IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 28, 2016, accepted February 12, 2016, date of publication March 8, 2016, date of current version March 21, 2016.

*Digital Object Identifier 10.1109/ACCESS.2016.2538269*

# Energy Efficiency Optimization for Mobile Ad Hoc Networks

**WEN-KUANG KUO[1] AND SHU-HSIEN CHU[2]**

[1]Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan 701, Taiwan
[2]Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA

Corresponding author: W.-K. Kuo (wkuo@ee.ncku.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Contract 104-2221-E-006-108-.

**ABSTRACT** Tremendous traffic demands for ubiquitous access and emerging multimedia applications significantly increase the energy consumption of battery-powered mobile devices. This trend leads to that energy efficiency (EE) becomes an essential aspect of mobile ad hoc networks (MANETs). In this paper, we explore EE optimization as measured in bits per Joule for MANETs based on the cross-layer design paradigm. We model this problem as a nonconvex mixed integer nonlinear programming (MINLP) formulation by jointly considering routing, traffic scheduling, and power control. Because the nonconvex MINLP problem is NP-hard in general, it is exceedingly difficult to globally optimize this problem. We, therefore, devise a customized branch and bound (BB) algorithm to efficiently solve this globally optimal problem. The novelties of our proposed BB algorithm include upper and lower bounding schemes and branching rule that are designed using the characteristics of the nonconvex MINLP problem. We demonstrate the efficiency of our proposed BB algorithm by offering numerical comparisons with a reference algorithm that uses the relaxation manners proposed in [1]–[3]. Numerical results show that our proposed BB algorithm scheme, respectively, decreases the optimality gap 81.98% and increases the best feasible solution 32.79% compared with the reference algorithm. Furthermore, our results not only provide insights into the design of EE maximization algorithms for MANETs by employing cooperations between different layers but also serve as performance benchmarks for distributed protocols developed for real-world applications.

**INDEX TERMS** Energy efficiency, MANET, cross layer, optimization, branch and bound.

## I. INTRODUCTION

A MANET is a self-organizing set of mobile devices that communicate with one another across multiple hops in a distributed manner. Because of the widespread use of cheaper, smaller, and more powerful portable devices, MANETs have become a promising and growing technique. With recent advances in information and communication technology (ICT), MANETs are able to support high network capacity and proliferating multimedia services, such as video on-demand, surveillance, remote education, and health monitoring, etc. MANET traffic produced for ubiquitous access and multimedia applications with quality of service (QoS) requirements considerably increases energy exhaustion of mobile devices. Energy is a scarce resource for mobile devices, which are typically driven by batteries with limited capacities. Further, progress in battery technology is slow and expected to improve little in the near future [4]. Under such critical conditions, optimal EE design that concentrates on the most economical ways of utilizing mobile device energy while ensuring proper network operations is an urgent requirement for MANETs.

EE optimization of mobile communication systems has received much attention in the literature. For instance, in [5], the authors optimized link-level EE of the wireless network under static and time-variant fading channels. In [6], the authors studied link-adaptive transmission for maximizing the EE of the orthogonal frequency division multiplexing (OFDM) system by presenting an energy-efficient water-filling power allocation algorithm. In [7], the authors introduced channel selection and power allocation mechanisms to optimize the EE of a distributed cognitive radio network where the transmitter directly sent data to the receiver (i.e., a single-hop network). In [8], the authors used game theory to develop multiuser detection and power control methods to optimize EE for each user in a wireless network. In [9], the author designed a noncooperative game where each user in a mobile network chooses its transmission power and rate to maximize the EE, while guaranteeing the

QoS requirements. In [10], the author proposed a power control algorithm using noncooperative game theory to enable multimedia transmission over a wireless network and maximize the EE for each user. In [11]–[14], the authors discussed resource allocation methods to optimize EE of cellular networks without routing capabilities. Although [5]–[14] examine EE optimization for different wireless communication systems, they consider only physical (PHY) and link layers but neglect network layer issues. Obviously, results shown in [5]–[14] are inapplicable to MANETs because one of the most important features of MANETs is to provide routable networking environments.

In [15], the authors presented an analytical manner for computing EE of the MANET by taking PHY and network layers into account but ignored link layer issues. In [16], the authors maximized EE for the MANET using cooperative multi-input-single-output transmissions. They handled multihop routing by proposing algorithms for selecting hop distance and the number of cooperating nodes around each relay node. Nevertheless, they disregarded PHY and link layer problems. As discussed in [4], cross-layer optimization can substantially enhance EE by designing resource allocation mechanisms that exploit the cooperations between different layers to adapt to variations of services, traffic, and environments. Although concentrating on the EE optimization problems in MANETs, [15], [16] ignore the consequential interdependencies between different layers of the entire network. Hence, their approaches only provision suboptimal solutions that, in some cases, are likely to be far off from the global optimum. To the best of our knowledge, maximizing EE of the MANET by jointly considering the PHY, link, and network layers has yet to be researched.

To fill this gap, we investigate this cross-layer optimization problem in this paper. We consider a set of communication sessions, in which each session has its own peak and minimum sustained rate demands in a time-slotted MANET. The source of each session generates data relayed to the destination though multihop routing. We define EE of the MANET as the total session rate divided by the aggregate power consumption of active nodes over the scheduling time period. Overall, we propose a cross-layer optimization framework to maximize EE by jointly computing routing path, transmission schedule, and power control corresponding to the network, link, and PHY layers, respectively. The routing problem involves how to choose the set of paths to route data from the source to the destination for each session. For transmission schedule, the problem involves determining the set of nodes that are active in each time slot. The power control problem is to specify the transmission power of each active node in each time slot. We formulate this problem as nonconvex MINLP problem $(P_1)$, which is NP-hard in general. Challenges of globally optimizing $(P_1)$ arise from the combinatorial property, and the nonconvexities of the linear fractional objective function and bilinear products appearing in its constraints.

To address these issues, we developed a novel BB algorithm to globally optimize $(P_1)$ by exploiting its specific nature. Key innovations of our proposed BB algorithm are as follows. First, to obtain upper bounds (UBs), we employ recent advances in piecewise linear relaxations (PLRs) of bilinear terms and piecewise convex hulls (PCHs) of *log* functions to build a relaxed mixed integer linear fractional programming (MILFP) model. Nevertheless, the MILFP is still a nonconvex MINLP and mathematically intractable [17], [18]. Therefore, we transform it into an equivalent mixed integer linear programming (MILP) problem using the scheme proposed by [17]. This allows us to apply effective MILP techniques to solve the MILFP with global optimality. Second, to compute lower bounds (LBs), we keep link activations and power allocations of $(P_1)$ fixed at their optimal values from the UB problem. We then transform $(P_1)$ into an equivalent linear programming (LP) model. We can obtain a valid LB by combining optimal solution of the LP problem with outcomes of link activations and power allocations gained by solving the UB problem. Third, for problem partition, we design a branching rule that can decrease relaxation error of the UB problem to the extent possible.

We compared our proposed BB algorithm with a reference BB algorithm using the relaxation manners suggested in [1]–[3]. Numerical results show that our algorithm significantly outperformed the reference algorithm in terms of computational complexity. The contributions of the paper are: First, our theoretical outcome provides a solution to determine the optimal EE of the MANET by exploiting the cross-layer design principle. Second, in the real world, it still deserves to design distributed algorithms and protocols. However, to the extent of our knowledge, there is no any technique that can optimize the nonconvex MINLP problem in a distributed manner. This results in that distributed algorithms and protocols are developed using heuristic or approximation algorithms, or even do not take EE into consideration (such as OLSR [19], DSR [20], DSDV [21], AODV [22], and TORA [23] etc.). The common disadvantage of heuristic and approximation algorithms is that they are unable to provide the theoretical guarantee for acquiring the global optimal solution. *Hence, the current distributed algorithms and protocols cannot achieve the optimal EE operation for the MANET by fully utilizing the cooperations between different layers. Moreover, it has never been studied how far the distributed algorithms and protocols perform from the optimal EE solution, and how to enhance their efficiencies.* In this work, we solve these problems by providing theoretical results that can furnish performance benchmark comparisons, and enable researchers to gauge the effectiveness of distributed algorithms and protocols. Furthermore, our analyses provide valuable insights into not only the impact of routing strategy, transmission schedule, and power control on EE, but also into the design of novel algorithms and protocols aiming to achieve high EE for the MANET.

The remaining paper is organized as follows. In Section 2, we describe the mathematical model for EE optimization of MANETs by jointly considering PHY, link and network layers. In Section 3, we develop a novel BB algorithm using the characteristics of the model formulated in Section 2. We also explain the details of the upper and lower bounding schemes and branching technique of the proposed BB algorithm. Section 4 presents the results of computational experiments. In these experiments, we elaborate the computational efficiency of the proposed BB algorithm. We also discuss the impacts of power control, traffic scheduling and routing on the design of EE optimization protocols and algorithms for MANETs. Finally, Section 5 concludes the paper.

## II. EE OPTIMIZATION PROBLEM

We consider a MANET comprised of one set of stationary nodes $N$ connected by a set $L$ of links. We consider every link $l = n_t \to n_r$ to be directional, where $n_t$ and $n_r$ are the transmitter and receiver of $l$, respectively. We assume channel time is divided into time slots of equal length, and every node is only equipped with one transceiver. Thus, no node can send and receive in the same time slot; further, it cannot send to or receive from multiple nodes simultaneously. In other words, all nodes operate in the half-duplex mode. There are $S$ sessions representing different end-to-end traffic demands within the MANET. We denote every session $s$ as $(n_s, n_d, r_s)$, where $n_s$, $n_d$, and $r_s$ are the source node, destination node, and average transmission rate of $s$, respectively. To guarantee the required QoS, we assume $r_s$ is within range $[r_s^{min}, r_s^{max}]$ for each session $s$, where $r_s^{min}$ and $r_s^{max}$ are two pre-specified QoS parameters, namely, minimum sustained rate and peak rate, respectively. Aside from being a source or destination of a session, we assume every node can relay data for other sessions and act as a router. Our goal is to maximize EE of the entire network while supporting QoS requirements of individual sessions in terms of power control, traffic scheduling, and route assignment. We describe the details of our mathematical model and formulate the EE optimization problem of the MANET below.

