small devices.

To overcome the limitation of computation resources at mobile devices, mobile cloud computing is proposed as a promising solution [1], [2]. Basically, mobile cloud computing aims at shifting the computation tasks and data storage from the mobile devices to entities with better computation and storage capabilities. Computation offloading is typically a service offered by a provider to the customer via networks in a typical client-server architecture [3]. In this architecture, mobile devices connect to the cloud server via a base station. Input data of the tasks will be transmitted first to the base station via wireless connection and then it will be sent to the cloud server via the backhaul network. This architecture and its related computation offloading problems in single-hop and multi-hop networks have been intensively investigated, e.g. [4], [5], [6] and references therein.

A different scenario is considered in this paper. We investigate a situation where, instead of remote cloud servers, a mobile node can exploit nearby devices with computation capability to enable parallel task execution. Nearby devices could be local computing servers at nearby small-cell base stations or other mobile nodes around that node, or computing devices that can be reached from that node via multi-hop transmission. Such a scenario, as depicted in Figure 1, is known as *wireless distributed computing* (WDC) [7] or *mobile ad-hoc cloud computing* [8]. WDC/mobile ad-hoc cloud computing is useful for offloading of splittable tasks. A splittable task can be partitioned into independent sub-tasks. The sub-tasks are then distributed to other devices and processed in parallel. Some practical applications of WDC/mobile ad-hoc cloud computing and splittable task model are discussed in references [8][9].

In [8], the authors proposed a general framework for mobile ad-hoc cloud computing and implementation with real devices. In [9], a similar framework is considered and an algorithm for task splitting is considered. They also implemented a test-bed consisting of android devices to validate their algorithm. They showed that by using a mobile ad-hoc cloud computing architecture, the execution time and the energy consumption can be reduced greatly. However, they neither considered the case when the source node can send the sub-tasks to the neighbors simultaneously (on orthogonal frequency channels) nor considered heterogeneous radio channel conditions. In their model, the wireless connections from the source node to the neighbors have the same rate. The neighbors are sorted based on their computation power. Based on this, transmissions from the source node to the neighbors are scheduled one after another. The task splitting is then considered taking into account both, delay caused by the data transmission and by the sub-task computation.

A similar scenario is considered in [10]. The authors focused on maximizing nodes' life-time. They investigated and proposed a solution to the problem of joint clock frequency

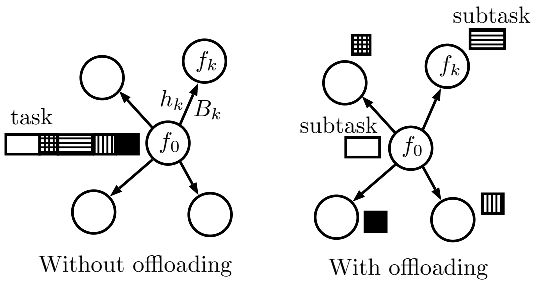

Fig. 1. Network with one source node and multiple neighbors

control, power control of mobile nodes, task splitting, and communication rate among mobile nodes. In [11], the WDC scenario in multi-hop wireless networks is considered. The authors proposed an algorithm for performing node selection and task splitting. However, in their model, the rates from the source node to the computing nodes are preallocated.

Different from the mentioned works, in this paper, we focus on the problem of minimizing the total execution time including transmission time and computation time. Our aim is to investigate and propose an optimal algorithm for the joint task splitting and radio resource allocation problem in a scenario where the source node can transmit data to the neighbors simultaneously. The wireless links are, however, coupled in such a way that there is a total transmit power constraint on the sum of the powers used on the wireless channels. We also consider the heterogeneous scenario where the wireless channels have different gains and the mobile nodes have different computation capabilities.

The rest of this paper is organized as follows: Section II presents the network model and splittable task model. The problem of optimal task splitting and radio resource allocations is formulated in Section III and its solution is presented in Section IV. The performance of the proposed solution is then investigated via numerical simulation in Section V. Finally, we conclude our work in Section VI.

