Cross-Layer Optimization for Data Rate Utility Problem in UWB-Based Ad Hoc Networks

Yi Shi, *Member, IEEE*, Y. Thomas Hou, *Senior Member, IEEE*, and Hanif D. Sherali

Abstract—There is growing interest in employing ultra-wideband (UWB) communication systems at the physical layer for multihop wireless networks. Recent efforts show that networking problems involving UWB systems should follow a cross-layer approach with consideration at multiple layers. Due to the nonlinear nature of the optimization problem, there are very limited theoretical results for this important problem. In this paper, we address this problem by considering a UWB-based ad hoc network. We study how to maximize capacity (in the form of a data rate utility) for a set of communication sessions. Via a cross-layer approach, we formulate this utility maximization problem into a nonlinear programming (NLP) problem, which takes into consideration routing, scheduling, and power control. We develop a solution procedure based on the so-called *branch-and-bound* framework. Within this framework, we employ a powerful optimization technique called reformulation linearization technique (RLT). We use numerical results to validate the efficacy of this solution procedure and offer insights on UWB-based ad hoc networks. This work provides a theoretical result for the achievable performance bound for a UWB-based ad hoc network.

Index Terms—Ultra-wideband (UWB), cross-layer optimization, nonlinear optimization, wireless networks.

✦

1 INTRODUCTION

ULTRA-WIDEBAND (UWB) systems transmit and receive signals through an extremely short-duration burst of radio frequency (RF) energy. The resultant waveforms have an extremely wide frequency band, and it is difficult to determine their actual RF center frequency (that is, they are "carrier free"). UWB waveforms have some rather unique properties and are suitable for certain military and commercial applications, including tactical handheld and network LPI/D radios, non-LOS LPI/D groundwave communications, precision geolocation systems, high-speed wireless LANs, collision avoidance sensors, and intelligent tags [9], [13], [17].

Recently, there has been growing interest in applying a UWB-based system for multihop ad hoc networks [4], [11], [15], [16], [22]. Due to the complexity of such a problem (that is, nonconvex), there are very limited theoretical solutions. In this paper, we address this theoretical problem following an optimization approach. We consider a set of source-destination communication pairs (which we call sessions) in a UWB-based ad hoc network. For each session, data is generated at the source node and must be relayed to its destination node (via multiple hops if needed). The objective is to maximize the total utility of these active sessions, where a session's utility is defined as the log function of its bit rate [15].

As expected, this optimization problem involves issues from multiple layers, that is, routing, scheduling, and power

control. The routing problem at the network layer considers what set of paths the data (from a source node) should take to its destination node, the link-layer scheduling component deals with how to use time slots for transmission and reception, and the physical-layer power control component considers how much transmission power a node should use in a particular time slot.

This problem was first studied by Radunovic and Le Boudec in [15]. The cross-layer optimization space also consists of routing, scheduling, and power control. Although Radunovic and Le Boudec were able to formulate the problem into a nonlinear program (NLP), they were not able to solve the optimization problem. Instead, they decoupled the joint optimization problem into subproblems based on layers, that is, a layer-decoupled approach. Such an approach is suboptimal to start with and is likely to be far away from the optimum in some cases. Further, under Radunovic and Le Boudec's layer-decoupled approach, their proposed solutions for routing and scheduling are heuristics instead of an exact solution. Specifically, for routing, their proposed solution is based on single-path routing. Single-path routing is clearly suboptimal when compared to the general case where flow-splitting/multi-path routing are allowed, as considered in this paper. For scheduling, they proposed a heuristic solution based on the so-called "exclusion region" property, which is based on the observation of a simple ring topology network rather than rigorous proof for a general network topology setting. As a result of such a heuristic approach, the results in [15] cannot offer any theoretical guarantee. Therefore, a formal mathematical solution procedure that can provide a guaranteed solution (instead of heuristics) to this cross-layer optimization problem remains an open problem. In this paper, we will develop a formal solution procedure to fill this gap.

We first formulate the problem into a mixed-integer nonpolynomial programming problem. After some reformulation by removing integer components and nonpolynomial components in the constraints and exploiting the linear

- *Y. Shi and Y. Thomas Hou are with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061. E-mail: {yshi, thou}@vt.edu.*
- *H.D. Sherali is with the Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061. E-mail: hanifs@vt.edu.*

rate-SINR property that is unique to UWB, we are able to obtain an NLP formulation. The main contribution of this paper is the development of a solution procedure to this NLP problem based on a *branch-and-bound* framework [12]. This framework includes the following steps:

- First, we develop an upper bound via linear relaxation. For linear relaxation, we employ a powerful technique called *Reformulation-Linearization Technique* (RLT) [19] to handle polynomial terms. The essence of RLT is to replace each polynomial term (that is, product of variables) with one new variable and add a set of new *linear* constraints based on the bounds for partition variables (variables in a nonlinear term). As a result, constraints with polynomial terms can be relaxed with a set of linear constraints. For a nonpolynomial term (that is, a $\log$ term), we propose to use three tangential supports, which constitutes a convex envelope linear relaxation and is very accurate for our problem setting. After these efforts, we can relax the NLP problem with a linear program (LP), whose solution provides an upper bound of the objective function.

- Second, we perform a local search to find a feasible solution and a lower bound. Since a relaxation solution may be infeasible for the original NLP problem, a local search algorithm is needed to obtain a feasible solution. We design such a local search algorithm, and the feasible solution obtained via this algorithm will serve as a lower bound of the objective function.

- Finally, we check whether the above lower and upper bounds are close to each other. If so, we are done. Otherwise, we need to obtain a tighter LP relaxation by further narrowing down the value intervals of partition variables. Specifically, branch-and-bound divides a partition variable's value interval into two intervals. Then, the original problem is divided into two new problems with tighter bounds.

During each iteration of the branch-and-bound procedure, it is important (from a computation perspective) to select a partition variable. We propose a partition variable selection policy not only based on the relaxation error but also based on its relative significance in our problem. It turns out that such a policy can significantly speed up the convergence process of branch-and-bound iterations.

The significance of this work is that it provides a theoretical result on the achievable performance bound for a UWB-based ad hoc network. For practical implementation, distributed algorithms and protocols still remain to be developed. The theoretical results should be used as a performance benchmark or measure for any of these distributed algorithms and protocols (rather than comparing them to some heuristic algorithms). That is, the results obtained by our algorithm provide a theoretical benchmark for the development of distributed algorithms and protocols.

The remainder of the paper is organized as follows: In Section 2, we give details of the network model for our problem and discuss routing, scheduling, and power control components. Section 3 presents the mathematical formulation of the cross-layer optimization problem. In Section 4, we present a solution procedure based on the branch-and-bound framework and give details on how to set initial value intervals for partition variables, how to perform linear relaxation (via RLT and three tangential supports), and how to find a solution to the original problem via a relaxation solution (local search). We also show how to select a partition variable so as to speed up computation. In Section 5, we present numerical results to validate the efficacy of our proposed solution procedure and give insights on the impact of different optimization components. Section 6 reviews related work; Section 7 concludes this paper.

2 NETWORK MODEL AND OPTIMIZATION SPACE

We consider a multihop wireless network consisting of N nodes over a two-dimensional area. Within the network, there are L (unidirectional) communication sessions. Denote the data rate for the lth session as $r(l)$ and $\mathbf{r}$ the rate vector for the L sessions. We say that a rate vector $\mathbf{r}$ is *feasible* if and only if there exists a "solution" such that for each session l, $1 \leq l \leq L$, rate $r(l)$ can be transported from the source node $s(l)$ to the destination node $d(l)$. By "solution," we mean a specific realization of routing, scheduling, and power control in the network that can be employed to achieve the rate vector.

In [15], Radunovic and Le Boudec advocate the use of $\sum_{l=1}^{L} \ln r(l)$ as a utility metric in the network optimization problem. The motivation for this choice is that such a $\log$-based utility function can achieve a good compromise between *fairness* and *efficiency*. A detailed justification of the choice of this $\log$-based utility function can be found in [14]. We adopt this utility function as our optimization objective in this paper. However, we emphasize that the solution procedure developed in this paper is general and can be applied to other definitions of the utility function.

We now examine the search space for a feasible solution such that the total utility can be maximized. As discussed, a solution for a feasible rate vector involves mechanisms from several layers. At the network layer, routing affects the rate vector through the choice of multihop routes from each source to its corresponding destination node. At the link layer, we need to find a scheduling policy for each node such that there is no node sending and receiving within the same time slot. At the physical layer, we need to perform power control for each node such that each link has enough capacity. We now take a closer look at each problem.

2.1 Scheduling

At the link layer, scheduling deals with how to coordinate transmission among the nodes in each "time slot." An important constraint is that a node cannot send and receive data within the same time slot. In this paper, we define *time slot* as a time interval within which each node in the network has a specific transmission behavior. This definition differs from the physical-layer "time frame" concept used for UWB (see, for example, [21]), which is directly tied to physical-layer implementation (that is, impulse generation). The time-slot definition in this paper is specifically tailored for network-layer modeling and problem formulation. Further, the time-slot definition in

this paper also differs from the one used in [15], where a node is allowed to only transmit to one node within a time slot.

Under our time-slot definition, a node is allowed to transmit to multiple nodes. In some sense, our time-slot definition can be viewed as an aggregate of certain time-slot behaviors in [15]. Such an approach for aggregated time-slot definition is extremely useful in mathematical modeling, in the sense that a smaller number of time slots may be sufficient to express a solution. As a result, the problem size may become smaller and require less computation time.

Note that such aggregation formulation does not exclude the time-slot arrangement corresponding to the optimal solution. This is because we attempt to aggregate conventional small time slots *only when* such aggregation is possible (without loss of optimality). Otherwise, aggregation will not be performed. For example, suppose node i transmits to nodes j_1, j_2, and j_3 in time slots 1, 2, and 3, respectively. If within these time slots, none of nodes j_1, j_2, and j_3 transmits, then we can aggregate these three time slots into one. On the other hand, if node j_3 must transmit within time slot 2 (and receive from i in time slot 3), then we will not be able to aggregate all three time slots into one. This is because node j_3 cannot receive and transmit in the same aggregated time slot. Instead, we will only aggregate time slots 1 and 3 into one time slot and leave time slot 2 unchanged. Note that such aggregation process will not lose any scheduling behavior that is allowed under those small time-slot cases and thus will not lose optimality.