### A. MATHMATICAL MODEL FOR THE EE OPTIMIZATION PROBLEM

For every link $l$ at every time slot $t$, we define binary variable $x_t^l$ as

$$x_t^l = \begin{cases} 1; & \textit{if link } l \textit{ is allowed to trasnmit at time slot } t \\ 0; & \textit{otherwise} \end{cases} \quad (\forall l \in L, t \in \Gamma), \quad (1)$$

where $\Gamma = \{1, \ldots, T\}$ and $T$ is the total number of scheduled time slots. We assume transmission power on link $l$ at time slot $t$, i.e., $p_t^l$, is continuously adjusted in given interval $[0, p_{max}]$. Following the same definition of [2] and [3], we define $0 \leq p_t^l \leq p_{max}$ if link $l$ is allowed to transmit during time slot $t$ and $p_t^l = 0$, otherwise. We thus have the following constraint

$$p_t^l \begin{cases} \in [0, p_{max}], & \textit{if } x_t^l = 1 \\ = 0, & \textit{if } x_t^l = 0 \end{cases} \quad (\forall l \in L, t \in \Gamma). \quad (2)$$

Note that being allowed to transmit does not necessarily mean a transmission actually occurs, which is decided by the optimization algorithm.

As every node is equipped only with a transceiver, it must adhere to half-duplex data transmission. Thus, in every time slot, at most only one of a node's incoming and outgoing links is permitted to transmit. To characterize this constraint, we have

$$\sum\nolimits_{l \in IL(i) \cup OL(i)} x_t^l \leq 1 \quad (\forall i \in N, t \in \Gamma), \quad (3)$$

where $IL(i) = \{k \to i | k \in N\}$ and $OL(i) = \{i \to j | j \in N\}$ are sets of links whose receiver and transmitter are both $i$, respectively.

The quality of a wireless link depends on the signal to interference and noise ratio ($SINR$). We express $SINR$ of link $l$ at time slot $t$ as

$$SINR_t^l = \frac{h_l p_t^l}{\sigma W + \sum\nolimits_{l' \in L, l' \neq l} h_{(TX_{l'}, RX_l)} p_t^{l'}} \quad (\forall l \in L, t \in \Gamma), \quad (4)$$

where $h_l$ is the propagation gain of link $l$, $W$ is the channel bandwidth, $\sigma$ is the thermal noise density, and $h_{(TX_{l'}, RX_l)}$ is the propagation gain from the transmitter of link $l'$ (i.e., $TX_{l'}$) to the receiver of link $l$ (i.e., $RX_l$). Using the Shannon capacity theorem, we can calculate the capacity of link $l$ at time slot $t$ as $W \log_2 \left(1 + SINR_t^l\right) (\forall l \in L, t \in \Gamma)$.

In the MANET, a source node generates data transmitted to its destination node, which forms a session. Owing to the limited transmission power of a node, it may be necessary to route data through multiple intermediate nodes to ease transmission over a long distance. To provide better routing flexibility and reliability, we adopt flow splitting and multipath routing. Specifically, every node can separate its incoming traffic into subflows which are then transmitted to varied next-hop nodes. Based on the above description, we have the following flow conservation constraints. Let $f_{s,t}^l$ denote the flow rate of session $s$ on link $l$ at time slot $t$. If node $i$ is the source node of session $s$, then

$$\sum\nolimits_{l \in OL(i), t \in \Gamma} f_{s,t}^l = r_s T \quad (\forall i \in N, s \in IS(i)), \quad (5)$$

where $IS(i)$ is the set of sessions whose source node is $i$. When both sides of (5) are multiplied by the duration of time slot, the left hand side yields the total amount of traffic (in bits) generated by the source node and the right hand side gives the average session rate multiplied by the scheduling time period. Evidently, both quantities must be equal.

If node $i$ is an intermediate node of sessions$s$, then we have

$$\sum\nolimits_{l \in IL(i), t \in \Gamma} f_{s,t}^l = \sum\nolimits_{l \in OL(i), t \in \Gamma} f_{s,t}^l \quad (\forall i \in N, s \in S - (IS(i) \cup OS(i))), \quad (6)$$

where $OS(i)$ is the set of sessions whose destination node is $i$. (6) implies that for every session, the net amount of incoming and outgoing traffic must be equal at each relay node.

In [1]–[3], the authors have verified that both (5) and (6) guarantee the flow balance equation at the destination node of session $s(\forall s \in S)$. Thus, we omit flow conservation constraints of destination nodes for all sessions.

Furthermore, at every time slot, the total amount of traffic from different sessions on a particular link cannot exceed the link's capacity. Therefore, we have the following constraint

$$\sum\nolimits_{s\in S} f_{s,t}^{l} \leq W \log_2 \left(1 + SINR_t^l\right) \quad (\forall l \in L, t \in \Gamma). \tag{7}$$

Finally, to support QoS requirements, we assume $r_s$ of each session $s$ must satisfy

$$r_s^{min} \leq r_s \leq r_s^{max} \quad (\forall s \in S) \tag{8}$$

Our purpose is to maximize EE of the MANET, which is defined as the total session rate divided by the power consumed by all nodes during the scheduling time period. This definition equals the amount of data transferred by all source-destination pairs divided by the energy consumed by the entire MANET. Thus, the objective function is shown as

$$EE = \frac{\sum\nolimits_{s\in S} r_s}{\sum\nolimits_{l=1}^{L} \sum\nolimits_{t=1}^{T} p_t^l} \quad \text{(bits/Joule)} \tag{9}$$

We express the resulting EE optimization model $(P)$ as follows

$$Max \frac{\sum\nolimits_{s\in S} r_s}{\sum\nolimits_{l=1}^{L} \sum\nolimits_{t=1}^{T} p_t^l} \tag{$P$}$$

subject to constraints (2)-(8), $x_t^l \in \{0, 1\}$,

$$0 \leq p_t^l \leq p_{max} \quad (\forall l \in L, t \in \Gamma)\,; r_s \geq 0 \quad (\forall s \in S);$$
$$f_{s,t}^{l} \geq 0 \quad (\forall l \in L, t \in \Gamma, s \in S).$$

### B. REFINEMENTS TO OUR MATHEMATICAL MODEL

Although we have successfully modeled $(P)$, it is only a primitive representation and unsuitable for mathematical treatments. To make formulations of $(P)$ more concise and easier to manipulate, we discuss necessary modifications and redevelop an equivalent model that is mathematically tractable as follows.

First, we reformulate (2) as (10), because (2) is inapplicable to mathematical programming.

$$0 \leq p_t^l \leq x_t^l p_{max} \quad (\forall l \in L, t \in \Gamma). \tag{10}$$

One can easily prove that (2) and (10) are equivalent.

Second, because the product form is easier to handle than the fractional form, we rewrite (4) as follows

$$\sigma W SINR_t^l + \sum\nolimits_{l'\in L, l'\neq l} h_{(TX_{l'}, RX_l)} p_t^{l'} SINR_t^l = h_l p_t^l \quad (\forall l \in L, t \in \Gamma). \tag{11}$$

We notice that (11) has nonconvex bilinear products (i.e., $p_t^{l'} SINR_t^l$) that cause $(P)$ presenting a multiplicity of local optima [24], [25]. To facilitate mathematical manipulations, we further curtail the number of bilinear terms. We define $I_t^l$ symbolizing interference of link $l$ at time slot $t$ as

$$I_t^l = \sum\nolimits_{l'\in L, l'\neq l} h_{(TX_{l'}, RX_l)} p_t^{l'} \quad (\forall l \in L, t \in \Gamma). \tag{12}$$

Next, we reformulate (11) as

$$\sigma W SINR_t^l + I_t^l SINR_t^l = h_l p_t^l \quad (\forall l \in L, t \in \Gamma). \tag{13}$$

Comparing (11) with (13), we discover that the number of bilinear products has been diminished from $O(\left|N^4\right| |T|)$ to $O(\left|N^2\right| |T|)$.

With the above modifications, we redevelop an equivalent model $(P_1)$ for $(P)$ as

$$Max \frac{\sum\nolimits_{s\in S} r_s}{\sum\nolimits_{l=1}^{L} \sum\nolimits_{t=1}^{T} p_t^l} \tag{$P_1$}$$

subject to constraints (3), (5)-(8), (10), (12), (13),

$$x_t^l \in \{0, 1\}, 0 \leq p_t^l \leq p_{max}, 0 \leq SINR_t^l \leq h_l p_{max}/\sigma W,$$
$$0 \leq I_t^l \leq p_{max} \sum\nolimits_{l'\in L, l'\neq l} h_{(TX_{l'}, RX_l)} \quad (\forall l \in L, t \in \Gamma);$$
$$r_s \geq 0 \quad (\forall s \in S); \quad f_{s,t}^{l} \geq 0 \quad (\forall l \in L, t \in \Gamma, s \in S).$$

By inspecting $(P_1)$, we observe that to globally optimize it, three major challenges need to be overcome: (1) there are binary variables with combinatorial nature, (2) the objective function is a nonconvex linear fractional function [17], [18], and (3) the bilinear terms are also nonconvex. These properties cause $(P_1)$ becoming a nonconvex MINLP which is NP-hard in general and thus incredibly difficult to solve. To settle this issue, we propose a novel BB algorithm for solving $(P_1)$ with global optimality in Section 3.

## III. OUR PROPOSED BB SOLUTION PROCEDURE

The BB algorithm, which consists of several components, defines a common framework to solve a wide class of optimization problems. The framework itself provides flexibility in the sense that for some its key components, one can design customized algorithms based on the problem's structure and properties. Because performance of the BB process critically hinges on these customized algorithms' efficiencies [26], focusing on the design of these algorithms is absolutely crucial. In the following subsections, we present the proposed BB procedure and problem-specific algorithms to its important elements developed by employing characteristics of $(P_1)$.

### A. THE MAIN ALGORITHM

The idea underlying the BB procedure is based on the "divide and conquer" paradigm, which begins by considering the original problem with the entire feasible region. The BB process performs lower-bounding and upper-bounding methods to the original problem to find LB and UB of the global optimum. If the gap between the bounds is within tolerance $\varepsilon$, then we have achieved an $\varepsilon$-optimal solution and the procedure terminates. Here, tolerance $\varepsilon$ is a sufficiently small positive constant that signifies the required precision of the ultimate solution. Otherwise, we partition the feasible region into two disjoint subregions, which together comprise the complete feasible region. Each of these subregions forms a new nonconvex MINLP model with the same objective function and constraints as the original problem, which collectively is called the subproblem.

The same algorithms are applied recursively to each subproblem to decide LB and UB of the subproblem. The LB and UB of the subproblem provide information about (1) whether a further partition on this subproblem is necessary, (2) whether the subproblem can be discarded, and (3) a better feasible solution to the original problem. The process itself generates a search tree where the root node represents the original problem. The tree search procedure continues until all nodes have been solved or deleted, or until an $\varepsilon$-optimal solution is found.