II. Network and Task Model

A. Network Model

The considered network is depicted in Figure 1. It consists of a source node with a splittable computation task and K neighbors. The source node has direct connection to the neighbors on orthogonal frequency channels. Let B be the total bandwidth of the system. In this work, we do not consider bandwidth allocation, thus, the bandwidth for each channel from the source to the neighbors is pre-allocated, e.g. $B_i = \frac{B}{K}$, $i = 1, \ldots, K$. We consider a block fading channel, i.e. the channel is invariant over a frame, and the channel coefficient is drawn from a distribution and is independent from frame to frame. We also assume that the data transmission duration is much longer than the coherence time of the channel. The transmit data will be divided into packets. In each frame, one packet will be transmitted. For each link i, let h_i denote the random variable representing the channel coefficient. Let $F_i(x)$ be the Complementary Cumulative Density Function (CCDF) of $|h_i^2|$. We assume the network operates at the rates satisfying the packet error rate of ϵ, where ϵ is a given designed system parameter, e.g., $\epsilon = 0.1$ is commonly used in practical communications systems. The transmission rate $C_{i,\epsilon}$ is choosen such that

$$\text{Prob}\left(B_i \log_2\left(1 + \frac{p_i |h_i|^2}{B_i N_0}\right) < C_{i,\epsilon}\right) = \epsilon, \tag{1}$$

where $p_i \geq 0$ is the transmit power allocated to the i-the link and N_0 is the power spectral density of the additive white Gaussian noise at the receiver. According to [12], we have

$$C_{i,\epsilon} = B_i \log_2\left(1 + \frac{F_i^{-1}(1-\epsilon) p_i}{B_i N_0}\right). \tag{2}$$

We assume that if a transmission of a data packet in a frame is erroneous, then that packet will be retransmitted in the next frame until it is successfully received at the receiver. With packet error rate of ϵ, the average number of frames for a successul transmission is $\frac{1}{1-\epsilon}$. Therefore, the average rate of the i-th link is

$$r_i = (1-\epsilon) C_{i,\epsilon} = (1-\epsilon) B_i \log_2\left(1 + a_i p_i\right), \tag{3}$$

where $a_i = \frac{F_i^{-1}(1-\epsilon)}{B_i N_0}$. Furthermore, the transmit power must satisfy the total sum transmit power constraint, i.e.,

$$\sum_{i=1}^{K} p_i \leq p_{\max}. \tag{4}$$

Remark 1. We have assumed that the perfect instantaneous channel state information (CSI) is known at the receiver. The instantaneous CSI can be obtained through pilot-aided channel estimation. The CCDF can be obtained from a channel model based on practical measurement or training. The source node only needs to know the value $F_i^{-1}(1-\epsilon)$. This can be obtained by a feedback channel from the receivers. The amount of feedback is small, therefore, it is feasible for practical implementation.

B. Splittable Task Model

We consider a task model when the original task can be partioned into a number of independent subtasks. The computation requirement of each subtask is proportional to the size of the subtask. Such a model is applicable for data-partioned-oriented applications [13]. One example are language translation applications. In these applications, the input audio file or text file can be split into smaller chunks and the chunks can be translated independently. The time to process each chuck is approximately proportional to the chunk's size.

Let $\mathcal{T}$ be the splittable task with W data input bits and requiring L CPU cycles of processing. The source partitions the original task into $K+1$ sub-tasks $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_K$ with $x_0, x_1, ..., x_K$, $0 \leq x_i \leq 1$, $i = 0, \ldots, K$, being the corresponding percentage of the original task that is partitioned into the sub-tasks. Sub-task T_0 is calculated locally at the source node and for each $i = 1, \ldots, K$, sub-task T_i is offloaded to the i-th neighbor, see Fig. 1. Similar to [9], we assume that the task $\mathcal{T}$ is uniformly splittable, i.e. a fraction of x of the original task has xW input data bits and requires xL CPU cycles for processing. The sub-tasks must cover the complete original task, i.e.,

$$\sum_{i=0}^{K} x_i = 1. \tag{5}$$

The computation load of sub-task $\mathcal{T}_i$ is given by x_iL. Let f_i denote the clock speed of the i-th neighbor, and f_0 denote the clock speed of the source node. Then the processing time of sub-task $\mathcal{T}_i$ is calculated as

$$T_{\mathrm{p},i} = \frac{x_iL}{f_i}, i = 0, \ldots, K. \tag{6}$$

We assume that the size of the results of the sub-tasks after processing is very small in comparison with the size of input data. This assumption is valid for a large range of applications [9]. Therefore, the time required for transmitting the results from the neighbors back to the source node is ignored.