Given the number of time slots K, denote t_k as the *normalized length* for time slot k, that is, the length of time slot k over the total length of all different time slots. We have

$$\sum_{k=1}^{K} t_k = 1.$$

2.2 Power Control

The power control problem deals with how much power a node should employ to transmit data in a particular time slot. Denote p_{ij}^k as the power that node i expends in time slot k for sending data to node j. Based on our time-slot definition, although a node cannot send and receive within the same time slot, a node can transmit to multiple nodes within the same time slot. The total power that a node i can expend at time slot k must satisfy the following power limit [22]:

$$\sum_{j \in \mathcal{T}_i} p_{ij}^k \leq P_{\max}, \tag{1}$$

where $\mathcal{T}_i$ is the set of one-hop neighbors of node i. This requirement comes from the power density limitation of UWB, that is, $\frac{g_{\text{nom}} \sum_{j \in \mathcal{T}_i} p_{ij}^k}{W} \leq Q_{\max}$. Thus, we have

$$P_{\max} = \frac{Q_{\max} W}{g_{\text{nom}}}, \tag{2}$$

where $Q_{\max}$ is the maximum allowed transmission power spectral density, g_{nom} is the gain at some fixed nominal distance [16], and W is the spectrum for the UWB network, that is, 7.5 GHz.

A widely used model for power gain is

$$g_{ij} = d_{ij}^{-\alpha}, \tag{3}$$

where d_{ij} is the distance between nodes i and j, and α is the path loss index. Denote $\mathcal{I}_i$ as the set of nodes that can make interference at node i and η as the ambient Gaussian noise density. Then, the achievable rate from node i to node j within time slot k is

$$c_{ij}^k = t_k W \log_2 \left(1 + \frac{g_{ij} p_{ij}^k}{\eta W + \sum_{m \in \mathcal{I}_j, q \in \mathcal{T}_m}^{(m,q) \neq (i,j)} g_{mj} p_{mq}^k} \right).$$

2.3 Routing

The routing problem at the network layer considers, for a session l, $1 \leq l \leq L$, how to relay a rate of $r(l)$ from the source node $s(l)$ to the destination node $d(l)$. To take advantage of the multipath availability (that is, network diversity) within an ad hoc network, we allow a node to split its incoming data into subflows and send to different next-hop nodes. Denote $f_{ij}(l)$ as the data rate that is attributed to the lth session on link (i, j). If node i is the source node $s(l)$, then

$$\sum_{j \in \mathcal{T}_i} f_{ij}(l) = r(l) \quad (1 \leq l \leq L). \tag{4}$$

If node i is an intermediate relay node, that is, $i \neq s(l)$ and $i \neq d(l)$, then we have the following flow balance:

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(l)} f_{ij}(l) - \sum_{m \in \mathcal{T}_i}^{m \neq d(l)} f_{mi}(l) = 0 \quad (1 \leq l \leq L). \tag{5}$$

If node i is the destination node $d(l)$, then

$$\sum_{m \in \mathcal{T}_i} f_{mi}(l) = r(l) \quad (1 \leq l \leq L). \tag{6}$$

It can be easily verified that if (4) and (5) are satisfied, (6) must be satisfied. As a result, there is no need to list (6) in the formulation once we have both (4) and (5).

Since the sum of data rates on link (i, j) cannot be greater than the link capacity, we must also have

$$\sum_{1 \leq l \leq L}^{s(l) \neq j, d(l) \neq i} f_{ij}(l) \leq \sum_{k=1}^{K} c_{ij}^k.$$

Table 1 lists all notation used in this paper.

3 PROBLEM FORMULATION

3.1 The Maximum Utility Problem

In this section, we formulate the maximizing data rate utility problem as an optimization problem. Since a node cannot send and receive within the same time slot, we must have that if $p_{ij}^k > 0$ for any $j \in \mathcal{T}_i$, then $p_{mi}^k = 0$ for all $m \in \mathcal{T}_i$; or if $p_{mi}^k > 0$ for any $m \in \mathcal{T}_i$, then $p_{ij}^k = 0$ for all $j \in \mathcal{T}_i$. Mathematically, this property can be formulated by using binary variables x_i^k ($1 \leq i \leq N$ and $1 \leq k \leq K$), which we define as follows:

$$x_i^k = \begin{cases} 1 & \text{time slot } k \text{ is used for receiving data at node } i, \\ 0 & \text{otherwise.} \end{cases}$$

TABLE 1
Notation

Symbol	Definition
c_{ij}^k	Achievable rate from node i to node j in time slot k under p_{ij}^k
$\mathbf{c}$	The vector of c_{ij}^k, $1 \le i \le N$, $j \in \mathcal{T}_i$, $1 \le k \le K$
$d(l)$	Destination node of l-th session
d_{ij}	Distance between nodes i and j
$f_{ij}(l)$	Rate from node i to node j for flow l
$\mathbf{f}$	The vector of $f_{ij}(l)$, $1 \le i \le N$, $j \in \mathcal{T}_i$, $1 \le l \le L$
g_{ij}	Propagation gain from node i to node j
g_{ii}	Self-interference parameter at node i
g_{nom}	Propagation gain at a nominal distance
$\mathcal{I}_i$	Set of nodes that can produce interference on node i
K	Total number of time slots for scheduling
L	Total number of source-destination node pairs
LB_z	A lower bound of problem z
LB	Maximum lower bound among all problems
N	Total number of nodes in the network
p_{ij}^k	Power spent by node i in time slot k for sending data to node j
$\mathbf{p}$	The vector of p_{ij}^k, $1 \le i \le N$, $j \in \mathcal{T}_i$, $1 \le k \le K$
P_{max}	$= Q_{\text{max}} W / g_{\text{nom}}$ is the power limit.
Q_{max}	Maximum allowed transmission power spectral density
$r(l)$	Rate from $s(l)$ to $d(l)$
$\mathbf{r}$	The vector of $r(l)$, $1 \le l \le L$
$s(l)$	Source node of l-th session
t_k	Normalized length of time slot k, $\sum_{k=1}^K t_k = 1$.
$\mathbf{t}$	The vector of t_k, $1 \le k \le K$
$\mathcal{T}_i$	The set of one-hop neighbors of node i
UB_z	An upper bound of problem z
UB	Maximum upper bound among all problems
W	$= 7.5$ GHz is the entire spectrum for UWB networks.
Y_i^k	Total power (signal and noise) received by node i in time slot k
$\mathbf{Y}$	The vector for Y_i^k, $1 \le i \le N$, $1 \le k \le K$
α	Path loss index
η	Ambient Gaussian noise density
ε	Desired small error bound in the final solution
ψ_ε	A $(1-\varepsilon)$ optimal solution

Since $\sum_{j\in\mathcal{T}_i} p_{ij}^k \le P_{\text{max}}$ and $\sum_{m\in\mathcal{T}_i} p_{mi}^k \le |\mathcal{T}_i| P_{\text{max}}$, we have the following constraints for each node i, which capture both the constraint that a node i cannot send and receive within the same time slot k and the constraint on the power level:

$$\sum_{j\in\mathcal{T}_i} p_{ij}^k \le P_{\text{max}}(1 - x_i^k),$$

$$\sum_{m\in\mathcal{T}_i} p_{mi}^k \le |\mathcal{T}_i| P_{\text{max}} x_i^k.$$

The maximum utility problem (MUP) can now be formulated as follows:

MUP

$$\text{Max} \quad \sum_{l=1}^L \ln r(l)$$

$$\text{s.t.} \quad \sum_{k=1}^K t_k = 1,$$

$$c_{ij}^k = t_k W \log_2 \left(1 + \frac{g_{ij} p_{ij}^k}{\eta W + \sum_{m\in\mathcal{I}_m, q\in\mathcal{T}_m}^{(m,t)\neq(i,j)} g_{mj} p_{mq}^k} \right)$$
$$(1 \le i \le N, j \in \mathcal{T}_i, 1 \le k \le K), \quad (7)$$

$$\sum_{j\in\mathcal{T}_i} p_{ij}^k \le P_{\text{max}}(1 - x_i^k) \quad (1 \le i \le N, 1 \le k \le K), \quad (8)$$

$$\sum_{m\in\mathcal{T}_i} p_{mi}^k \le |\mathcal{T}_i| P_{\text{max}} x_i^k \quad (1 \le i \le N, 1 \le k \le K), \quad (9)$$

$$\sum_{k=1}^K c_{ij}^k - \sum_{1\le l\le L}^{s(l)\neq j, d(l)\neq i} f_{ij}(l) \ge 0 \quad (1 \le i \le N, j \in \mathcal{T}_i),$$

$$\sum_{j\in\mathcal{T}_i} f_{ij}(l) - r(l) = 0 \quad (1 \le l \le L, i = s(l)),$$

$$\sum_{j\in\mathcal{T}_i}^{j\neq s(l)} f_{ij}(l) - \sum_{m\in\mathcal{T}_i}^{m\neq d(l)} f_{mi}(l) = 0 \quad (1 \le l \le L, 1 \le i \le N,$$
$$i \neq s(l), d(l)),$$

$$r(l), f_{ij}(l) \ge 0 \quad (1 \le l \le L, 1 \le i \le N,$$
$$i \neq d(l), j \in \mathcal{T}_i, j \neq s(l)),$$

$$x_i^k = 0 \text{ or } 1, t_k, p_{ij}^k, c_{ij}^k \ge 0 \quad (1 \le i \le N, j \in \mathcal{T}_i, 1 \le k \le K).$$

The formulation for MUP is in the form of a *mixed-integer nonpolynomial programming* problem. Since even a special case of this problem such as a mixed-integer program or a nonpolynomial program is NP-hard in general [6], problem MUP is likely to be NP-hard, although we do not give a proof in this paper. Our approach to this problem is described as follows: As the first step, we show how to remove the integer (binary) variables and the nonpolynomial (log) terms in the constraints and reformulate MUP as an NLP problem. Since an NLP problem remains NP-hard in general, in Section 4, we present a solution procedure based on *branch-and-bound* and apply a novel RLT.