1. Let $\widehat{LB} = -\infty$, $x^* = \emptyset$. Specify the value of $\varepsilon$. Let $W=\{(P_1)\}$ and $UB_{(P1)} = \infty$
2. **If** $W = \emptyset$
   {**If** $\widehat{LB} = -\infty$, $(P_1)$ is infeasible. **Else** $x^*$ is the optimal solution.}
3. Let $\widehat{UB} = max_{q \in W}\{UB_q\}$. Compute $\varepsilon_0$.
   **If** $\varepsilon_0 \leq \varepsilon$, stop and $x^*$ is the optimal solution.
   **Else** choose a node $q$ with $UB_q = \widehat{UB}$ from $W$. Let $W = W \setminus \{q\}$.
4. Solve the UB problem of $q$.
   **If** this problem is infeasible, go to step 2.
   **Else** update $UB_q$= objective value of the UB problem of $q$.
   Calculate $\varepsilon_q$. **If** $\varepsilon_q \leq \varepsilon$, go to step 2.
5. Solve the LB problem of $q$ to acquire a solution $x_q^*$ with the objective value $LB_q$.
   **If** $LB_q > \widehat{LB}$ {
   Let $x^* = x_q^*$ and $\widehat{LB} = LB_q$.
   Delete from $W$ all nodes $q'$ with $UB_{q'} \leq \widehat{LB}$. Update $\varepsilon_q$.
   **If** $\varepsilon_q \leq \varepsilon$, go to step 2. }
6. Apply the branching rule to $q$ and generate two new subproblems $q_1$ and $q_2$.
7. Let $UB_{q1} = UB_{q2} = UB_q$. Add $q_1$ and $q_2$ to $W$. Go to step 3.

**FIGURE 1. Our proposed BB algorithm.**

Fig.1 shows our proposed BB algorithm. We denote the UB, LB, and optimal solution of $(P_1)$ as $\widehat{UB}$, $\widehat{LB}$, and $x^*$, respectively. In step 1, we initialize values of $\widehat{LB}$ and $x^*$, and set convergence tolerance $\varepsilon$. We add the root node (i.e., $(P_1)$) to problem list $W$, which serves as the collection of subproblems generated during the tree search. We assign an initial value to $UB_{(P1)}$, which designates the UB of $P_1$. In step 2, we check whether $W$ is empty. If so, we further examine $\widehat{LB}$. If $\widehat{LB}$is $-\infty$, $(P_1)$ is infeasible; otherwise we obtain optimal solution $x^*$.

In step 3, we update $\widehat{UB}$ as the maximum $UB_q$ among all nodes $q$ in $W$, where $UB_q$ is the UB of node $q$. We then compute the optimality gap $\varepsilon_0$ between $\widehat{UB}$ and $\widehat{LB}$, which is defined as

$$\varepsilon_0 = \begin{cases} \infty, & if\ \widehat{LB} = -\infty \\ \left|\frac{\widehat{UB} - \widehat{LB}}{\widehat{UB}}\right|, & otherwise. \end{cases}$$

If $\varepsilon_0 \leq \varepsilon$, we end the procedure by achieving $\varepsilon$-convergence. Otherwise, we select and delete node $q$ with $\mathrm{UB}_q = \widehat{UB}$ from W. In step 4, we solve the UB problem of $q$ to find its optimal objective value. If this problem is infeasible, we proceed to step 2. Otherwise, we update $UB_q$ and calculate the approximation gap $\varepsilon_q$ defined as

$$\varepsilon_q = \begin{cases} \infty, & if\ \widehat{LB} = -\infty \\ \left|\frac{UB_q - \widehat{LB}}{UB_q}\right|, & otherwise. \end{cases}$$

If $\varepsilon_q \leq \varepsilon$, we go to step 2 because the best feasible solution in $q$ cannot be strictly better than $\widehat{LB}$ above the $\varepsilon$ tolerance. Therefore, we exclude $q$ from further consideration and choose another node from $W$ to investigate.

In step 5, we solve the LB problem of $q$ to obtain feasible solution $x_q^*$. Since $x_q^*$ is feasible to $(P_1)$, its objective value $LB_q$ provides a LB of $(P_1)$. We then compare values of $\widehat{LB}$ and $LB_q$. If $LB_q > \widehat{LB}$, we update $\widehat{LB}$ and$x^*$, because $LB_q$ improves the best known feasible solution of $(P_1)$ so far. We also delete all nodes $q'$ with $UB_{q'} \leq \widehat{LB}$ from $W$ because they cannot possibly contain the global optimum. Subsequently, we update $\varepsilon_q$ and check whether $\varepsilon_q \leq \varepsilon$. If yes, we proceed to step 2 for the same reason mentioned in step 4. In contrast, if $LB_q \leq \widehat{LB}$ or $\varepsilon_q > \varepsilon$, we go to step 6. This is because there may still be scope for locating a better LB in $q$. Hence, we need to partition $q$ further. In step 6, we divide $q$ into two child nodes $q_1$ and $q_2$ by employing the branching rule. In step 7, we assign UB of $q_1$ and $q_2$, $UB_{q1}$ and $UB_{q2}$, as $UB_q$. We then add $q_1$ and $q_2$ to $W$, and go to step 3.

Clearly, the success of the BB algorithm primarily relies on how fast $\varepsilon_0$ decreases. To expedite the convergence rate of the BB algorithm, $\varepsilon_0$ must quickly reduce during the tree search procedure. Because $\varepsilon_0$ is determined by the tightness of the LB and UB, designing efficient bounding schemes to obtain stringent bounds thus becomes critical for the BB process. To approach this issue, we propose novel upper and lower bounding methods in Sections 3.2 and 3.3, respectively. Further, to improve UBs through the BB process, we devise an innovative branching strategy to diminish the largest relaxation error of the UB problem in Section 3.4.

## B. UPPER BOUNDING SCHEME

### 1) MOTIVATION

In [1]–[3], the authors studied different cross-layer optimization problems in wireless communications. These problems are all nonconvex MINLPs with bilinear terms and *log* functions but lack fractional functions. [1]–[3] also used BB algorithms to solve these optimization problems. To build the relaxation problem for the BB procedure, [1]–[3] all used the same linearization methods, including the reformulation-linearization technique (RLT) and the single convex hull (SCH), to approximate the bilinear term and the *log* function, respectively. Although being helpful for creating the relaxation problem, the relaxation methods used in [1]–[3] have several major drawbacks. First, as reported by exhaustive computational studies presented in [24] and [25], the RLT yields weak relaxations for the bilinear term and is very slow in declining UBs of the BB procedure for

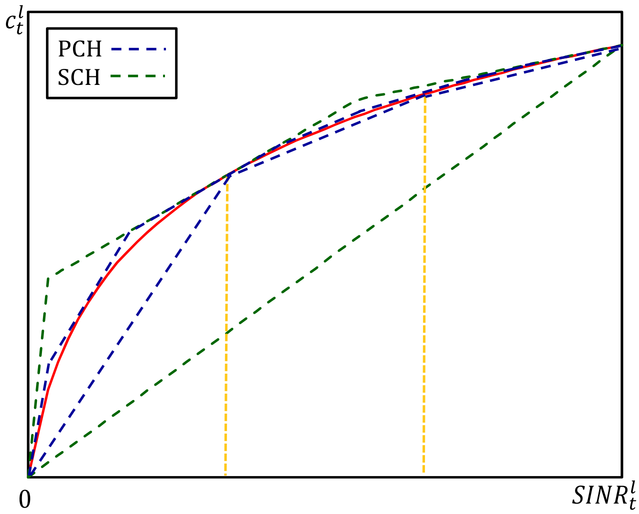


**FIGURE 2.** PCH and SCH for $C_t^l = ln(1 + SINR_t^l)$.

maximization problems. It thus requires exploration of many more nodes, considerably decelerating convergence of the BB process and causing excessive computational complexity. Second, as illustrated in Fig. 2, the SCH is too loose to accurately bound the *log* function. Moreover, since linear fractional functions are nonconvex, it needs special mathematical processing to obtain the global optimum for problems with such functions. BB algorithms in [1]–[3] thus are incapable of globally optimizing ($P_1$) because they do not handle this issue. To improve these defects, we propose a novel upper bounding scheme using PLR and PCH to bound the bilinear product and the *log* function, respectively. We then establish a MILFP model serving as a tight UB problem of ($P_1$). However, MILFPs are NP-hard in general. Therefore, we transform the MILFP model into its equivalent MILP problem using the method suggested in [17]. In this way, we can find the UB of ($P_1$) by exploiting cutting-edge MILP solvers that have become highly effective over the past decades [26]–[28].

To remedy the shortcoming of RLT, the authors in [24] proposed fifteen PLRs to relax bilinear products. To use the PLRs in [24], we first divide the original domain of one of the variables involved in the bilinear product into smaller subdomains. Then we bound the bilinear term using more accurate linear subenvelopes in each subdomain, and employ a set of newly defined binary variables to choose the optimal subdomain. The advantage of the PLRs is that they can dramatically accelerate convergence of the BB algorithm by providing much tighter relaxations. Results in [24] show that these PLRs can reduce the optimality gap (i.e., $\varepsilon_0$) at the root node of the BB procedure from 20%–40% (i.e., two subdomains) to 75%–95% (i.e., fifteen subdomains) as compared with the RLT. Further, [25] have demonstrated that the PLRs can significantly curtail the number of nodes required for investigation during the BB process, thus decreasing the computational time from several-fold to hundreds-fold compared to RLT. To improve the quality of UB, we propose to use one of the fifteen PLRs to approximate the bilinear term.

In ($P_1$), as both $SINR_t^l$ and $I_t^l$ participate in the bilinear product, one can select either of them to partition. Nevertheless, we observe that $SINR_t^l$ also appears in the *log* function while $I_t^l$ does not. If we choose to divide on the dimension of $SINR_t^l$, we can construct the PLR and the PCH for the bilinear product and the *log* function, respectively, by separating only one unique domain. This not only strengthens the linear relaxations for both the bilinear term and the *log* function but also avoids increasing the number of variables and constraints in the relaxed formulation. Therefore, we opt to divide on the domain of $SINR_t^l$.

In [24], the authors recommend four representatives that are the most efficient among the fifteen PLRs, namely, *nf4l*, *nf4r*, *nf6t*, and *nf7r*. However, by carefully examining the four most effective PLRs, we find only *nf4l* can accomplish the above goal. Therefore, we apply *nf4l* to build a series of subenevelopes to bound the bilinear term over each subrange of $SINR_t^l$. We then use the same partitioning to establish the PCH for the *log* function.