III. Optimization Problem Formulation

Computation offloading in wireless networks must take into account the delay caused by data transmission via wireless communications because the nodes can process the sub-task only when the data transmission of the input bits is finished. The time when sub-task $\mathcal{T}_i$, $i = 1, \ldots, K$, is finished is

$$T_i = \frac{x_iW}{(1-\epsilon)B_i \log_2(1 + a_ip_i)} + \frac{x_iL}{f_i} \tag{7}$$

where the first term is the data transmission time duration and the second term is the processing time duration. No communication is required for sub-task $\mathcal{T}_0$ at the source node. Thus, the time when the source node finishes sub-task $\mathcal{T}_0$ is

$$T_0 = \frac{x_0L}{f_0}. \tag{8}$$

The completion time is defined as the time duration from the transmission of the sub-tasks to the neighbor nodes until all sub-tasks are finished. Therefore, the completion time is defined by $\max\limits_{0\le i\le K} T_i$. The problem of minimizing the completion time can now be formulated as

$$\begin{aligned} &\min \max_{0\le i\le K} T_i \\ &\sum_{i=1}^{K} p_i \le p_{\max},\ p_i \ge 0,\ i = 1, \ldots, K \\ &\sum_{i=0}^{K} x_i = 1,\ x_i \ge 0,\ i = 0, 1, \ldots, K. \end{aligned} \tag{9}$$

In general, problem (9) is non-linear and non-convex. There is no efficient algorithm to find the optimal solution. However, by exploiting the unique characteristics of the problem, in the next section we show that the optimal solution can be found by solving another convex problem.

IV. Power Allocation and Task Splitting Algorithms

By adding a new variable T, the minimum completion time problem can be transformed to the following problem:

$$\min \quad T \tag{10}$$

$$\text{s.t.} \quad \frac{x_0L}{f_0} \le T \tag{11}$$

$$\frac{x_iW}{(1-\epsilon)B_i \log_2(1 + a_ip_i)} + \frac{x_iL}{f_i} \le T,\ i = 1, \ldots, K \tag{12}$$

$$\sum_{i=1}^{K} p_i \le p_{\max},\ p_i \ge 0,\ i = 1, \ldots, K \tag{13}$$

$$\sum_{i=0}^{K} x_i = 1,\ x_i \ge 0,\ i = 0, 1, \ldots, K \tag{14}$$

where T, $x_0, x_1, \ldots, x_K$, and $p_1, \ldots, p_K$ are variables. For fixed $p_1, \ldots, p_K$, the problem is linear with respect to T, $x_0, x_1, \ldots, x_K$.

We now propose two different algorithms to solve the above optimization problem. The first is a suboptimal algorithm, which first allocates the power and then optimizes the task splitting. In the second, optimal algorithm, power allocation and task splitting are optimized jointly.

A. *Optimal task splitting algorithm for preallocated power*

In this case, we assume the powers p_i are already allocated according to some given criterion, e.g., maximum sum rate. By introducing the new parameters

$$b_0 = \frac{L}{f_0},\ b_i = \frac{W}{(1-\epsilon)B_i \log_2(1 + a_ip_i)} + \frac{L}{f_i}, i = 1, \ldots K, \tag{15}$$

the corresponding problem obtained from problem (10) for fixed p_i can be written as

$$\min \quad T \tag{16}$$

$$\text{s.t.} \quad x_ib_i \le T,\ i = 1, \ldots, K, \tag{17}$$

$$\sum_{i=0}^{K} x_i = 1,\ x_i \ge 0,\ i = 1, \ldots, K. \tag{18}$$

Due to (17), $\sum\limits_{i=1}^{K} x_i \le T \sum\limits_{i=1}^{K} \frac{1}{b_i}$ holds. Moreover, because of (18), we obtain $1 \le T \sum\limits_{i=1}^{K} \frac{1}{b_i}$. As a result,

$$T \ge \frac{1}{\sum\limits_{i=1}^{K} \frac{1}{b_i}}. \tag{19}$$

The minimum of T is achieved in case of equality, i.e., if and only if $x_ib_i = T$, for all $i = 0, \ldots, K$. Therefore, the optimal solution of problem (16) is:

$$T^* = \frac{1}{\sum\limits_{i=1}^{K} \frac{1}{b_i}} \text{ and } x_i^* = T^* \frac{1}{b_i},\ i = 1, \ldots, K. \tag{20}$$

One important property is that, at the optimum point, all the inequality constraints become equality constraints.