3.2 Problem Reformulation

We now show how to remove the integer constraints in (8) and (9) and the nonpolynomial (log) constraint in (7). The purpose of the integer (binary) variables x_i^k is to capture the fact that a node cannot send and receive within the same time slot; that is, if a node i sends data to any node j in a time slot k, then the data rate that can be received by node i within this time slot must be zero. Instead of using integer (binary) variables, we could use the following approach to achieve the same purpose.

We introduce the notion of *self-interference parameter* g_{jj} [11] with the following property:

$$g_{jj} \gg \max\{g_{ij} : i \in \mathcal{T}_j\}.$$

We incorporate this into the bit rate calculation in (7), that is

$$c_{ij}^k = t_k W$$
$$\cdot \log_2 \left(1 + \frac{g_{ij} p_{ij}^k}{\eta W + \sum_{m\in\mathcal{I}_j, q\in\mathcal{T}_m}^{(m,q)\neq(i,j)} g_{mj} p_{mq}^k + \sum_{q\in\mathcal{T}_j} g_{jj} p_{jq}^k} \right).$$
$$(10)$$

Thus, when any $p_{jq}^k > 0$, that is, node j is transmitting to a node q, we have $c_{ij}^k \approx 0$ even if $p_{ij}^k > 0$. In other words, when

node j is transmitting to any node q, the link capacity on link (i,j) is *effectively* shut down to zero. With this new notion of g_{ii}, we can capture the same transmission/ receiving behavior of a node without the need for integer (binary) variables x_i^k as in the MUP formulation. As a result, we can remove constraints (8) and (9) and just use the power limit constraint in (1).

To write (10) in a more compact form, we redefine $\mathcal{I}_i$ to include node i. Thus, (10) is now in the same form as in (7). Denote

$$Y_i^k = \sum_{m \in \mathcal{I}_i, q \in \mathcal{T}_m} g_{mi} \cdot p_{mq}^k. \tag{11}$$

We have

$$
\begin{aligned}
c_{ij}^k &= t_k W \log_2 \left(1 + \frac{g_{ij}p_{ij}^k}{\eta W + \sum_{m \in \mathcal{I}_j, q \in \mathcal{T}_m}^{(m,t) \neq (i,j)} g_{mj}p_{mq}^k} \right) \\
&= t_k W \log_2 \left(1 + \frac{g_{ij}p_{ij}^k}{\eta W + Y_j^k - g_{ij}p_{ij}^k} \right).
\end{aligned}
$$

To remove the nonpolynomial term in (7), we use the linear rate-SINR property that is unique to UWB. That is, we have a linear approximation for the $\log$ function, that is, $\ln(1 + x) \approx x$ when $0 < x \ll 1$. We have $c_{ij}^k \approx \frac{t_k W}{\ln 2} \frac{g_{ij}p_{ij}^k}{\eta W + Y_j^k - g_{ij}p_{ij}^k}$, which is equivalent to

$$\eta W c_{ij}^k + Y_j^k c_{ij}^k - g_{ij}p_{ij}^k c_{ij}^k - \frac{W g_{ij}}{\ln 2} t_k p_{ij}^k = 0.$$

Finally, without loss of generality, we assume the following ordering among t_k's:

$$t_1 \le t_2 \le \cdots \le t_k.$$

With these reformulation efforts, we now have a revised MUP (or R-MUP) formulation as follows:

R-MUP

$$\text{Max} \qquad \sum_{l=1}^{L} \ln r(l), \tag{12}$$

$$
\text{s.t.} \qquad \sum_{k=1}^{K} t_k = 1
$$
$$
t_k - t_{k-1} \ge 0 \quad (2 \le k \le K), \tag{13}
$$
$$
\sum_{j \in \mathcal{T}_i} p_{ij}^k \le P_{\max} \quad (1 \le i \le N, 1 \le k \le K),
$$

$$\sum_{m \in \mathcal{I}_i, q \in \mathcal{T}_m} g_{mi}p_{mq}^k - Y_i^k = 0 \quad (1 \le i \le N, 1 \le k \le K),$$

$$
\eta W c_{ij}^k + Y_j^k c_{ij}^k - g_{ij}p_{ij}^k c_{ij}^k - \frac{W g_{ij}}{\ln 2} t_k p_{ij}^k = 0
$$
$$
(1 \le i \le N, j \in \mathcal{T}_i, 1 \le k \le K), \tag{14}
$$

$$\sum_{k=1}^{K} c_{ij}^k - \sum_{1 \le l \le L}^{s(l) \neq j, d(l) \neq i} f_{ij}(l) \ge 0 \quad (1 \le i \le N, j \in \mathcal{T}_i),$$

$$\sum_{j \in \mathcal{T}_i} f_{ij}(l) - r(l) = 0 \quad (1 \le l \le L, i = s(l)),$$

$$
\sum_{j \in \mathcal{T}_i}^{j \neq s(l)} f_{ij}(l) - \sum_{m \in \mathcal{T}_i}^{m \neq d(l)} f_{mi}(l) = 0 \quad (1 \le l \le L, 1 \le i \le N,
$$
$$
i \neq s(l), d(l)),
$$

$$
r(l), f_{ij}(l) \ge 0 \quad (1 \le l \le L, 1 \le i \le N,
$$
$$
i \neq d(l), j \in \mathcal{T}_i, j \neq s(l)),
$$

$$t_k, p_{ij}^k, c_{ij}^k, Y_i^k \ge 0 \quad (1 \le i \le N, j \in \mathcal{T}_i, 1 \le k \le K).$$

Although the R-MUP formulation is simpler than the MUP formulation, it is still an NLP problem, which is NP-hard in general [6]. In the Section 4, we develop a solution procedure based on branch-and-bound [12] and the novel RLT [19] to solve this NLP problem.

4   A SOLUTION PROCEDURE FOR REVISED MAXIMUM UTILITY PROGRAM

4.1   Overview of the Solution Procedure

4.1.1   Duality Approach Is Not Applicable

The R-MUP formulation is in the form of an NLP problem. For certain NLP problems, it is possible to find a solution via its Lagrange dual problem. Specifically, if the objective function and constraint functions in the primal problem satisfy suitable *convexity* requirements, then the primal and dual problem have the same optimal objective value [3]. Unfortunately, such a duality-based approach, although attractive, is not applicable to our problem. This is because R-MUP is a *nonconvex* optimization problem (see (14)). There is likely a *duality gap* between the objective values of the optimal primal and dual solutions. As a result, a solution approach for the Lagrange dual problem cannot be used to solve our problem. Although there are efforts on solving nonconvex optimization problems via a duality (see, for example, [18] by Rubinov and Yang, where Lagrange-type dual problems are formulated with zero duality gap), we find that the complexity of such an approach is prohibitively high (much higher than the branch-and-bound solution approach proposed in this paper).

4.1.2   Branch-and-Bound Framework

Branch-and-bound is a generic framework to solve non-convex optimization problems [12]. Under this approach, we aim to find a $(1 - \varepsilon)$ optimal solution, where ε is a small positive constant reflecting our desired accuracy in the final solution. Initially, a branch-and-bound approach analyzes all variables in nonlinear terms (denote these variables as partition variables) and determines the value intervals for these variables. By using some *relaxation technique*, branch-and-bound obtains an LP relaxation for the original NLP problem; its solution provides an upper bound (UB) to the objective function. With the relaxation solution as a starting point, branch-and-bound uses a *local search* algorithm to find a feasible solution to the original NLP problem, which provides a lower bound (LB) for the objective function (see Fig. 1a for an example). If the obtained lower and upper
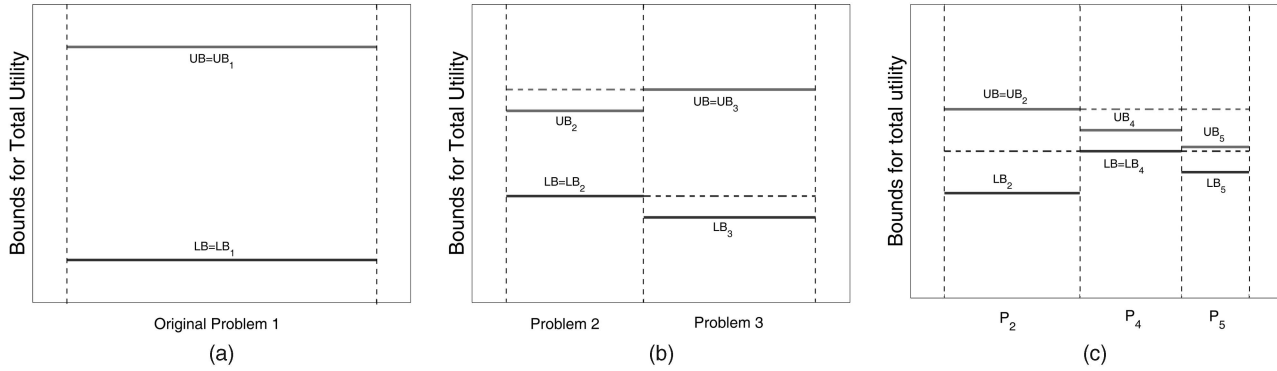
Fig. 1. Illustration of the branch-and-bound solution procedure. (a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

bounds are close to each other, that is, $LB \geq (1 - \varepsilon)UB$, we are done.

Define the relaxation error for a nonlinear term as the difference between the value of this term and the value of its corresponding new variable in the relaxation solution. If the relaxation errors for nonlinear terms are not small, then the lower bound LB could be far away from the upper bound UB. To close this gap, we must have a tighter LP relaxation, that is, with smaller relaxation errors. This could be achieved by further narrowing down the value intervals of partition variables. Specifically, branch-and-bound selects a partition variable and divides its value interval into two intervals by its value in the relaxation solution (see Fig. 1b). Then, the original problem (denoted as problem 1) is divided into two new problems (denoted as problem 2 and problem 3). Again, branch-and-bound performs relaxation and local search on these two new problems. Now, we have LB_2 and UB_2 for problem 2 and LB_3 and UB_3 for problem 3. Since the relaxations in problems 2 and 3 are both tighter than that in problem 1, we have $\max\{UB_2, UB_3\} \leq UB_1$ and $\max\{LB_2, LB_3\} \geq LB_1$. The upper bound of the original problem is updated from $UB = UB_1$ to $UB = \max\{UB_2, UB_3\}$, and the lower bound of the original problem is updated from $LB = LB_1$ to $LB = \max\{LB_2, LB_3\}$. As a result, we now have a smaller gap between UB and LB. If $LB \geq (1 - \varepsilon)UB$, we are done. Otherwise, we choose a problem with the maximum upper bound (Problem 3 in Fig. 1b) and perform partition for this problem.