### 2) THE CONSTRUCTION OF MILFP PROBLEM

First, we divide the domain of each $SINR_t^l$ variable into $K$ subdomains by defining a grid of points, $SINR_t^l(k)$, such that

$$SINR_t^l(k) = \underline{SINR_t^l} + kd \quad \forall l \in L, t \in \Gamma,\ k \in 0, 1, \ldots, K), \tag{14}$$

where $d = (\overline{SINR_t^l} - \underline{SINR_t^l})/K$, and $\underline{SINR_t^l}$ and $\overline{SINR_t^l}$ are lower and upper bounds of $SINR_t^l$, respectively. We call the number of subdomains $K$ the partition level.

We then introduce a set of binary variables $\lambda_t^l(k)$, that are equal to one if and only if $SINR_t^l$ lies within the $k^{th}$ segment as shown in (15),

$$\lambda_t^l(k) = \begin{cases} 1; & \textit{if } SINR_t^l(k-1) \le SINR_t^l \le SINR_t^l(k) \\ 0; & \textit{otherwise} \end{cases} \quad (\forall l \in L, t \in \Gamma, k \in \Psi), \tag{15}$$

where $\Psi = \{1, \ldots, K\}$. Since $SINR_t^l$ only lies in one of the $K$ subdomains, we have the following constraint

$$\sum\nolimits_{k=1}^{K} \lambda_t^l(k) = 1 \quad (\forall l \in L, t \in \Gamma). \tag{16}$$

Further, we define a set of continuous variables $\alpha_t^l(k)$ ($\forall l \in L, t \in \Gamma, k \in \Psi$), where $0 \le \alpha_t^l(k) \le d\lambda_t^l(k)$. These variables denote the deviation of $SINR_t^l$ from grid point $SINR_t^l(k-1)$, if $SINR_t^l(k-1) \le SINR_t^l \le SINR_t^l(k)$, or zero otherwise. We thus model $SINR_t^l$ as (17).

$$SINR_t^l = \sum\nolimits_{k=1}^{K} [SINR_t^l(k-1)\lambda_t^l(k) + \alpha_t^l(k)] \quad (\forall l \in L, t \in \Gamma) \tag{17}$$

We also reformulate $I_t^l$ by introducing another set of continuous variables $\beta_t^l(k)$ ($\forall l \in L, t \in \Gamma, k \in \Psi$), where $0 \le \beta_t^l(k) \le (\overline{I_t^l} - \underline{I_t^l})\lambda_t^l(k)$, and $\underline{I_t^l}$ and $\overline{I_t^l}$ are lower and upper bounds of $I_t^l$, respectively. Here, $\beta_t^l(k)$ signifies the deviation of $I_t^l$ from $\underline{I_t^l}$ if $SINR_t^l(k-1) \le SINR_t^l \le SINR_t^l(k)$; otherwise, it is zero. We hence denote $I_t^l$ as follows

$$I_t^l = \underline{I_t^l} + \sum\nolimits_{k=1}^{K} \beta_t^l(k) \quad (\forall l \in L, t \in \Gamma). \tag{18}$$

Finally, we define variable $w_t^l = I_t^l SINR_t^l$ and rewrite (13) as follows

$$\sigma W SINR_t^l + w_t^l = h_l p_t^l \quad (\forall l \in L, t \in \Gamma). \tag{19}$$

We then bound $w_t^l$ $(\forall l \in L, t \in \Gamma)$ using the *nf4l* given in (20)-(23).

$$w_t^l \ge \underline{I_t^l} SINR_t^l + \sum\nolimits_{k=1}^{K} \{SINR_t^l(k-1)\beta_t^l(k)\} \tag{20}$$

$$\begin{aligned} w_t^l \ge\ & \underline{I_t^l} SINR_t^l + \sum\nolimits_{k=1}^{K} \{SINR_t^l(k-1)\beta_t^l(k)\} \\ & + \sum\nolimits_{k=1}^{K} \{d\beta_t^l(k)\} + (\overline{I_t^l} - \underline{I_t^l}) \sum\nolimits_{k=1}^{K} \{\alpha_t^l(k) - d\lambda_t^l(k)\} \end{aligned} \tag{21}$$

$$\begin{aligned} w_t^l \le\ & \underline{I_t^l} SINR_t^l + \sum\nolimits_{k=1}^{K} \{SINR_t^l(k-1)\beta_t^l(k)\} \\ & + (\overline{I_t^l} - \underline{I_t^l}) \sum\nolimits_{k=1}^{K} \alpha_t^l(k), \end{aligned} \tag{22}$$

$$\begin{aligned} w_t^l \le\ & \underline{I_t^l} SINR_t^l + \sum\nolimits_{k=1}^{K} \{SINR_t^l(k-1)\beta_t^l(k)\} \\ & + \sum\nolimits_{k=1}^{K} d\beta_t^l(k), \end{aligned} \tag{23}$$

To eliminate the nonlinearity of the *log* function, we first define variable $c_t^l = ln(1 + SINR_t^l)$ and rewrite (7) as follows

$$\sum\nolimits_{s \in S} f_{s,t}^l \le \frac{W}{ln2} c_t^l \quad (\forall l \in L, t \in \Gamma). \tag{24}$$

We then bound $c_t^l$ by the PCH using the same partitioning of $SINR_t^l$ (i.e., (14)-(17)), as depicted in Fig. 2. For every subdomain $k \in \Psi$, the PCH consists of three tangents that are tangential at points $SINR_t^l(k-1)$, $\gamma = \frac{1}{3}[SINR_t^l(k-1) + SINR_t^l(k)]$, and $SINR_t^l(k)$, as well as one secant between $SINR_t^l(k-1)$ and $SINR_t^l(k)$. To derive formulas of the PCH, we take the tangent at $SINR_t^l(k-1)$ as an example. If $SINR_t^l$ lies within subdomain $k$, as elaborated in Fig. 2, we have

$$c_t^l \le \frac{SINR_t^l - SINR_t^l(k-1)}{1 + SINR_t^l(k-1)} + ln\big[1 + SINR_t^l(k-1)\big]. \tag{25}$$

This is because $c_t^l$ is smaller than the tangent at $SINR_t^l(k-1)$ denoted by the right hand side of (25). By using (17), we can rewrite this tangent as $\frac{\alpha_t^l(k)}{1+SINR_t^l(k-1)} + ln[1 + SINR_t^l(k-1)]$. We then equivalently express (25) as (26) using the method developed in [29].

$$\begin{aligned} c_t^l \le \sum\nolimits_{k=1}^{K} \Bigg\{ & \frac{\alpha_t^l(k)}{1 + SINR_t^l(k-1)} \\ & + \lambda_t^l(k) ln\big[1 + SINR_t^l(k-1)\big] \Bigg\} \quad (\forall l \in L, t \in \Gamma) \end{aligned} \tag{26}$$

In (26), since only one of the $K$ subdomains is active, only the tangent in the active subdomain exists, while tangents in all other subdomains are discarded. Therefore, no matter which subdomain is active, we can always represent the tangent at $SINR_t^l(k-1)$ $(\forall k \in \Psi)$ using (26). Similarly, one can find formulas of tangents at $\gamma$ and $SINR_t^l(k)$, and the secant between $SINR_t^l(k-1)$ and $SINR_t^l(k)$ as (27)-(29). Note that In Fig.2, we draw the PCH with 3 subdomains and the SCH used in [1]–[3]. As illustrated in the figure, the proposed PCH is much tighter than the SCH and thus can improve the quality of the UB.

$$\begin{aligned} c_t^l \le \sum_{k=1}^{K} \{ & \frac{\alpha_t^l(k) + \lambda_t^l(k)\big[SINR_t^l(k-1) - \gamma\big]}{1 + \gamma} \\ & + \lambda_t^l(k) ln(1 + \gamma)\} \quad (\forall l \in L, t \in \Gamma) \end{aligned} \tag{27}$$

$$\begin{aligned} c_t^l \le \sum_{k=1}^{K} \Bigg\{ & \frac{\alpha_t^l(k) + \lambda_t^l(k)\big[SINR_t^l(k-1) - SINR_t^l(k)\big]}{1 + SINR_t^l(k)} \\ & + \lambda_t^l(k) ln\big[1 + SINR_t^l(k)\big] \Bigg\} \quad (\forall l \in L, t \in \Gamma), \end{aligned} \tag{28}$$

$$\begin{aligned} c_t^l \ge \sum_{m=1}^{M} \{ & \lambda_t^l(k) \ln[1 + SINR_t^l(k-1)] \\ & + \frac{ln\big[1 + SINR_t^l(k)\big] - \ln\big[1 + SINR_t^l(k-1)\big]}{SINR_t^l(k) - SINR_t^l(k-1)} \alpha_t^l(k)\} \\ & (\forall l \in L, t \in \Gamma) \end{aligned} \tag{29}$$

Using the above relaxation scheme, we form UB problem $(P_2)$ from $(P_1)$, as follows

$$Max \frac{\sum_{s \in S} r_s}{\sum_{l=1}^{L} \sum_{t=1}^{T} p_t^l} \tag{$P_2$}$$

subject to constraints (3), (5), (6), (8), (10), (12), (16)-(24), (26)-(29),

$$\begin{aligned} & x_t^l \in \{0,1\}, 0 \le p_t^l \le p_{max}, 0 \le SINR_t^l \le h_l p_{max}/\sigma W, \\ & 0 \le c_t^l \le ln(1 + h_l p_{max}/\sigma W), w_t^l \ge 0, \\ & 0 \le I_t^l \le p_{max} \sum\nolimits_{l' \in L, l' \ne l} h_{(TX_{l'}, RX_l)} \ (\forall l \in L, t \in \Gamma); \\ & f_{s,t}^l \ge 0 \ (\forall l \in L, s \in S, t \in \Gamma); r_s \ge 0 \quad (\forall s \in S); \\ & \lambda_t^l(k) \in \{0,1\}, 0 \le \alpha_t^l(k) \le d\lambda_t^l(k), \\ & 0 \le \beta_t^l(k) \le (\overline{I_t^l} - \underline{I_t^l})\lambda_t^l(k) \quad (\forall l \in L, t \in \Gamma, k \in \Psi). \end{aligned}$$

### 3) THE MILFP TO MILP TRANSFORMATION

By examining $(P_2)$, we discover that the objective function is a linear fractional function and all constraints are linear except for some variables that are restricted to binary values. Therefore, it is a MILFP problem. Owing to the nonconvexity of the objective function and combinatorial feature, MILFPs are nonconvex MINLPs and NP-hard in general. To solve MILFPs for global optimality, the authors in [17] propose a novel approach to transform MILFPs into their equivalent

MILPs which are then optimized using efficient MILP techniques. The results of extensive computational experiments in [17] demonstrate that this approach requires significantly less computational efforts than several commercial MINLP optimization packages. Hence, we apply this method to reformulate $(P_2)$ into its equivalent MILP model.