B. Optimal joint power allocation and task splitting algorithm

Exploiting the insight obtained from solving the optimal task splitting for preallocated power, we now propose an optimal solution to the joint power allocation and task splitting problem (10). First, we prove the following lemma:

Lemma 1. *At the optimum point, the inequality constraints (11) and (12) become equalities.*

Proof. Let T^*, x_i^*, p_i^* be the optimal solution of the optimization problem (10). If we fix the variables p_i to p_i^*, then T^* and x_i^* are the optimal solution to the reduced linear problem for fixed powers. This problem has been solved in Section IV-A. According to the result in Section IV-A, at the optimum point all the inequality constraints become equality constraints. □

Applying the lemma, in order to find the optimal solution of the original problem, we need to consider only the case when constraints (11) and (12) become equality. As a result the variables x_i can be rewritten as a function of T and p_i as

$$x_0 = T\frac{f_0}{L} \text{ and } x_i = T\frac{(1-\epsilon)f_iB_i\log_2(1+a_ip_i)}{Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)}. \tag{21}$$

Combined with the constraint (14) it must hold

$$\begin{aligned} 1 &= T\frac{f_0}{L} + \sum_{i=1}^{K} T\frac{(1-\epsilon)f_iB_i\log_2(1+a_ip_i)}{Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)} \\ &= T\left(\frac{\sum_{i=0}^{K} f_i}{L} - \frac{W}{L}g(p_1,\dots,p_K)\right) \end{aligned} \tag{22}$$

where

$$g(p_1,\dots,p_K) = \sum_{i=1}^{K}\frac{f_i^2}{Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)}. \tag{23}$$

Thus,

$$T = \frac{1}{\frac{\sum_{i=0}^{K} f_i}{L} - \frac{W}{L}g(p_1,\dots,p_K)}. \tag{24}$$

Let $g_{\min}$ be the global minimum of $g(p_1,\dots,p_K)$ under the constraints (13). We obtain

$$T \geq \frac{1}{\frac{\sum_{i=0}^{K} f_i}{L} - \frac{W}{L}g_{\min}} =: T_{\text{lb}}. \tag{25}$$

As a consequence of (25), T is lower bounded by T_{lb}. Moreover, suppose g achieves the global minimum $g_{\min}$ at $p_1^*,\dots,p_K^*$. If we choose $T = T_{\text{lb}}$, $p_i = p_i^*$, and x_i are calculated from (21) with $T = T_{\text{lb}}$, $p_i = p_i^*$, then the constraints of the problem (10) are all satisfied and T achieves its lower bound value T_{lb}. Therefore, in order to solve problem (10) we only have to solve the following power allocation problem with variables p_i:

$$\begin{aligned} \min \quad & \sum_{i=1}^{K}\frac{f_i^2}{Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)} && (26) \\ \text{s.t.} \quad & \sum_{i=1}^{K} p_i \leq p_{\max} && (27) \\ & p_i \geq 0,\ i=1,\dots,K. && (28) \end{aligned}$$

For each i, function $Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)$ is concave and positive with respect to $p_i \geq 0$, thus the function $\frac{f_i^2}{Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)}$ is convex with respect to $p_i \geq 0$. Therefore, the above optimization problem is convex. Moreover, since Slater's constraints qualifications hold true, we may impose the KKT conditions to find the optimal solutions. The corresponding Lagrangian is

$$\begin{aligned} \mathcal{L}(\mathbf{p}) = &\sum_{i=1}^{K}\frac{f_i^2}{Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)} \\ &- \sum_{i=1}^{K}\nu_ip_i + \lambda(\sum_{i=1}^{K}p_i - p_{\max}) \end{aligned} \tag{29}$$

where the variables ν_i, λ are all nonnegative coefficients representing the Lagrange multipliers

$$\begin{gathered} u_i(p_i) = \lambda,\ i=1,\dots,K \\ \lambda(\sum_{i=1}^{K}p_i - p_{\max}) = 0 \\ p_i \geq 0,\ i=1,\dots,K, \end{gathered} \tag{30}$$

where

$$u_i(p_i) = \frac{\log_2(e)(1-\epsilon)f_i^2La_i}{(1+a_ip_i)\left(Wf_i+(1-\epsilon)LB_i\log_2(1+a_ip_i)\right)^2}. \tag{31}$$