Note that during the iteration process for branch-and-bound, if we find a problem z with $(1 - \varepsilon)UB_z \leq LB$, then we can conclude that this problem cannot provide much improvement on LB (see Problem 5 in Fig. 1c). That is, further branching on this problem will not yield much improvement, and we can thus remove this problem from further consideration. Eventually, once we find $LB \geq (1 - \varepsilon)UB$ or the problem list is empty, the branch-and-bound procedure terminates. It has been proved that under very general conditions, a branch-and-bound solution procedure always converges [19].

Fig. 2 shows the general framework of the branch-and-bound procedure. Note that the key components in the branch-and-bound procedure are problem specific and must be carefully designed to make it work. These include

1. how to estimate initial value intervals for partition variables,
2. how to obtain a linear relaxation,
3. how to perform local search, and
4. if the nonlinear term with the maximum relaxation error has multiple variables, which partition variable should we select for branch-and-bound.

In the rest of this section, we design each of these components.

4.2 Initial Value Intervals for Partition Variables

The linear relaxation would be tight if the value interval for each variable in a nonlinear term is small enough. Thus, we should have tight initial value intervals for these variables. For R-MUP, t_k, p_{ij}^k, c_{ij}^k, $r(l)$, and Y_i^k are the variables that are in nonlinear terms whose value intervals are candidates to be partitioned.

Main Algorithm

1.   Initialization:
2.   Let the initial best solution $\psi_\varepsilon = \emptyset$ and the initial lower bound $LB = -\infty$.
3.   Determine partition variables (variables in a nonlinear term) and estimate their initial value intervals.
4.   Let the initial problem list include only the original problem, denoted as problem 1.
5.   Introduce one new variable for each nonlinear term and add linear constraints for these variables to build a linear relaxation.
6.   Denote the solution to the linear relaxation as $\hat{\psi}_1$ and its objective value as the upper bound UB_1.
7.   Let the initial upper bound $UB = UB_1$.
8.   Iteration:
9.   Select problem z that has the maximum UB_z among all problems in the problem list.
10.   Find a feasible solution ψ_z from its $\hat{\psi}_z$ via a local search algorithm and denote its objective value as LB_z.
11.   If $(LB_z > LB)$ {
12.   Update $\psi_\varepsilon = \psi_z$ and $LB = LB_z$.
13.   If $LB \geq (1 - \varepsilon)UB$, we stop with $(1 - \varepsilon)$ optimal solution ψ_ε.
14.   Otherwise, remove all problems z' with $(1 - \varepsilon)UB_{z'} \leq LB$ from the problem list. }
15.   Define relaxation error for a nonlinear term as the difference between the value of this term and the value of its corresponding new variable in $\hat{\psi}_z$.
16.   Select a partition variable with the maximum relaxation error and divide its value interval into two intervals by its value in $\hat{\psi}_z$.
17.   Define two new problems z_1 and z_2 based on these two intervals.
18.   Remove problem z from the problem list.
19.   Obtain upper bounds UB_{z1} and UB_{z2} for problems z_1 and z_2 via their linear relaxations.
20.   If $(1 - \varepsilon)UB_{z1} > LB$, add problem z_1 into the problem list.
21.   If $(1 - \varepsilon)UB_{z2} > LB$, add problem z_2 into the problem list.
22.   If the problem list is empty, we stop with $(1 - \varepsilon)$ optimal solution ψ_ε.
23.   Otherwise, go to the next iteration.

Fig. 2. Main algorithm of the branch-and-bound solution procedure.

Since $t_1 \leq t_2 \leq \cdots \leq t_K$ and $\sum_{k=1}^{K} t_k = 1$, we have $0 \leq t_k \leq \frac{1}{K+1-k}$. For p_{ij}^k, we have $0 \leq p_{ij}^k \leq P_{\max}$. For c_{ij}^k, its upper bound can be calculated by having node i transmit to node j with $P_{\max}$ while silencing off all the other transmissions that may produce interference at node j and $t_k = 1$. We have $0 \leq c_{ij}^k \leq W \log_2(1 + \frac{g_{ij}P_{\max}}{\eta W})$. The upper bound of Y_i^k is achieved when all nodes in $\mathcal{I}_i$ transmit with $P_{\max}$. Thus, we have $0 \leq Y_i^k \leq P_{\max} \sum_{m \in \mathcal{I}_i} g_{mi}$.

We now develop an upper bound for $r(l)$. Recall that $r(l)$ is the aggregate rate that can be potentially split and routed among multiple paths between source node $s(l)$ and destination node $d(l)$. Since $r(l)$ should be no more than the maximum transmission rate from source node $s(l)$ and the maximum receiving rate to destination node $d(l)$, we analyze these two bounding rates individually. At source node $s(l)$, its transmission upper bound $C_{s(l)}$ can be calculated by having node $s(l)$ transmit to its nearest neighbor with $P_{\max}$ on all time slots. Assuming that the nearest neighbor of $s(l)$ is j, we have $C_{s(l)} = W \log_2\left(1 + \frac{g_{s(l),j}P_{\max}}{\eta W}\right)$. At destination node $d(l)$, it turns out that an upper bound for receiving rate $C_{d(l)}$ is achieved when each node $m \in \mathcal{T}_{d(l)}$ transmits to $d(l)$ with $P_{\max}$ in all time slots. This result is stated in the following lemma. A formal proof is given in the Appendix.

Lemma 1. *Denote $C_{d(l)}$ as an upper bound of the maximum rate that node $d(l)$ can receive. We then have*

$$C_{d(l)} = \sum_{m \in \mathcal{T}_{d(l)}} W \log_2 \left(1 + \frac{g_{m,d(l)}P_{\max}}{\eta W + P_{\max}\sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)}} \right).$$

Based on the rate analysis at source node $s(l)$ and destination node $d(l)$ for the lth session, we have

$$0 \leq r(l) \leq \min\{C_{s(l)}, C_{d(l)}\}.$$

Denote $\mathbf{t}$, $\mathbf{p}$, $\mathbf{c}$, $\mathbf{r}$, and $\mathbf{Y}$ as the vectors for variables t_k, p_{ij}^k, c_{ij}^k, $r(l)$, and Y_i^k, respectively. Also denote Ω as the set of all possible value of $\mathbf{t}$, $\mathbf{p}$, $\mathbf{c}$, $\mathbf{r}$, and $\mathbf{Y}$. For the original problem (denoted as problem 1), we have $\Omega_1 = \{(\mathbf{t},\mathbf{p},\mathbf{c},\mathbf{r},\mathbf{Y}) : 0 \leq t_k \leq \frac{1}{K+1-k}, 0 \leq p_{ij}^k \leq P_{\max}, 0 \leq c_{ij}^k \leq W \log_2(1 + \frac{g_{ij}P_{\max}}{\eta W}), 0 \leq r(l) \leq \min\{C_{s(l)}, C_{d(l)}\}, 0 \leq Y_i^k \leq P_{\max}\sum_{m \in \mathcal{I}_i} g_{mi}\}$.

4.3 Linear Relaxation

During each iteration of the branch-and-bound procedure, we need a linear relaxation to obtain an upper bound of the objective function (see line 5 in Fig. 2). For the polynomial term, we propose to employ RLT [19]. For the nonpolynomial term (that is, log term), we propose to employ three tangential supports, which constitutes a convex envelope linear relaxation. We elaborate each technique below.

4.3.1 Reformulation-Linearization Technique

We first show how RLT can obtain a linear relaxation for a polynomial term. Specifically, $Y_j^k c_{ij}^k$, $p_{ij}^k c_{ij}^k$, and $t_k p_{ij}^k$ in (14) are polynomial terms. RLT enables us to use new variables to replace those polynomial terms and add linear constraints for these new variables, thus relaxing a nonlinear
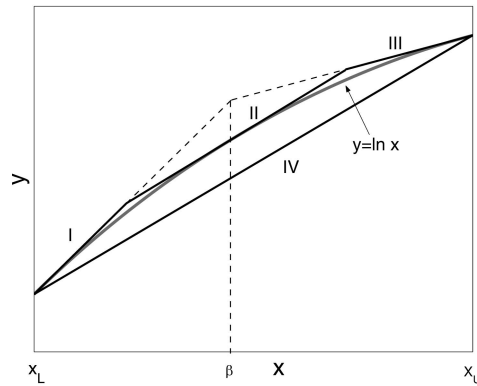
Fig. 3. A convex envelope for $y = \ln x$ with three tangential supports (I, II, and III) and a chord (IV).

constraint into linear constraints. These RLT constraints are derived from the value intervals of the original variables.