To perform the reformulation, we first utilize a procedure similar to the Charnes–Cooper transformation [17], [18] except that the variable transformations are only applied to continuous variables. We define variable $u$ and a set of variables $\dot{r}_s$ such that $u = 1/\sum_{l=1}^{L}\sum_{t=1}^{T} p_t^l$ and $\dot{r}_s = r_s/\sum_{l=1}^{L}\sum_{t=1}^{T} p_t^l = r_s u$, where $u$ and $\dot{r}_s$ are positive because $\sum_{l=1}^{L}\sum_{t=1}^{T} p_t^l > 0$ and $r_s > 0$ for all feasible solutions of $(P_1)$. According to this definition, we can transform the fractional objective function to the linear function given by $\sum_{s\in S} r_s/\sum_{l=1}^{L}\sum_{t=1}^{T} p_t^l = \sum_{s\in S}\dot{r}_s$.

We then multiply both sides of all constraints of $(P_2)$ by $u$. The resulting constraints are given in (30)-(54)

$$\sum\nolimits_{l\in IL(i)\cup OL(i)} ux_t^l \le u \quad (\forall i\in N, t\in\Gamma), \tag{30}$$

$$\sum\nolimits_{l\in OL(i),t\in\{1,2,\ldots,T\}} \dot{f}_{s,t}^l = \dot{r}_s T \quad (\forall i\in N, s\in IS(i)), \tag{31}$$

$$\sum\nolimits_{l\in IL(i),t\in\{1,2,\ldots,T\}} \dot{f}_{s,t}^l = \sum\nolimits_{l\in OL(i),t\in\{1,2,\ldots,T\}} \dot{f}_{s,t}^l \quad (\forall i\in N, s\in S-(IS(i)\cup OS(i))), \tag{32}$$

$$r_s^{min} u \le \dot{r}_s \le r_s^{max} u \quad (\forall s\in S) \tag{33}$$

$$0 \le \dot{p}_t^l \le p_{max} u x_t^l \quad (\forall l\in L, t\in\Gamma), \tag{34}$$

$$\dot{I}_t^l = \sum\nolimits_{l'\in L, l'\neq l} h_{(TX_{l'},RX_l)} \dot{p}_t^{l'} \quad (\forall l\in L, t\in\Gamma), \tag{35}$$

$$\sum\nolimits_{k=1}^{K} u\lambda_t^l(k) = u \quad (\forall l\in L, t\in\Gamma), \tag{36}$$

$$\dot{SINR}_t^l = \sum\nolimits_{k=1}^{K} [SINR_t^l(k-1)u\lambda_t^l(k) + \dot{\alpha}_t^l(k)] \quad (\forall l\in L, t\in\Gamma) \tag{37}$$

$$\dot{I}_t^l = \underline{I}_t^l u + \sum\nolimits_{k=1}^{K} \dot{\beta}_t^l(k) \quad (\forall l\in L, t\in\Gamma), \tag{38}$$

$$\sigma W \dot{SINR}_t^l + \dot{w}_t^l = h_l \dot{p}_t^l \quad (\forall l\in L, t\in\Gamma), \tag{39}$$

$$\dot{w}_t^l \ge \underline{I}_t^l \dot{SINR}_t^l + \sum\nolimits_{k=1}^{K}\{SINR_t^l(k-1)\dot{\beta}_t^l(k)\} \quad (\forall l\in L, t\in\Gamma) \tag{40}$$

$$\dot{w}_t^l \ge \underline{I}_t^l \dot{SINR}_t^l + \sum\nolimits_{k=1}^{K}\{SINR_t^l(k-1)\dot{\beta}_t^l(k)\} + \sum\nolimits_{k=1}^{K}\left\{d\dot{\beta}_t^l(k)\right\} + (\overline{I}_t^l - \underline{I}_t^l)\sum\nolimits_{k=1}^{K} \times\{\dot{\alpha}_t^l(k) - du\lambda_t^l(k)\} \quad (\forall l\in L, t\in\Gamma), \tag{41}$$

$$\dot{w}_t^l \le \underline{I}_t^l \dot{SINR}_t^l + \sum\nolimits_{k=1}^{K}\{SINR_t^l(k-1)\dot{\beta}_t^l(k)\} + (\overline{I}_t^l - \underline{I}_t^l)\sum\nolimits_{k=1}^{K}\dot{\alpha}_t^l(k) \quad (\forall l\in L, t\in\Gamma) \tag{42}$$

$$\dot{w}_t^l \le \underline{I}_t^l \dot{SINR}_t^l + \sum\nolimits_{k=1}^{K}\{SINR_t^l(k-1)\dot{\beta}_t^l(k)\} + \sum\nolimits_{k=1}^{K} d\dot{\beta}_t^l(k) \quad (\forall l\in L, t\in\Gamma). \tag{43}$$

$$\sum\nolimits_{s\in S} \dot{f}_{s,t}^l \le \frac{W}{ln2}\dot{c}_t^l \quad (\forall l\in L, t\in\Gamma), \tag{44}$$

$$\dot{c}_t^l \le \sum\nolimits_{k=1}^{K}\{\frac{\dot{\alpha}_t^l(k)}{1+SINR_t^l(k-1)} + u\lambda_t^l(k) \times ln[1+SINR_t^l(k-1)]\} \quad (\forall l\in L, t\in\Gamma) \tag{45}$$

$$\dot{c}_t^l \le \sum\nolimits_{k=1}^{K}\{\frac{\dot{\alpha}_t^l(k) + u\lambda_t^l(k)\,[SINR_t^l(k-1)-\gamma]}{1+\gamma} + u\lambda_t^l(k)\,ln(1+\gamma)\} \quad (\forall l\in L, t\in\Gamma) \tag{46}$$

$$\dot{c}_t^l \le \sum\nolimits_{k=1}^{K}\{\frac{\dot{\alpha}_t^l(k) + u\lambda_t^l(k)\left[SINR_t^l(k-1) - SINR_t^l(k)\right]}{1+SINR_t^l(k)} + u\lambda_t^l(k)\,ln\left[1+SINR_t^l(k)\right]\} \quad (\forall l\in L, t\in\Gamma) \tag{47}$$

$$\dot{c}_t^l \ge \sum\nolimits_{m=1}^{M}\{u\lambda_t^l(k)\,ln[1+SINR_t^l(k-1)] + \frac{ln\left[1+SINR_t^l(k)\right] - ln[1+SINR_t^l(k-1)]}{SINR_t^l(k) - SINR_t^l(k-1)}\} \times \dot{\alpha}_t^l(k) \quad (\forall l\in L, t\in\Gamma), \tag{48}$$

$$0 \le \dot{SINR}_t^l \le h_l p_{max} u/\sigma W \quad (\forall l\in L, t\in\Gamma), \tag{49}$$

$$0 \le \dot{f}_{s,t}^l \quad (\forall l\in L, s\in S, t\in\Gamma), \tag{50}$$

$$0 \le \dot{I}_t^l \le [p_{max}\sum\nolimits_{l'\in L, l'\neq l} h_{(TX_{l'},RX_l)}]u \quad (\forall l\in L, t\in\Gamma), \tag{51}$$

$$0 \le \dot{c}_t^l \le ln\left(1 + h_l p_{max}/\sigma W\right) u \quad (\forall l\in L, t\in\Gamma), \tag{52}$$

$$0 \le \dot{\alpha}_t^l(k) \le du\lambda_t^l(k) \quad (\forall l\in L, t\in\Gamma, k\in\Psi), \tag{53}$$

$$0 \le \dot{\beta}_t^l(k) \le \left(\overline{I}_t^l - \underline{I}_t^l\right) u\lambda_t^l(k) \quad (\forall l\in L, t\in\Gamma, k\in\Psi), \tag{54}$$

where $\dot{p}_t^l = up_t^l, \dot{f}_{s,t}^l = uf_{s,t}^l, \dot{I}_t^l = uI_t^l, \dot{SINR}_t^l = uSINR_t^l$, $\dot{w}_t^l = uw_t^l, \dot{c}_t^l = uc_t^l, \forall l\in L, t\in\Gamma$, and $\dot{\alpha}_t^l(k) = u\alpha_t^l(k)$, $\dot{\beta}_t^l(k) = u\beta_t^l(k), \forall l\in L, t\in\Gamma, k\in\Psi$.

Additionally, as $u = 1/\sum_{l=1}^{L}\sum_{t=1}^{T} p_t^l$, we have the following constraint

$$\sum\nolimits_{l=1}^{L}\sum\nolimits_{t=1}^{T}\dot{p}_t^l = 1. \tag{55}$$

Note that nonlinear terms in (30)–(55) are $ux_t^l$ and $u\lambda_t^l(k)$, which are products of a binary variable and a continuous variable. We can exactly linearize such nonlinear terms using Glover's linearization scheme [17]. To achieve this end, we first introduce auxiliary variables $\dot{q}_t^l = ux_t^l$ and $\dot{v}_t^l(k) = u\lambda_t^l(k)$, and rewrite (30), (34), (36), (37), (41), (45)–(48), (53) and (54) as (56)-(66).