From (30), it follows that λ must be strictly positive. Moreover, for each $i = 1,\dots K$, the function $u_i(p_i)$ is strictly decreasing for $p_i \geq 0$. Thus, the inverse function u_i^{-1} exists and it is also strictly decreasing. Thus, the system of equations (30) has a unique solution λ^*, p_i^* satisfying

$$\begin{gathered} p_i^* = \left[u_i^{-1}(\lambda^*)\right]^+ \\ \sum_{i=1}^{K}p_i^* = p_{\max} \end{gathered} \tag{32}$$

where $[x]^+ = \max\{0,x\}$. The new system of equations can be solved by using bisection search on λ. The proposed algorithm is similar to the well-known water-filling algorithm. In this work, the water level depends not only on the quality of the channel to the neighbors, but also on the computation power of the neighbors. According to (32), it may happen that some neighbors are not used if their channel is too bad or their computation capabilities are too low.

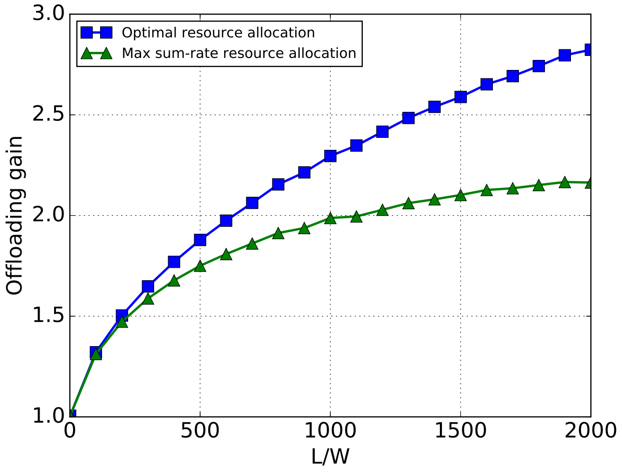

Fig. 2. Offloading gain as function of L/W, here $K = 4$ neighbors

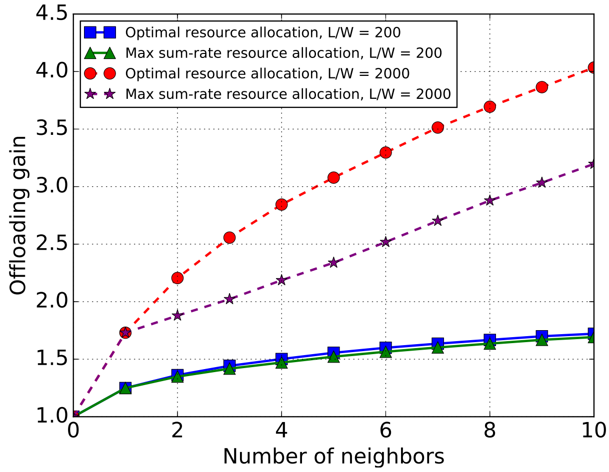

Fig. 3. Offloading gain as function of number of neighbors

V. Numerical Evaluation

In this section, the performance of the proposed algorithms is investigated as a function of $\frac{L}{W}$ and K, where L/W is the ratio of the number L of CPU cycles of the task to the number W of bits of the same task and K is the number of neighbors. We simulate a scenario of a source node and varying number K of neighbors. The processing speed of the source node is $f_0 = 10^9$ CPU cycles per second, and for node i, $i = 1, \ldots, K$, f_i is randomly chosen from the set $\{0.5, 1.0, 1.5, 2.0\} \times 10^9$ CPU cycles per second. The size of the input data is $W = 10$Mbits. The total available bandwidth is $B = 10$MHz. The bandwidth of each link is $B_i = \frac{B}{K} = \frac{10}{K}$ MHz. The channel gain of the i-th link is given by $|h_i|^2 d_i^{-3}$, where the channel coefficient h_i is a random variable following the Rayleigh distribution with zero mean and unit variance. The position of the i-th neighbor is chosen randomly within the circle with radius of 100m around the source node. The power spectral density of the additive white Gausian noise is $N_0 = 10^{-12}$W/Hz. The total transmit power is $p_{\max} = 1$W. The packet error rate is $\epsilon = 0.1$.