As an example, we introduce a new variable u_{ij}^k for $Y_j^k c_{ij}^k$, that is, $u_{ij}^k := Y_j^k c_{ij}^k$. Assume that $(Y_j^k)_L \leq Y_j^k \leq (Y_j^k)_U$ and $(c_{ij}^k)_L \leq c_{ij}^k \leq (c_{ij}^k)_U$. Then, we have $[Y_j^k - (Y_j^k)_L] \cdot [c_{ij}^k - (c_{ij}^k)_L] \geq 0$, $[Y_j^k - (Y_j^k)_L] \cdot [(c_{ij}^k)_U - c_{ij}^k] \geq 0$, $[(Y_j^k)_U - Y_j^k] \cdot [c_{ij}^k - (c_{ij}^k)_L] \geq 0$, and $[(Y_j^k)_U - Y_j^k] \cdot [(c_{ij}^k)_U - c_{ij}^k] \geq 0$. From the above relationships, we obtain the following *linear* constraints (also called RLT constraints [19]) for u_{ij}^k:

$$\left(Y_j^k\right)_L \cdot c_{ij}^k + \left(c_{ij}^k\right)_L \cdot Y_j^k - u_{ij}^k \leq \left(Y_j^k\right)_L \cdot \left(c_{ij}^k\right)_L,$$
$$\left(Y_j^k\right)_U \cdot c_{ij}^k + \left(c_{ij}^k\right)_L \cdot Y_j^k - u_{ij}^k \geq \left(Y_j^k\right)_U \cdot \left(c_{ij}^k\right)_L,$$
$$\left(Y_j^k\right)_L \cdot c_{ij}^k + \left(c_{ij}^k\right)_U \cdot Y_j^k - u_{ij}^k \geq \left(Y_j^k\right)_L \cdot \left(c_{ij}^k\right)_U,$$
$$\left(Y_j^k\right)_U \cdot c_{ij}^k + \left(c_{ij}^k\right)_U \cdot Y_j^k - u_{ij}^k \leq \left(Y_j^k\right)_U \cdot \left(c_{ij}^k\right)_U.$$

Through this relaxation, we can replace $Y_j^k c_{ij}^k$ with u_{ij}^k in (14) and add RLT constraints for u_{ij}^k into the R-MUP formulation.

Following the same token, we can let $v_{ij}^k = p_{ij}^k c_{ij}^k$ and $w_{ij}^k = t_k p_{ij}^k$. From $(p_{ij}^k)_L \leq p_{ij}^k \leq (p_{ij}^k)_U$, $(c_{ij}^k)_L \leq c_{ij}^k \leq (c_{ij}^k)_U$, and $(t_k)_L \leq t_k \leq (t_k)_U$, we can obtain the RLT constraints for v_{ij}^k and w_{ij}^k as well. As a result, in the relaxed problem, we are able to have all constraints in linear form.

4.3.2 Linear Relaxation for the Log Function

Now, we show how to obtain a linear relaxation for a nonpolynomial term. We denote $h(l) = \ln r(l)$ for $\ln r(l)$. Note that the function $y = \ln x$, over suitable bounds of x, can be bounded by four segments (or a convex envelope), where segments I, II, and III are tangential supports, and segment IV is the chord (see Fig. 3). In particular, three tangent segments are at $(x_L, \ln x_L)$, $(\beta, \ln \beta)$, and $(x_U, \ln x_U)$, where $\beta = \frac{x_L \cdot x_U \cdot (\ln x_U - \ln x_L)}{x_U - x_L}$ is the horizontal location for the point that intersects extended tangent segments I and III; segment IV is the segment that joins points $(x_L, \ln x_L)$ and $(x_U, \ln x_U)$. The convex region defined by the four segments can be described by the following four *linear* constraints:

$$x_L \cdot y - x \leq x_L(\ln x_L - 1),$$
$$\beta \cdot y - x \leq \beta(\ln \beta - 1),$$
$$x_U \cdot y - x \leq x_U(\ln x_U - 1),$$
$$(x_U - x_L)y + (\ln x_L - \ln x_U)x \geq x_U \cdot \ln x_L - x_L \cdot \ln x_U.$$

As a result, the nonpolynomial (log) term can also be relaxed into a linear objective and constraints.

After we replace all nonlinear terms with new variables and add the corresponding linear constraints into the R-MUP formulation, we have the following LP relaxation for R-MUP formulation:

$$\text{Max} \quad \sum_{l=1}^{L} h(l)$$

$$\text{s.t.} \quad \sum_{k=1}^{K} t_k = 1,$$

$$t_k - t_{k-1} \geq 0 \quad (2 \leq k \leq K),$$

$$\sum_{j \in \mathcal{T}_i} p_{ij}^k \leq P_{\max} \quad (1 \leq i \leq N, 1 \leq k \leq K),$$

$$\sum_{m \in \mathcal{I}_i, q \in \mathcal{T}_m} g_{mi} p_{mq}^k - Y_i^k = 0 \quad (1 \leq i \leq N, 1 \leq k \leq K),$$

$$\eta W c_{ij}^k + Y_j^k c_{ij}^k - g_{ij} p_{ij}^k c_{ij}^k - \frac{W g_{ij}}{\ln 2} t_k p_{ij}^k = 0$$
$$(1 \leq i \leq N, j \in \mathcal{T}_i, 1 \leq k \leq K),$$

$$\text{RLT constraints for } u_{ij}^k, v_{ij}^k, \text{ and } w_{ij}^k$$
$$(1 \leq i \leq N, j \in \mathcal{T}_i, 1 \leq k \leq K),$$

$$\sum_{k=1}^{K} c_{ij}^k - \sum_{1 \leq l \leq L}^{s(l) \neq j, d(l) \neq i} f_{ij}(l) \geq 0 \quad (1 \leq i \leq N, j \in \mathcal{T}_i),$$

$$\sum_{j \in \mathcal{T}_i} f_{ij}(l) - r(l) = 0 \quad (1 \leq l \leq L, i = s(l)),$$

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(l)} f_{ij}(l) - \sum_{m \in \mathcal{T}_i}^{m \neq d(l)} f_{mi}(l) = 0 \quad (1 \leq l \leq L, 1 \leq i \leq N,$$
$$i \neq s(l), d(l)),$$

Three tangential supports' constraints for $(r(l), h(l))$
$$(1 \leq l \leq L),$$

$$h(l), f_{ij}(l) \geq 0 \quad (1 \leq l \leq L, 1 \leq i \leq N, i \neq d(l),$$
$$j \in \mathcal{T}_i, j \neq s(l)),$$

$$u_{ij}^k, v_{ij}^k, w_{ij}^k \geq 0 \quad (1 \leq i \leq N, j \in \mathcal{T}_i, 1 \leq k \leq K),$$

$$(\mathbf{t}, \mathbf{p}, \mathbf{c}, \mathbf{r}, \mathbf{Y}) \in \Omega.$$

4.4 Local Search Algorithm

An LP relaxation for a problem z can be solved in polynomial time. Denote the relaxation solution as $\hat{\psi}_z$, which provides an upper bound to problem z but may not be feasible. We now show how to obtain a feasible solution ψ_z based on $\hat{\psi}_z$ (see line 10 in Fig. 2).

Now, we discuss the main idea on how to determine a feasible solution ψ_z based on the relaxation solution $\hat{\psi}_z$. Since $\hat{\mathbf{t}}$ in $\hat{\psi}_z$ is feasible, we can set $\mathbf{t} = \hat{\mathbf{t}}$ in our local search algorithm. Note that in R-MUP, we introduced the notion of a self-interference parameter to remove the binary variables in MUP. Then, in $\hat{\mathbf{p}}$, it is possible that $\hat{p}_{ij}^k > 0$ and $\hat{p}_{mi}^k > 0$ for certain node i within a time slot k, which is infeasible to our

---

A Local Search Algorithm for p

1. If $(p_{ij}^k)_L > 0$, then time slot k must be used on link (i, j) and we should set $\hat{p}_{mi}^k = 0$ (node i cannot use time slot k for receiving) for all m and $\hat{p}_{jq}^k = 0$ (node j cannot use time slot k for transmission) for all q.
2. Choose a node i that meets the following requirement. If no such node exists, then let $\mathbf{p} = \hat{\mathbf{p}}$ and we are done.
3.   First, identify a node such that all nodes that transmit data to this node already have their power updated.
4.   If no such node exists, choose any node that has not yet updated its power.
5. If there is no conflicting time slot, i.e., a time slot used for both transmission and receiving, at node i under $\hat{\mathbf{p}}$, then go to line 2.
6. Otherwise, define T_t as the total normalized length of time slots used by node i for transmission and T_{r-t} as the total normalized length of time slots used by node i only for receiving.
   //Node i tries to remove conflicting on time slots
7. If $(T_t \leq T_{r-t})$ {
   //Node i tries to use all conflicting time slots for transmission.
8.   For a conflicting time slot k, if for each node m with $\hat{p}_{mi}^k > 0$, time slot k is not the only time slot used on link (m, i), then set $\hat{p}_{mi}^k = 0$.
9.   Otherwise, //time slot k is the only time slot used on a link (m, i)
10.   if for each node j with $\hat{p}_{ij}^k > 0$, time slot k is not the only time slot used on link (i, j), then set $\hat{p}_{ij}^k = 0$ and update T_t and T_{r-t}; else, set $\hat{p}_{mi}^k = 0$ for each m. }
11. else { //Node i tries to release some conflicting time slots
12.   Node i chooses a conflicting time slot k in the non-decreasing order of $\sum_{j \in \mathcal{T}_i} \hat{p}_{ij}^k$.
13.   If for each node j with $\hat{p}_{ij}^k > 0$, time slot k is not the only time slot used on link (i, j), then set $\hat{p}_{ij}^k = 0$ and update T_t and T_{r-t}.
14.   Node i chooses a conflicting time slot k in the non-decreasing order of $\sum_{m \in \mathcal{T}_i} \hat{p}_{mi}^k$.
15.   If for each node m with $\hat{p}_{mi}^k > 0$, time slot k is not the only time slot used on link (m, i), then set $\hat{p}_{mi}^k = 0$.
16.   While $(T_t > (T_t + T_{r-t})/2)$, node i chooses a conflicting time slot k in the non-decreasing order of $\sum_{j \in \mathcal{T}_i} \hat{p}_{ij}^k$
17.   Set $\hat{p}_{ij}^k = 0$ and update T_t, T_{r-t}.
18.   For each remaining conflicting time slot k, set $\hat{p}_{mi}^k = 0$ for each m. }
19. Go to line 2.

Fig. 4. A local search algorithm to obtain $\mathbf{p}$.

original problem MUP. Therefore, it is necessary to find a feasible $\mathbf{p}$ from $\hat{\mathbf{p}}$ by changing some $\hat{p}_{ij}^k$ or $\hat{p}_{mi}^k$ to zero such that no node can send and receive within the same time slot. We apply the following two guidelines when we set some transmission powers to zero. First, we try to maintain the same connectivity in ψ_z as that in $\hat{\psi}_z$ wherever possible. That is, if a time slot k is the only time slot used on link (i, j), then we should avoid making $p_{ij}^k = 0$ in ψ_z (see lines 8, 10, 13, and 15 in Fig. 4). Second, we try to split the total time slots used at a node i into two groups of equal length whenever possible: one group for transmission and the other group for receiving (see lines 7 and 16 in Fig. 4).