$$\sum\nolimits_{l\in IL(i)\cup OL(i)} \dot{q}_t^l \le u \quad (\forall i\in N, t\in\Gamma), \tag{56}$$

$$0 \le \dot{p}_t^l \le p_{max}\dot{q}_t^l \quad (\forall l\in L, t\in\Gamma), \tag{57}$$

$$\sum\nolimits_{k=1}^{K}\dot{v}_t^l(k) = u \quad (\forall l\in L, t\in\Gamma), \tag{58}$$

$$\dot{SINR}_t^l = \sum\nolimits_{k=1}^{K}[SINR_t^l(k-1)\dot{v}_t^l(k) + \dot{\alpha}_t^l(k)] \quad (\forall l\in L, t\in\Gamma), \tag{59}$$

$$\dot{w}_t^l \geq \underline{I}_t^l \dot{SINR}_t^l + \sum_{k=1}^{K} \{SINR_t^l(k-1)\dot{\beta}_t^l(k)\} + \sum_{k=1}^{K} \left\{d\dot{\beta}_t^l(k)\right\} + (\overline{I_t^l} - \underline{I}_t^l) \sum_{k=1}^{K} \{\dot{\alpha}_t^l(k) - d\dot{v}_t^l(k)\} \quad (\forall l \in L, t \in \Gamma) \tag{60}$$

$$\dot{c}_t^l \leq \sum_{k=1}^{K} \{\frac{\dot{\alpha}_t^l(k)}{1 + SINR_t^l(k-1)} + \dot{v}_t^l(k) \times ln[1 + SINR_t^l(k-1)]\} \quad (\forall l \in L, t \in \Gamma), \tag{61}$$

$$\dot{c}_t^l \leq \sum_{k=1}^{K} \{\frac{\dot{\alpha}_t^l(k) + \dot{v}_t^l(k)\left[SINR_t^l(k-1) - \gamma\right]}{1+\gamma} + \dot{v}_t^l(k) ln(1+\gamma)\} \quad (\forall l \in L, t \in \Gamma) \tag{62}$$

$$\dot{c}_t^l \leq \sum_{k=1}^{K} \left\{\frac{\dot{\alpha}_t^l(k) + \dot{v}_t^l(k)\left[SINR_t^l(k-1) - SINR_t^l(k)\right]}{1 + SINR_t^l(k)} + \dot{v}_t^l(k)\, ln\left[1 + SINR_t^l(k)\right]\right\} \quad (\forall l \in L, t \in \Gamma), \tag{63}$$

$$\dot{c}_t^l \geq \sum_{m=1}^{M} \{\dot{v}_t^l(k) ln[1 + SINR_t^l(k-1)] + \frac{ln\left[1+SINR_t^l(k)\right] - \ln[1+SINR_t^l(k-1)]}{SINR_t^l(k) - SINR_t^l(k-1)}\} \times \dot{\alpha}_t^l(k) \quad (\forall l \in L, t \in \Gamma), \tag{64}$$

$$0 \leq \dot{\alpha}_t^l(k) \leq d\dot{v}_t^l(k) \quad (\forall l \in L, t \in \Gamma, k \in \Psi), \tag{65}$$

$$0 \leq \dot{\beta}_t^l(k) \leq (\overline{I_t^l} - \underline{I}_t^l)\dot{v}_t^l(k) \quad (\forall l \in L, t \in \Gamma, k \in \Psi). \tag{66}$$

Next, we append the new constraints given below

$$0 \leq \dot{q}_t^l \leq H \cdot x_t^l \quad (\forall l \in L, t \in \Gamma), \tag{67}$$

$$u - H \cdot (1 - x_t^l) \leq \dot{q}_t^l \leq u \quad (\forall l \in L, t \in \Gamma), \tag{68}$$

$$0 \leq \dot{v}_t^l(k) \leq H \cdot \lambda_t^l(k) \quad (\forall l \in L, t \in \Gamma, k \in \Psi), \tag{69}$$

$$u - H \cdot \left(1 - \lambda_t^l(k)\right) \leq \dot{v}_t^l(k) \leq u \quad (\forall l \in L, t \in \Gamma, k \in \Psi), \tag{70}$$

where $H$ is a sufficiently large number. Constraints (67) and (68) imply that if $x_t^l$ is zero, $\dot{q}_t^l$ should be zero. Otherwise, if $x_t^l$ is one, $\dot{q}_t^l$ should be equal to $u$. Hence, (67)–(68) are linearization constraints for $\dot{q}_t^l = ux_t^l$. Similarly, (69)–(70) are formulations to linearize $\dot{v}_t^l(k) = u\lambda_t^l(k)$.

After performing the above transformation, we can reconstruct an equivalent MILP model ($P_3$) for ($P_2$) as follows

$$Max \sum_{s \in S} \dot{r}_s \tag{$P_3$}$$

subject to constraints (31)-(33), (35), (38)-(40), (42)-(44), (49)-(52), (55)-(70),

$$u \geq 0; x_t^l \in \{0, 1\}, \dot{p}_t^l \geq 0, \dot{I}_t^l \geq 0, \dot{SINR}_t^l \geq 0, \dot{w}_t^l \geq 0,$$
$$\dot{f}_{s,t}^l \geq 0 \quad (\forall l \in L, s \in S, t \in \Gamma); \lambda_t^l(k) \in \{0, 1\},$$
$$\dot{c}_t^l \geq 0, \dot{q}_t^l \geq 0 \quad (\forall l \in L, t \in \Gamma); \dot{r}_s \geq 0 \quad (\forall s \in S);$$
$$\dot{\alpha}_t^l(k) \geq 0, \dot{\beta}_t^l(k) \geq 0, \dot{v}_t^l(k) \geq 0 \quad (\forall l \in L, t \in \Gamma, k \in \Psi).$$

In [17], the authors have proven that under this transformation, both MILFP and MILP models are mathematically equivalent. They also have proven that this transformation preserves the one-to-one mapping relationship between solutions of MILFP and MILP problems. We refer readers to [17] for details of the proofs. Manifestly, based on these properties, ($P_3$) is equivalent to ($P_2$) and can serve as the UB problem of ($P_1$). Hence, one can easily find the UB of ($P_1$) using state-of-the-art MILP solvers to optimize ($P_3$).

### C. LOWER BOUNDING STRATEGY

Because any feasible solution of ($P_1$) yields a valid LB, we propose a scheme using the optimal solution of ($P_2$) as a starting point to search for a good feasible solution of ($P_1$). We first optimize ($P_3$) globally and transform its solution to that of ($P_2$). The resulting values of $x_t^l$ and $p_t^l(\forall l \in L, t \in \Gamma)$ always provide a set of feasible solutions to the same variables in ($P_1$). We thus fix $x_t^l$ and $p_t^l$ variables at their optimal values from ($P_2$). Given feasible $x_t^l$ and $p_t^l$, we can compute values of $SINR_t^l$ ($\forall l \in L, t \in \Gamma$) using (12) and (13). Constraints (3) and (10) are also satisfied and thus can be omitted. Furthermore, the denominator of (9) becomes a constant. Thus, optimizing (9) is equivalent to maximizing its numerator. Consequently, we can equivalently transform ($P_1$) to LP problem ($P_4$) as follows

$$Max \sum_{s \in S} r_s \tag{$P_4$}$$

subject to constraints (5)-(8),

$$r_s \geq 0 \ (\forall s \in S); \quad f_{s,t}^l \geq 0 \ (\forall s \in S, l \in L, t \in \Gamma).$$

If ($P_4$) is feasible, its solution is also feasible to $f_{s,t}^l$ and $r_s$ variables in ($P_1$). Therefore, we can find a feasible solution to ($P_1$) by combining values of $x_t^l$, $p_t^l$, $f_{s,t}^l$ with $r_s$ ($\forall s \in S, l \in L, t \in \Gamma$) using the above scheme. Otherwise, it exhibits there does not exist a session rate satisfying the minimum sustained and peak rate requirements for all sessions. In this case, we do not update the LB.

### D. BRANCHING RULE

To partition the current subproblem, we have to choose a branching variable and corresponding branching point. Because both bilinear terms and *log* functions are relaxed in ($P_2$), we select variables involved with these relaxations for branching. Therefore, we have $I_t^l$ and $SINR_t^l$ as branching candidates. We define the relaxation error of the bilinear term and *log* function as $RE_{l,t}^w = \left|\widetilde{w_t^l} - \widetilde{I_t^l}\widetilde{SINR_t^l}\right|$ and $RE_{l,t}^c = \left|\widetilde{c_t^l} - ln\left(1 + \widetilde{SINR_t^l}\right)\right|$, respectively, where $\widetilde{w_t^l}$, $\widetilde{I_t^l}$, $\widetilde{SINR_t^l}$, and $\widetilde{c_t^l}$ are values of $w_t^l$, $I_t^l$, $SINR_t^l$, and $c_t^l$ variables, respectively, at the solution of ($P_2$).

We also define the largest relaxation error (*LRE*) as $max\left\{RE_{l,t}^w, RE_{l,t}^c; \forall l \in L, t \in \Gamma\right\}$. To decide the exact branching variable, we select the one associated with the *LRE*, because dividing on such a variable most decreases relaxation errors of ($P_2$). In case the *LRE* coincides with a *log* function, we let the branching variable be the

**TABLE 1. Locations of nodes.**

| Node | Location | Node | Location | Node | Location | Node | Location |
|---|---|---|---|---|---|---|---|
| 1 | (19, 146) | 6 | (89, 236) | 11 | (119, 190) | 16 | (208, 15) |
| 2 | (25, 81) | 7 | (100, 32) | 12 | (157, 82) | 17 | (218, 46) |
| 3 | (12, 247) | 8 | (102, 8) | 13 | (161, 241) | 18 | (225, 63) |
| 4 | (28, 174) | 9 | (104, 128) | 14 | (171, 159) | 19 | (231, 193) |
| 5 | (3, 9) | 10 | (111, 63) | 15 | (199, 127) | 20 | (245, 227) |

corresponding $SINR_t^l$ and the branching point be $\widetilde{SINR}_t^l$. However, if the *LRE* corresponds to a bilinear product, we need to decide a single variable for branching. Specifically, we choose branching variable $v_b$ as (71), where $\overline{I_t^l}$ and $\underline{I_t^l}$ are the current upper and lower bounds of $I_t^l$, respectively, and $\overline{SINR_t^l}$ and $\underline{SINR_t^l}$ are the current upper and lower bounds of $SINR_t^l$, respectively. After choosing a specific $v_b$ for branching, we partition at $\widetilde{v}_b$. Note that in (71), we choose the $v_b$ whose relaxation solution value is closest to the midpoint of its current range. This is intended to cause a large disturbance in the created subproblem and decrease the UB as much as possible.

After deciding the branching variable, we split the current search space into two subregions along the branching variable axis at the branching point and create two subproblems. When solving either of the two subproblems, we replace the original relaxations of both bilinear terms and *log* functions with stronger ones by exploiting the range reduction.

## IV. NUMERICAL RESULTS AND DISICUSSIONS

In this section, we describe the evaluation of optimal EE of the MANET using our proposed BB algorithm. We first compare its computational efficiency versus a reference BB algorithm. In the reference BB algorithm, we used the relaxation methods proposed in [1]–[3] to establish the UB problem. Specifically, we used the RLT to approximate the bilinear term and the SCH to bound the log function. We created another MILFP model accordingly, which was then transformed into its equivalent MILP problem using the same technique described in Section 3.2.3. The other elements of the reference algorithm were the same as those of our proposed BB algorithm. Subsequently, we experiment on optimal EE over different values of $T$ and various routing strategy to gain more insights on how power control, traffic scheduling and routing influence the optimal solution.