The performance of the proposed optimal and sub-optimal algorithms is measured by the offloading gain. It is defined as the ratio of the completion time achieved by local computing to the completion time achieved by offloading. In the first simulation, we investigate the offloading gain as a function of $\frac{L}{W}$. The number of neighbors is fixed with $K = 4$. The result is shown in Figure 2. The offloading gain increases with L/W. At low L/W, or in data-intensive domain, the offloading gain is low whereas the offloading gain is higher at high L/W (computation-intensive domain). This is to be expected because at low L/W when performing offloading, most of the time is spent for data transmission and, thus, reduces the benefit of having more computation resources. The difference in performance of the optimal algorithm and the sub-optimal algorithm also increases with the ratio L/W. This can be explained as follows. At low L/W, only a small part of the original task is offloaded. Therefore, the penalty of choosing the non-optimum power levels is not much. However, at high L/W, a larger part of the original task is offloaded. Therefore, non-optimum selection of transmit power results in longer transmission time and, thus, increases the overall completion time.

In the second simulation, we investigate the offloading gain as a function of the number K of neighbors. Two representative values of L/W are used in the simulation, $L/W = 200$ for data-intensive domain and $L/W = 2000$ for computation-intensive domain. The result is shown in Figure 3. The offloading gain increases with the number of neighbors. This is to be expected because more neighbors means more computation resources and more diversity of wireless channels. Much higher offloading gain is achieved in computation-intensive domain. The difference of the performance between the optimal and sub-optimal algorithm is larger and increases faster in the computation-intensive domain than in the data-intensive domain.

VI. Conclusion

In this work, we consider the problem of optimal radio resource allocation and task splitting in a wireless distributed computing scenario. The joint problem is formulated as an non-linear non-convex optimization problem. Two algorithms, one simple sub-optimal algorithm and one optimal algorithm, are proposed. Via numerical evaluation, we show that high offloading gain can be obtained by the proposed algorithms. When the size of the task's input dominates the computation load, then the offloading gain is lower.

Acknowledgement

This work has been performed in the project B03 within the Collaborative Research Center (CRC) 1053-MAKI.

REFERENCES

[1] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[2] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, no. 4, pp. 51–56, 2010.

[3] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proc. of the 6-th conference on Computer Systems*, 2011, pp. 301–314.

[4] C. Xian, Y.-H. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *Proc. of International Conference on Parallel and Distributed Systems*, vol. 2, 2007, pp. 1–8.

[5] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. of ACM/IEEE International Symposium on Low Power Electronics and Design*, 2012, pp. 279–284.

[6] S. Müller, H. Al-Shatri, M. Wichtlhuber, D. Hausheer, and A. Klein, "Computation offloading in wireless multi-hop networks: Energy minimization via multi-dimensional knapsack problem," in *Proc. of 26-th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2015, pp. 1717–1722.

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

[8] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. of the 1st ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond*, ser. MCS '10, 2010, pp. 6:1–6:5.

[9] G. Calice, A. Mtibaa, R. Beraldi, and H. Alnuweiri, "Mobile-to-mobile opportunistic task splitting and offloading," in *Proc. of 11-th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2015, pp. 565–572.

[10] S. Kim and S. Lee, "Link capacity-energy aware wdc for network lifetime maximization," *IEEE Transactions on Mobile Computing*, vol. 14, no. 8, pp. 1615–1628, 2015.

[11] T. Sookoor, D. Doria, D. Bruno, D. Shires, B. Swenson, and L. Pollock, "Offload destination selection to enable distributed computation on battlefields," in *Proc. of IEEE Military Communications Conference (MILCOM)*, 2014, pp. 841–848.

[12] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.

[13] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, 2015.