An algorithm that determines a feasible solution ψ_z is shown in Fig. 4. Initially, if in problem z, a transmission power lower bound $(p_{ij}^k)_L$ is not zero, then we have $p_{ij}^k > 0$; that is, time slot k must be used on link (i, j). As a result, it cannot be used on node i for receiving or on node j for transmission. We should set the corresponding transmission powers as zero. We then update the transmission power at one node in each iteration. In an iteration, we first select a node, say, i, and update the transmission power on this node to resolve the time-slot conflict. Whenever possible, we choose such a node that all nodes that transmit data to this node already have their power updated. We count and will update the total normalized length of used time slots on node i. Define T_t as the total normalized length of time slots used by node i for transmission (including conflicting time slots used for both transmission and receiving) and T_{r-t} as the total normalized length of time
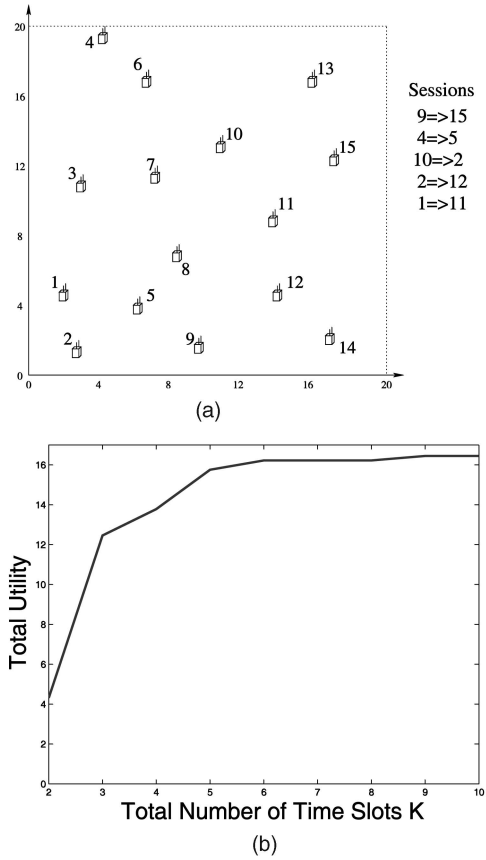
slots used by node i only for receiving. For each of these conflicting time slots, we need to determine its use for either transmission or receiving, and we aim to make $T_t \approx T_{r-t}$. If $T_t \leq T_{r-t}$, then we try to use all conflicting time slots for transmission. Otherwise, we try to use some conflicting time slots for transmission such that $T_t \approx T_{r-t}$. As we discussed in the previous paragraph, we try to maintain the same connectivity whenever possible. After we finish updating on node i, in the next iteration, we move on to another node until all nodes are updated. Thus, our algorithm terminates with a feasible **p**.

After we obtain **t** and **p** for ψ_z, we can compute c_{ij}^k from (7). The routing solution **f** in ψ_z can be obtained by solving the following concave optimization problem through a standard approach [3]:

$$\text{Max} \quad \sum_{l=1}^{L} \ln r(l)$$

$$\text{s.t.} \quad \sum_{1 \leq l \leq L}^{s(l) \neq j, d(l) \neq i} f_{ij}(l) \leq \sum_{k=1}^{K} c_{ij}^k \quad (1 \leq i \leq N, j \in \mathcal{T}_i),$$

$$\sum_{j \in \mathcal{T}_i} f_{ij}(l) - r(l) = 0 \quad (1 \leq l \leq L, i = s(l)),$$

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(l)} f_{ij}(l) - \sum_{m \in \mathcal{T}_i}^{m \neq d(l)} f_{mi}(l) = 0 \quad \left(1 \leq l \leq L, 1 \leq i \leq N,\right.$$
$$\left. i \neq s(l), d(l)\right),$$

$$r(l), f_{ij}(l) \geq 0 \quad \left(1 \leq l \leq L, 1 \leq i \leq N,\right.$$
$$\left. i \neq d(l), j \in \mathcal{T}_i, j \neq s(l)\right).$$

4.5 Further Enhancements

4.5.1 Selection of Partition Variables

Note that in line 16 of the branch-and-bound procedure (see Fig. 2), we choose a partition variable in the nonlinear term with the maximum relaxation error. If this nonlinear term is $\ln r(l)$, we will partition on $r(l)$. If this nonlinear term has multiple variables, for example, $Y_j^k c_{ij}^k$, then we need to choose a partition variable from Y_j^k and c_{ij}^k. Specifically, if $((Y_j^k)_U - (Y_j^k)_L) \cdot \min\{\hat{Y}_j^k - (Y_j^k)_L, (Y_j^k)_U - \hat{Y}_j^k\} \geq ((c_{ij}^k)_U - (c_{ij}^k)_L) \cdot \min\{\hat{c}_{ij}^k - (c_{ij}^k)_L, (c_{ij}^k)_U - \hat{c}_{ij}^k\}$ (where $\hat{Y}_j^k$ and $\hat{c}_{ij}^k$ are the values of Y_j^k and c_{ij}^k in the relaxation solution, $(Y_j^k)_L$ and $(Y_j^k)_U$ are the current lower and upper bounds of Y_j^k, and $(c_{ij}^k)_L$ and $(c_{ij}^k)_U$ are the current lower and upper bounds of c_{ij}^k), we partition on Y_j^k and obtain two new value intervals $[(Y_j^k)_L, \hat{Y}_j^k]$ and $[\hat{Y}_j^k, (Y_j^k)_U]$. Otherwise, we partition on c_{ij}^k. Similarly, we can perform partition if this nonlinear term is $p_{ij}^k c_{ij}^k$ or $t_k p_{ij}^k$.

For our specific problem, further improvement can be made to the partition variable selection policy by exploiting the physical interpretation of variables and weighing their significance. For example, it is clear that variable t_k directly affects the final solution. As a result, the algorithm will run much more efficiently if we give it a higher priority when we choose a partition variable. This is precisely what we have done in our algorithm implementation. That is, when we choose a partition variable, we give the highest priority to t_k, the second highest priority to p_{ij}^k, then c_{ij}^k, and finally

Y_i^k. Note that the choice of partition variable will not hamper the convergence property of the algorithm [19], although it will yield different computational times.

4.5.2 Elimination of Nonimproving Problems

There are two types of problems that can be eliminated before solving their LP relaxations. In the first case, if a problem is found to be infeasible, then there is no need to solve a full-scale LP relaxation. For example, after we partition on p_{ij}^k, if a node must send and receive within the same time slot in a new problem, that is, $(p_{ij}^k)_L > 0$ and $(p_{mi}^k)_L > 0$, then this new problem must be infeasible.

In the second case, if a problem cannot provide much improvement, then there is no need to solve a full-scale LP either. For example, after we partition on $r(l)$, if $(1 - \varepsilon) \sum_{l=1}^{L} \ln(r(l)_U) \leq LB$, then this new problem cannot provide much improvement and can be eliminated from problem list.

5 NUMERICAL EXAMPLES AND DISCUSSIONS

In this section, we apply the solution procedure to some example networks and offer further insights on the optimization problem. These numerical results are important as they are not obvious from our theoretical development of the solution procedure in Section 4.

We consider randomly generated networks of 15 nodes deployed over a 20×20 area. All distances are based on normalized length in (3). There are five sessions in the network. The path loss index is $\alpha = 2$, and the nominal gain is chosen as $g_{nom} = 0.02$. The power density limit Q_{max} is assumed to be 1 percent of the white noise η [16].

We will compare a cross-layer approach to decoupled approaches to our problem.

5.1.1 Scheduling—Number of Required Time Slots K

We first investigate how the total utility is affected when the total available time slots K change. One would expect that the total utility is a nondecreasing function of K. This is because the more time slots available, the more opportunities (or larger design space) for each node to avoid interference with other nodes, thus yielding more utility for the network. However, there is a price to pay for having a large K. The larger the K is, the larger the total length of all time slots in a cycle, and thus, the larger the delay at each node. Further, the larger the K, the larger the size of our optimization problem, which means more computational time will be incurred. Therefore, we wish to choose a minimum K that can provide near-optimal performance.

We consider two randomly generated networks (see Figs. 5a and 6a). The total utility under different K are shown in Figs. 5b and 6b, respectively. Since multihop is not allowed when $K = 1$ (a node cannot send and receive within the single time slot), the minimum required K is two. As expected, the total utility is a nondecreasing function of K. It is interesting to note that there is a "knee" point in each figure; that is, when K is larger than a certain number, there is hardly much increase in the total utility. This suggests that for practical purposes, it is sufficient to choose a certain K for a particular network.

Fig. 5. A 15-node ad hoc network with five sessions. (a) Network topology. (b) Total utility as a function of total time slots K.
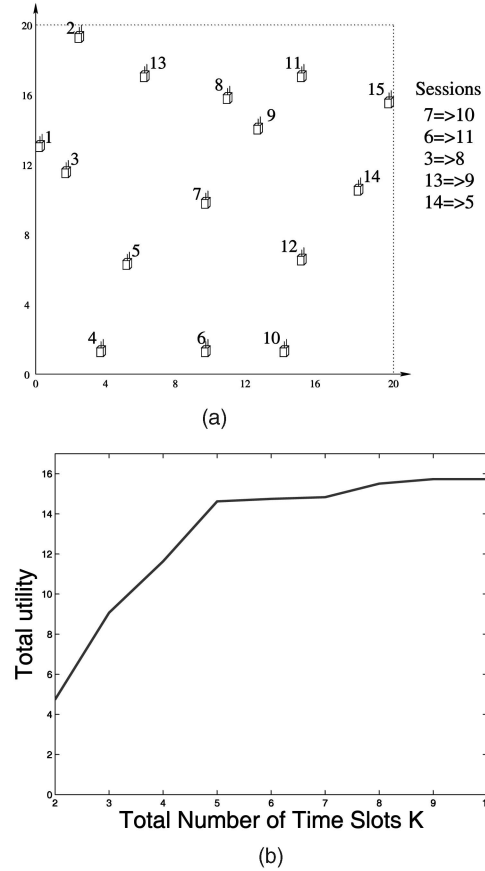

Fig. 6. A second instance of a 15-node ad hoc network with five sessions. (a) Network topology. (b) Total utility as a function of total time slots K.