For the proposed and reference BB algorithms, we used CPLEX [30] to solve the UB and LB problems (i.e., ($P_3$) and ($P_4$)). We ran the programs in a machine with two Intel Xeon 3.3 GHz processors. Each processor had eight cores. Since CPLEX supports a parallel optimization mode, we solved ($P_3$) and ($P_4$) using all available cores. We consider a MANET with twenty nodes randomly located in a $250 \times 250$ m$^2$ region. Table 1 shows the locations of these nodes. There are four sessions among the nodes. Table 2 presents the source node, destination node, $r_s^{min}$, and $r_s^{max}$ for the sessions. We set maximum transmission power to $p_{max} = 100$ mW and channel bandwidth $W = 5$ MHZ. We assume propagation gain $h_l = \vartheta d_l^{-4}$, where $d_l$ is the distance between transmitter and receiver of link $l$, and $\vartheta = 0.002$ is a constant characterizing the antenna gain and average channel attenuation [31].

**TABLE 2. Traffic profiles of sessions.**

| Session | Source node | Destination node | $r_s^{min}$ (Mbps) | $r_s^{max}$ (Mbps) |
|---|---|---|---|---|
| 1 | 20 | 3 | 1.4 | 7.6 |
| 2 | 18 | 1 | 1 | 4.5 |
| 3 | 2 | 14 | 2.5 | 9.7 |
| 4 | 5 | 16 | 0.8 | 5.9 |

### A. COMPUTATION EFFICIENCY EVALUATION

Table 3 illustrates computational outcomes for the $T = 5$ case, including the reference BB algorithm and our proposed BB algorithm with different partitioning levels $K$. We ran all experimental instances with a CPU time limit of 40 minutes. In Table 3, columns $UB_{root}$ and $CPU_{root}$ are the UB and computational time of the root node of the BB tree, respectively. Columns *UB* and *LB* show the best upper and lower bounds found within the time limit, respectively. The remaining columns denote the optimality gap and the number of investigated BB nodes when the time limit is reached.

From these results, we first observe that at the root node, as discussed in Section 3.2.1, our proposed scheme with different values of $K$ always provides a more rigorous UB than that of the reference BB algorithm, and the UB tightens as $K$ enlarges. Usually, larger $K$ also leads to fewer nodes required to explore in the BB procedure as it provides tighter relaxations. However, we also find that the computational time of the root node grows with an increased value of $K$, because the number of variables and constraints in ($P_3$) increases when $K$ augments. Therefore, larger $K$ increases the size and solution time of the relaxation problem. Apparently, an important tradeoff exists between the computational time of each node and the number of nodes requiring investigation during the BB process. One thus has to carefully select the value of $K$ to lessen the solution time of the BB algorithm as much as possible. We tested many cases to select a proper value for $K$, and we observed that our algorithm minimizes the solution time when three partitions are used. Therefore, we used $K = 3$ for all simulations in this section. Second, our proposed scheme with different values of $K$ always achieved a smaller optimality gap and better feasible solution within the same time limit as compared with the reference BB algorithm. The optimality gap of our proposed approach with $K = 3$ and the reference algorithm

**TABLE 3. Computational outcomes for $T = 5$.**

| | $UB_{root}$ | $CPU_{root}$ (sec) | $UB$ | $LB$ | % gap | No. nodes |
|---|---|---|---|---|---|---|
| Reference | 7.645*10^8 | 246 | 6.976*10^8 | 3.593*10^8 | 48.49 | 14 |
| Proposed $K=3$ | 5.928*10^8 | 1029 | 5.228*10^8 | 4.771*10^8 | 8.74 | 3 |
| Proposed $K=5$ | 5.786*10^8 | 1395 | 5.386*10^8 | 4.592*10^8 | 14.74 | 2 |

were 8.74% and 48.49%, respectively. The best feasible solution found by our proposed algorithm with $K = 3$ and the reference algorithm were $4.771{*}10^8$ and $3.593{*}10^8$, respectively. This shows that our proposed method respectively decreased the optimality gap 81.98% and increased the best feasible solution 32.79%, which achieved significant computational improvements.

Note that the reason that our proposed scheme took tens of minutes to solve ($P_1$) is due to its NP-hard nature. For the MANET considered in this simulation, the number of $\boldsymbol{x}_t^l$ variables is 1900 and thus there are $2^{1900}$ ($\approx 10^{572}$) possible combinations of link activations. In every possible combination (i.e., fixed $\boldsymbol{x}_t^l$ value, $\forall l \in L, t \in \Gamma$), ($P_1$) becomes a nonconvex nonlinear programming (NLP) problem that is NP-hard. This means if a brute force method is used, one has to solve $10^{572}$ NP-hard problems. It is very difficult to solve an NP-hard problem because we cannot find a polynomial time algorithm approaching such a problem, let alone tackle $10^{572}$ NP-hard problems, which is far too many to realistically address. However, our proposed scheme (with $K = 3$) can obtain a solution which is at least 91.26% optimal within 40 minutes. Obviously, compared with the reference algorithm and a brute force method, our proposed scheme can provide good solutions while preserving high computational efficiency.

Table 4 presents details of the optimal solution obtained using our proposed algorithm for the above $T = 5$ case. As shown in Table 4.1, at time slot 1, there were two active links, namely, links $20 \rightarrow 13$ and $2 \rightarrow 9$ corresponding to sessions 1 and 3, respectively. The transmission powers of the two links were 1.16 mw and 4.17 mw, respectively. Since both links transmitted concurrently, they interfered with each other. Hence, the resulting capacities of the two links were 7.36 Mbps and 12.96 Mbps, respectively. The capacities also limited the flow rates of the two links as 7.35 Mbps and 12.95 Mbps, respectively. Similarly, Tables 4.2 through 4.5 display the results of time slots 2 to 5, respectively. Note that we only show outcomes of active links for the five time slots in Table 4. Fig. 3 illustrates the routing topology corresponding to Table 4. From Fig. 3, we can find the routing path for each session. For instance, the path of session 1 consisted of 3 links, namely, $20 \rightarrow 13$, $13 \rightarrow 6$, and $6 \rightarrow 3$, which were scheduled to transmit at

**TABLE 4. Table 4. Optimization results of different time slots. Table 4.1. Optimal EE plan for time slot 1. Table 4.2. Optimal EE plan for time slot 2. Table 4.3. Optimal EE plan for time slot 3. Table 4.4. Optimal EE plan for time slot 4. Table 4.5. Optimal EE plan for time slot 5.**

| Active link | Transmission power (mw) | Capacity (Mbps) | Flow rate (Mbps) |
|---|---|---|---|
| 20→13 | 1.16 | 7.36 | 7.35 |
| 2→9 | 4.17 | 12.96 | 12.95 |

Table 4.1.

| Active link | Transmission power (mw) | Capacity (Mbps) | Flow rate (Mbps) |
|---|---|---|---|
| 13→6 | 0.51 | 7.43 | 7.35 |
| 18→12 | 0.33 | 5.68 | 5.6 |

Table 4.2.

| Active link | Transmission power (mw) | Capacity (Mbps) | Flow rate (Mbps) |
|---|---|---|---|
| 6→3 | 0.68 | 7.35 | 7.35 |
| 12→9 | 0.43 | 5.6 | 5.6 |

Table 4.3.

| Active link | Transmission power (mw) | Capacity (Mbps) | Flow rate (Mbps) |
|---|---|---|---|
| 5→8 | 1.68 | 4.86 | 4.85 |
| 9→14 | 1.64 | 12.97 | 12.95 |

Table 4.4.

| Active link | Transmission power (mw) | Capacity (Mbps) | Flow rate (Mbps) |
|---|---|---|---|
| 8→16 | 1.45 | 4.87 | 4.85 |
| 9→1 | 0.84 | 5.63 | 5.6 |

Table 4.5.

time slot 1, 2 and 3, respectively. The instantaneous flow rates of the three links were all 7.35 Mbps, which brought about the average session rate being equal to 1.47 Mbps. One can also discover that the average session rates of sessions 2 through 4 were 1.12 Mbps, 0.97 Mbps, and 2.59 Mbps, respectively. The total depleted power was 12.89mw. Hence, the optimal EE was $4.771{*}10^8$ (bits/Joule).

### B. THE IMPACT OF TRAFFIC SCHEDULING

Fig.4 shows the curve of optimal EE versus different numbers of $T$ for the same example MANET considered in Fig. 3. We observe that when $T$ increased, optimal EE enlarged up to a maximum point that occurred at $T = 5$. This was because each node had more choices of traffic scheduling and routing to mitigate interference from other nodes.

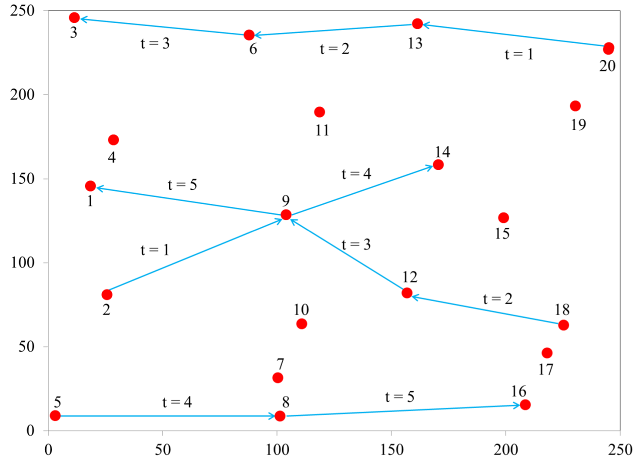

**FIGURE 3.** Routing topology for $T = 5$.

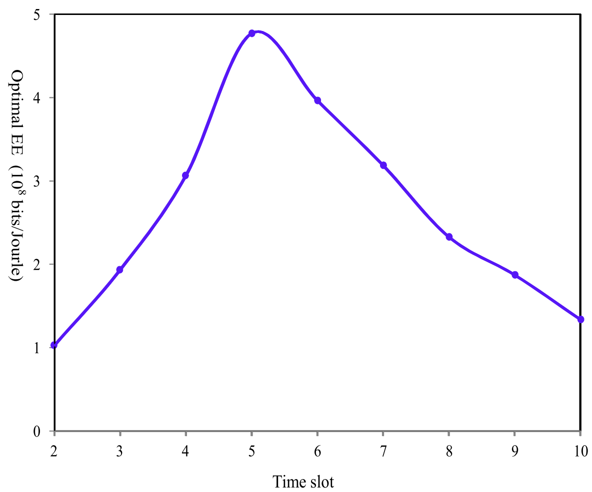

**FIGURE 4.** Optimal EE versus different numbers of $T$.

Therefore, EE increased as $T$ augmented. However, beyond the maximum point, optimal EE decreased as $T$ increased. This was because as $T$ enlarged, each session had to send more traffic to satisfy the minimum sustained and peak rate requirements. Hence, every active node consumed more power to send the increased amount of traffic. However, when EE reached its maximal point, interference of each active node had been diminished to be very small. Therefore, when $T$ increased, it was hard to further reduce interference of each active node. Nevertheless, each active node must spend more power to transmit the increased data, thus magnifying dissipated power of the entire network. Because the total session rate was limited to a fixed range and the total amount of power increased as $T$ augmented, optimal EE lessened as $T$ enlarged. This result indicates that it is essential to select a suitable value of $T$ to enable the network operating around the optimal EE. To the best of our knowledge, this is still an open issue. Thus, it is also a direction for future research on optimizing EE of MANETs.