In general, since 1) the total utility is a nondecreasing function of K and 2) the total utility cannot increase to infinity and must saturate at some value, there must exist a knee point for K. However, the specific value of the knee point depends on the specific network settings and parameters and may be different for different network instances. For example, for the network in Fig. 6a, the knee point for K is around eight, which is different from that for the network in Fig. 5a, which is six.

5.1.2 Scheduling—Length Allocation for Each Time Slot

We now show how the allocation of length for each time slot affects performance. To plot the complete routing topology (multipath for each session) legibly on a figure, we use the first three sessions in Fig. 5a for this investigation (the results for the network in Fig. 6a are similar). Such small L will enable us to plot the complete routing topology (multipath for each session) legibly on a figure. We choose $K = 6$ in this case based on our results in Fig. 5b.

Fig. 7 shows an optimal routing for these three sessions. Clearly, an optimal routing for each session is a multipath routing. Given this routing topology, Table 2 shows the results under different allocations for normalized time-slot length. The first policy is the optimal length allocation obtained via our solution, which apparently offers the largest total utility. The second time-slot allocation policy is equal allocation. Finally, we also list the performance of this policy under two random time-slot allocations. Note that equal allocation is not a good policy and has about the same performance as a random allocation policy.
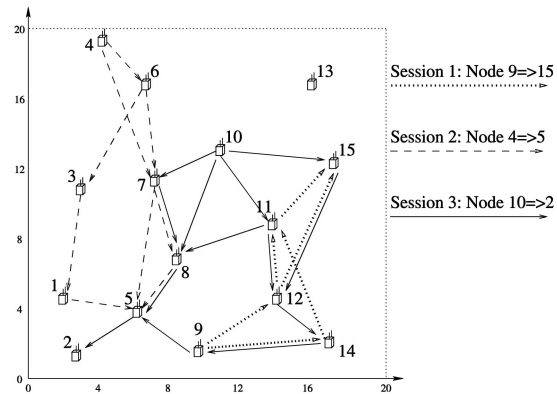
Fig. 7. Optimal routing for three sessions with $K = 6$.

5.1.3 Routing

For a given optimal time-slot length (obtained via our solution procedure), we now study the impact of routing on our cross-layer optimization problem. We again use the first three sessions ($L = 3$) in Fig. 5a and $K = 6$ time slots for this investigation. In addition to our cross-layer optimal routing, we also consider the following two routing approaches, namely, minimum-energy routing and minimum-hop routing. Under minimum-energy routing, the energy cost is defined as d_{ij}^{α} for link (i, j). Under minimum-hop routing, the cost is measured in the number of hops.

Table 3 shows the results of this study. Clearly, cross-layer optimal routing outperforms both minimum-energy

TABLE 2
Total Utility and the Corresponding Sum of Rates under
Different Time-Slot Length Allocation Policies
with $K = 6$ and $L = 3$

Allocation of Normalized Time Slot Length	$\sum_{l=1}^{L} \ln r(l)$	$\sum_{l=1}^{L} r(l)$
Optimal: (0.14, 0.14, 0.14, 0.15, 0.21, 0.22)	11.61	149.14
Equal: (0.16, 0.16, 0.17, 0.17, 0.17, 0.17)	9.51	78.42
Random: (0.11, 0.11, 0.13, 0.16, 0.20, 0.29)	9.15	69.04
Random: (0.14, 0.15, 0.16, 0.18, 0.18, 0.19)	9.63	84.14

TABLE 3
Total Utility and the Corresponding Sum of Rates under
Different Routing Strategies for $K = 6$ and $L = 3$

Routing Strategy	$\sum_{l=1}^{L} \ln r(l)$	$\sum_{l=1}^{L} r(l)$
Optimal Routing	11.61	149.14
Minimum-Energy Routing	9.39	117.25
Minimum-Hop Routing	8.63	53.99

and minimum-hop routing approaches. Both minimum-energy routing and minimum-hop routing are minimum-cost routing (with different link costs). Minimum-cost routing only uses a single path; that is, multipath routing is not allowed, which may not provide a good solution. Moreover, it is very likely that multiple sessions share the same path (see path $7 \rightarrow 8 \rightarrow 5$ in Fig. 8a). Thus, the rates for these sessions are bounded by the capacity of this path. On the other hand, minimum-hop routing has its own unique problem. Minimum-hop routing (see Fig. 8b) prefers a fewer number of hops (with a long distance on each hop) toward the destination node. Clearly, a long-distance hop will reduce its corresponding link's capacity, due to the distance gain factor.

6 RELATED WORK

Cross-layer optimization for wireless networks has been an active research area over the past few years. However, most of these investigations focused on narrow-band wireless networks (for example, [2], [5], and [20]), and these results usually cannot be carried over to UWB-based networks.

An overview of UWB technology is given in [13]. Physical-layer issues associated with UWB-based multiple-access communications can be found in [7], [8], [21], and references therein. In this section, we focus our attention on related work addressing networking-related problems for UWB-based networks.

In [10], Negi and Rajeswaran first showed that in contrast to previously published results, the throughput for UWB-based ad hoc networks increases with node density. This important result is mainly due to the large bandwidth and the ability of power and rate adaptation of UWB-based nodes, which alleviate interference. More importantly, this result demonstrates the significance of physical-layer properties on network-layer metrics such as network capacity. In [1], Baldi et al. considered the admission control problem based on a flexible cost function in UWB-based networks. Under their approach, a communication cost is attached to each path, and the cost of a path is the sum of costs associated with the links it comprises. An admissibility test is then made based on the cost of a path.
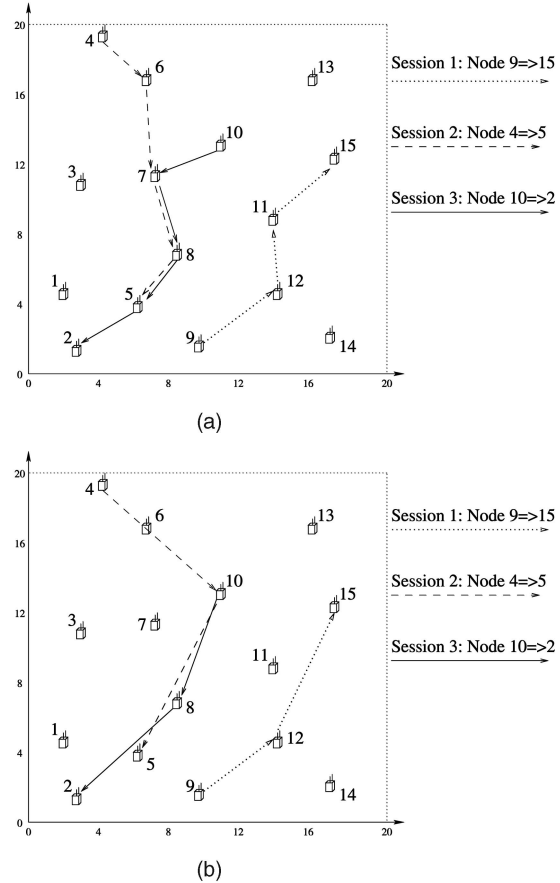

Fig. 8. Minimum-energy routing and minimum-hop routing for $L = 3$ in the 15-node network. (a) Minimum-energy routing. (b) Minimum-hop routing.

There is no explicit consideration of joint cross-layer optimization of routing, scheduling, and power control in this admissibility test. In [4], Cuomo et al. studied a multiple-access scheme for UWB. Power control and rate allocation problems were formulated for both elastic bandwidth data traffic and guaranteed-service traffic. The impact of routing was not considered.

The most relevant research for our work are [11] and [15]. In [11], Negi and Rajeswaran studied how to maximize proportional rate allocation in a single-hop UWB network (the source node transmits to the destination node directly). The problem was formulated as a cross-layer optimization problem with scheduling and power control constraints similar to the ones used in this paper. The difference is that in this paper, we consider a multihop network environment where routing is also part of the cross-layer optimization problem. As a result, the problem is considerably more difficult. The review of [15] was given in detail in Section 1.

7 CONCLUSION

In this paper, we studied the data rate utility problem in a UWB-based ad hoc network. We formulated the problem into an NLP with consideration of physical-layer, link-layer, and network-layer variables. We proposed a solution procedure based on the branch-and-bound framework and developed the key components under this

framework, including setting the value intervals for partition variables, linear relaxation, and local search algorithm. A powerful technique that we employed in the branch-and-bound framework is the RLT, which is able to replace nonlinear terms in the constraints with linear ones. We also used numerical results to validate the efficiency of the theoretical solution and offered additional insights on this problem.

The significance of this work is that it provides a theoretical result on the achievable performance bound for a UWB-based ad hoc network. For practical implementation, distributed algorithms and protocols still remain to be developed, which should be future work in this field. For such efforts, the theoretical results presented in this paper can be used as a performance benchmark or measure for any of these distributed algorithms and protocols (rather than comparing them to some heuristic algorithms).

APPENDIX

PROOF OF LEMMA 1

To obtain an upper bound of the received data rate at node $d(l)$, we focus on the last hop transmission. Since we are only considering the last hop transmission, the search space now only consists of scheduling and power control. For any given scheduling policy (the number of time slots K and normalized length of each time slot), we show that the achievable bit rate increases with the transmission power of each node at any time slot. This finding suggests that each node should use the maximum transmission power in all time slots. Subsequently, we compute an upper bound of the maximum receiving rate at $d(l)$.