**TABLE 5.** Optimization results of the minimum hop-count routing.

| Time slot | Active link | Transmission power (mw) | Capacity (Mbps) | Flow rate (Mbps) |
|---|---|---|---|---|
| 1 | $20 \rightarrow 13$ | 1.03 | 7.7 | 7.7 |
| 2 | $13 \rightarrow 3$ | 9.72 | 7.7 | 7.7 |
| 3 | $18 \rightarrow 1$ | 28.29 | 5.45 | 5.45 |
| 4 | $2 \rightarrow 14$ | 36.93 | 12.65 | 12.65 |
| 5 | $5 \rightarrow 16$ | 14.85 | 4.3 | 4.3 |

### C. THE INFLUENCE OF ROUTING STRATEGY

Due to the simplicity of implementation in practice and the lessened wastage of MANET resources such as network links and buffers, the minimum hop count has become the most widespread metric used by MANET routing protocols. It has been chosen as the default metric for many routing protocols, for instance, [19]–[23], etc. To investigate the influence of routing methodologies on the optimal EE, we fixed the value of $T$ as 5 and selected the minimum hop count as the performance comparison index. In Fig. 3, if the minimum hop-count routing was used, the destination node of each session could be reached from the source node with 1 hop. However, we had 5 time slots. We thus let the session 1 having the longest distance among the 4 sessions use 2 hops that occupied 2 time slots respectively. For the other sessions, each session used one time slot to send data between source and destination nodes (i.e. one-hop routing).[1] Table 5 presents the optimal results for the minimum hop-count routing. The total session rate and power consumption were 6.02 Mbps and 90.82 mw, respectively. The optimal EE thus was $6.629{*}10^7$ (bits/Joule). Compared to the results shown in Table 4, our proposed cross-layer optimization approach improved the EE by 7.87 times over the minimum hop-count routing, which revealed a substantial enhancement. This was because the minimum hop-count routing usually prefers longer distance for each hop, which causes higher power consumption for transmitting the same amount of data.

## V. CONCLUSION

In this paper, we addressed the EE optimization problem for MANETs, jointly considering routing, traffic scheduling, and power control. According to the cross-layer design principle, we formulated this problem as a nonconvex MINLP model that is intrinsically NP-hard. We proposed a customized BB algorithm for the global optimization of the problem by exploiting the specific structure of this type of model. In our algorithm, we approximated bilinear terms with powerful PLRs and *log* functions with PCHs to build a MILFP model serving as a tight UB problem. We then effectively transformed the MILFP model to its equivalent MILP formulation, allowing for the global optimization of the MILFP using efficient MILP methods. We obtained tight UBs on the global optimum by solving the resulting

[1] Identifying the allocation of time slot in advance is the same as knowing values of $x_t^l$ variables. Furthermore, constraints (5) and (6) can also be simplified if the routing protocol is already known. Hence, one can equivalently transform ($P_1$) to a nonconvex NLP problem that can be globally optimized by modifying our proposed BB algorithm.

MILP-based relaxations. Our proposed BB algorithm also incorporated a novel lower bounding strategy and branching rule designed to accelerate convergence.

We presented our numerical results of applying our proposed BB algorithm to EE optimization of a MANET. We also compared the efficiency of our proposed BB algorithm with another scheme. Our algorithm performed quite well with respect to computational complexity. In the future, we aim to concentrate on developing distributed protocols and algorithms that can be realistically implemented to optimize EE of MANETs. Moreover, we plan to compare the performance of distributed protocols and mechanisms with that found using the global optimization technique of this paper, thus encouraging the development of novel distributed protocols and algorithms to improve EE of MANETs.

## REFERENCES

[1] Y. Shi, Y. T. Hou, S. Kompella, and H. D. Sherali, "Maximizing capacity in multihop cognitive radio networks under the SINR model," *IEEE Trans. Mobile Comput.*, vol. 10, no. 7, pp. 954–967, Jul. 2011.

[2] Y. Shi, Y. T. Hou, and H. D. Sherali, "Cross-layer optimization for data rate utility problem in UWB-based ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 6, pp. 764–777, Jun. 2008.

[3] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, "Bridging the gap between protocol and physical models for wireless networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1404–1416, Jul. 2013.

[4] D. Feng, C. Jiang, G. Lim, L. J. Cimini, Jr., G. Feng, and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 167–178, first quarter 2013.

[5] C. Isheden, Z. Chong, E. Jorswieck, and G. Fettweis, "Framework for link-level energy efficiency optimization with informed transmitter," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2946–2957, Aug. 2012.

[6] G. Miao, N. Himayat, and G. Y. Li, "Energy-efficient link adaptation in frequency-selective channels," *IEEE Trans. Commun.*, vol. 58, no. 2, pp. 545–554, Feb. 2010.

[7] S. Gao, L. Qian, and D. Vaman, "Distributed energy efficient spectrum access in cognitive radio wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 10, pp. 5202–5213, Oct. 2009.

[8] F. Meshkati, H. V. Poor, S. C. Schwartz, and N. B. Mandayam, "An energy-efficient approach to power control and receiver design in wireless data networks," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1885–1894, Nov. 2005.

[9] F. Meshkati, H. V. Poor, S. C. Schwartz, and R. V. Balan, "Energy-efficient resource allocation in wireless networks with quality-of-service constraints," *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3406–3414, Nov. 2009.

[10] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291–303, Feb. 2002.

[11] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in OFDMA systems with large numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3292–3304, Sep. 2012.

[12] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in multi-cell OFDMA systems with limited backhaul capacity," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3618–3631, Oct. 2012.

[13] D. W. K. Ng, E. S. Lo, and R. Schober, "Wireless information and power transfer: Energy efficiency optimization in OFDMA systems," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6352–6370, Dec. 2013.

[14] K. T. K. Cheung, S. Yang, and L. Hanzo, "Achieving maximum energy-efficiency in multi-relay OFDMA cellular networks: A fractional programming approach," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 2746–2757, Jul. 2013.

[15] Q. Zhao and L. Tong, "Energy efficiency of large-scale wireless networks: Proactive versus reactive networking," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 5, pp. 1100–1112, May 2005.

[16] J. Zhang, L. Fei, Q. Gao, and X.-H. Peng, "Energy-efficient multihop cooperative MISO transmission with optimal hop distance in wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3426–3435, Oct. 2011.

[17] D. Yue, G. Guillén-Gosálbez, and F. You, "Global optimization of large-scale mixed-integer linear fractional programming problems: A reformulation-linearization method and process scheduling applications," *AIChE J.*, vol. 59, no. 11, pp. 4255–4272, 2013.

[18] E. B. Bajalinov, *Linear-Fractional Programming: Theory, Methods, Applications and Software*. New York, NY, USA: Springer, 2003.

[19] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, document RFC 3626, IETF, Oct. 2003;. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt.

[20] D. Johnson, Y. Hu, and D. Maltz, *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*, document RFC 4728, IETF, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc4728.txt.

[21] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. SIGCOMM*, Oct. 1994, pp. 234–244.

[22] C. E. Perkins, E. M. Royer, and S. Das, "Ad hoc on-demand distance vector routing," document RFC 3561, IETF, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt.

[23] V. D. Park and M. S. Corson, *Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification*, Internet Draft, Jul. 2001. [Online]. Available: http://tools.ietf.org/id/draft-ietf-manet-tora-spec-04.txt.

[24] C. E. Gounaris, R. Misener, and C. A. Floudas, "Computational comparison of piecewise—Linear relaxations for pooling problems," *Ind. Eng. Chem. Res.*, vol. 48, no. 12, pp. 5742–5766, 2009.

[25] R. Misener and C. A. Floudas, "Global optimization of large-scale generalized pooling problems: Quadratically constrained MINLP models," *Ind. Eng. Chem. Res.*, vol. 49, no. 11, pp. 5424–5438, 2010.

[26] M. Jünger *et al.*, Eds., *50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art*. New York, NY, USA: Springer, 2010.

[27] R. M. Lima and I. E. Grossmann, "Computational advances in solving mixed integer linear programming problems," in *Proc. AIDIC*, 2011, pp. 151–160.

[28] T. Koch *et al.*, "MIPLIB 2010: Mixed integer programming library version 5," *Math. Program. Comput.*, vol. 3, no. 2, pp. 103–163, 2011.

[29] E. Balas, "Disjunctive programming and a hierarchy of relaxations for discrete optimization problems," *SIAM. J. Algebraic Discrete Methods*, vol. 6, no. 3, pp. 466–486, 1985.

[30] *IBM ILOG CPLEX*, accessed on Jan. 2014. [Online]. Available: http://www01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[31] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

**WEN-KUANG KUO** received the B.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1994, the M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1996, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 2003. Since 2003, he has been with the Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, where he is currently an Assistant Professor. His current research interests include multimedia transmission over wireless networks, cross-layer optimization of wireless communication systems, protocol design and performance evaluation, and heterogeneous wireless/wireline network integration.

**SHU-HSIEN CHU** received the B.S. degree in mathematics and the M.S. degree in applied mathematics from National Cheng Kung University, Tainan, Taiwan, in 2007 and 2009, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Minnesota, Minneapolis, USA. His current research interests include mathematical modeling, optimization, magnetic resonance imaging, image processing, machine learning, signal processing, network optimization, and wireless communication.