Denote $f_m(l)$ as the data rate from node $m \in \mathcal{T}_{d(l)}$ to node $d(l)$ and

$$(f_m(l))^* = W \log_2 \left(1 + \frac{g_{m,d(l)} P_{\max}}{\eta W + P_{\max} \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)}} \right).$$

We will show that $(f_m(l))^*$ can be transmitted from node m to node $d(l)$, and for any set of $f_m(l)$ that can be transmitted to $d(l)$, we have $\sum_{m \in \mathcal{T}_{d(l)}} \hat{f}_m(l) \leq \sum_{m \in \mathcal{T}_{d(l)}} (f_m(l))^*$. Thus, an upper bound of the maximum receiving rate at $d(l)$ is $\sum_{m \in \mathcal{T}_{d(l)}} (f_m(l))^* = C_{d(l)}$.

First, we show that $(f_m(l))^*$ can be transmitted to $d(l)$. That is, there exists a scheduling and power control policy to transmit these $(f_m(l))^*$ to $d(l)$. As an existence proof for such a policy, let time slot 1 be used for receiving at $d(l)$ and $t_1 = 1$. Let each node $m \in \mathcal{T}_{d(l)}$ send its data to $d(l)$ via one hop with power $p_{m,d(l)}^1 = P_{\max}$. The link capacity $c_{m,d(l)}^1$ can be computed from (7). It is easy to verify that $(f_m(l))^* = c_{m,d(l)}^1$, that is, node m can send $(f_m(l))^*$ to $d(l)$.

Second, we show that this set of $(f_m(l))^*$ achieves the maximum receiving rate. That is, there does not exist another set of $\hat{f}_m(l)$ with a one-hop solution such that $\sum_{m \in \mathcal{T}_{d(l)}} \hat{f}_m(l) > \sum_{m \in \mathcal{T}_{d(l)}} (f_m(l))^*$.

For any given scheduling policy, we now consider a time slot k, $1 \leq k \leq K$. Since $\sum_{j \in \mathcal{T}_m} p_{mj}^k \leq P_{\max}$ in (1), we must have $p_{m,d(l)}^k \leq P_{\max}$. Then, $\frac{g_{m,d(l)} p_{m,d(l)}^k}{\eta W} \leq \frac{g_{m,d(l)} P_{\max}}{\eta W}$. However, according to (2), $P_{\max} = \frac{Q_{\max} W}{g_{\text{nom}}}$, we have $\frac{g_{m,d(l)} p_{m,d(l)}^k}{\eta W} \leq \frac{g_{m,d(l)}}{\eta W} \frac{Q_{\max} W}{g_{\text{nom}}} = \frac{Q_{\max} g_{m,d(l)}}{\eta g_{\text{nom}}}$. For UWB, we have $\frac{Q_{\max}}{\eta} \ll 1$ (for example, on the order of 10^{-2} [16]). Since $g_{m,d(l)}$ and g_{nom} are comparable, we have

$$\frac{g_{m,d(l)} p_{m,d(l)}^k}{\eta W} \ll 1 \quad \text{for } m \in \mathcal{T}_{d(l)}. \tag{15}$$

Denote $\xi = \max_{m \in \mathcal{T}_{d(l)}} \frac{g_{m,d(l)} p_{m,d(l)}^k}{\eta W}$. Then, we have $\xi \ll 1$. We now consider the total rate $\sum_{m \in \mathcal{T}_{d(l)}} c_{m,d(l)}^k$ received by $d(l)$ in time slot k. According to (7), we have $c_{m,d(l)}^k = t_k W \log_2 \left(1 + \frac{g_{m,d(l)} p_{m,d(l)}^k}{\eta W + \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)} p_{i,d(l)}^k} \right)$. Using (15), we have

$$c_{m,d(l)}^k \approx \frac{t_k W}{\ln 2} \frac{g_{m,d(l)} p_{m,d(l)}^k}{\eta W + \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)} p_{i,d(l)}^k}.$$

Thus,

$$\frac{\partial \left(\sum_{m \in \mathcal{T}_{d(l)}} c_{m,d(l)}^k \right)}{\partial \left(g_{m,d(l)} p_{m,d(l)}^k \right)} = \frac{t_k W}{\ln 2} \left[\frac{1}{\eta W + \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)} p_{i,d(l)}^k} - \sum_{j \in \mathcal{T}_{d(l)}}^{j \neq m} \frac{g_{j,d(l)} p_{j,d(l)}^k}{\left(\eta W + \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq j} g_{i,d(l)} p_{i,d(l)}^k \right)^2} \right].$$

To simplify the notation, denote $a = \eta W$ and $b = \sum_{i \in \mathcal{T}_{d(l)}} g_{i,d(l)} p_{i,d(l)}^k$. Then, we have

$$\frac{\partial \left(\sum_{m \in \mathcal{T}_{d(l)}} c_{m,d(l)}^k \right)}{\partial \left(g_{m,d(l)} p_{m,d(l)}^k \right)} = \frac{t_k W}{\ln 2} \left[\frac{1}{a + b - g_{m,d(l)} p_{m,d(l)}^k} - \sum_{j \in \mathcal{T}_{d(l)}}^{j \neq m} \frac{g_{j,d(l)} p_{j,d(l)}^k}{\left(a + b - g_{j,d(l)} p_{j,d(l)}^k \right)^2} \right]$$

$$\geq \frac{t_k W}{\ln 2} \left[\frac{1}{a + b} - \sum_{j \in \mathcal{T}_{d(l)}}^{j \neq m} \frac{g_{j,d(l)} p_{j,d(l)}^k}{(a + b - \xi a)^2} \right]$$

$$\geq \frac{t_k W}{\ln 2} \left[\frac{1}{a + b} - \frac{b}{(a + b - \xi a)^2} \right]$$

$$= \frac{t_k W}{\ln 2} \frac{[(1 - 2\xi)(a^2 + ab) + \xi^2 a^2]}{(a + b)(a + b - \xi a)^2} > 0.$$

The last inequality holds because $\xi \ll 1$. Since $\sum_{m \in \mathcal{T}_{d(l)}} c_{m,d(l)}^k$ is an increasing function of $p_{m,d(l)}^k$, to maximize the receiving rate at $d(l)$ in time slot k, we must have $p_{m,d(l)}^k = P_{\max}$ for all nodes $m \in \mathcal{T}_{d(l)}$ and all time slots k. Thus, to maximize the receiving rate at $d(l)$, the rate of any node $m \in \mathcal{T}_{d(l)}$ is

$$f_m(l) \leq \sum_{k=1}^{K} c_{m,d(l)}^k$$

$$= \sum_{k=1}^{K} t_k W \log_2 \left(1 + \frac{g_{m,d(l)} P_{\max}}{\eta W + \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)} P_{\max}} \right)$$

$$= W \sum_{k=1}^{K} t_k \log_2 \left(1 + \frac{g_{m,d(l)} P_{\max}}{\eta W + P_{\max} \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)}} \right) \quad (16)$$

$$= W \log_2 \left(1 + \frac{g_{m,d(l)} P_{\max}}{\eta W + P_{\max} \sum_{i \in \mathcal{T}_{d(l)}}^{i \neq m} g_{i,d(l)}} \right)$$

$$= (f_m(l))^*.$$

This completes the proof.   □

ACKNOWLEDGMENTS

REFERENCES

[1] P. Baldi, L. De Nardis, and M.-G. Di Benedetto, "Modeling and Optimization of UWB Communication Networks through a Flexible Cost Function," *IEEE J. Selected Areas in Comm.,* vol. 20, no. 9, pp. 1733-1744, Dec. 2002.

[2] N. Bambos, S. Chen, and G. Pottie, "Radio Link Admission Algorithms for Wireless Networks with Power Control and Active Link Quality Protection," *Proc. IEEE INFOCOM '95,* pp. 97-104, Apr. 1995.

[3] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms,* second ed. John Wiley & Sons, 1993.

[4] F. Cuomo, C. Martello, A. Baiocchi, and F. Capriotti, "Radio Resource Sharing for Ad Hoc Networking with UWB," *IEEE J. Selected Areas in Comm.,* vol. 20, no. 9, pp. 1722-1732, Dec. 2002.

[5] T. Elbatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad-Hoc Networks," *Proc. IEEE INFOCOM '02,* pp. 976-984, June 2002.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* pp. 245-248. W.H. Freeman, 1979.

[7] *IEEE 802.15 WPAN High Rate Alternative PHY Task Group 3a,* http://grouper.ieee.org/groups/802/15/pub/TG3a.html, 2007.

[8] N. Blefari-Melazzi, M.G. Di Benedettio, M. Geria, H. Luediger, M.Z. Win, and P. Withington, eds., *IEEE J. Selected Areas in Comm.,* special issue on ultra-wideband radio in multiaccess wireless comm., vol. 20, no. 9, Dec. 2002.

[9] *Ultra Wideband (UWB) Frequently Asked Questions (FAQ),* Multispectral Solutions, Inc., http://www.multispectral.com/, July 2003.

[10] R. Negi and A. Rajeswaran, "Capacity of Power Constrained Ad-Hoc Networks," *Proc. IEEE INFOCOM '04,* pp. 443-453, Mar. 2004.

[11] R. Negi and A. Rajeswaran, "Scheduling and Power Adaptation for Networks in the Ultra Wide Band Regime," *Proc. IEEE Global Telecomm. Conf. (Globecom '04),* pp. 139-145, 2004.

[12] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization,* pp. 354-367. John Wiley & Sons, 1999.

[13] R.C. Qiu, H. Liu, and X. Shen, "Ultra-Wideband for Multiple Access Communications," *IEEE Comm. Magazine,* vol. 43, no. 2, pp. 80-87, Feb. 2005.

[14] B. Radunovic and J.-Y. Le Boudec, "Rate Performance Objectives of Multi-Hop Wireless Networks," *Proc. IEEE INFOCOM '04,* pp. 1916-1927, Mar. 2004.

[15] B. Radunovic and J.-Y. Le Boudec, "Optimal Power Control, Scheduling, and Routing in UWB Networks," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 7, pp. 1252-1270, Sept. 2004.

[16] A. Rajeswaran, G. Kim, and R. Negi, "A Scheduling Framework for UWB & Cellular Networks," *Proc. First Int'l Conf. Broadband Networks (Broadnets '04),* pp. 386-395, Oct. 2004.

[17] J.H. Reed, *An Introduction to Ultra Wideband Communication Systems.* Prentice Hall, 2005.

[18] A. Rubinov and X. Yang, *Lagrange-Type Functions in Constrained Non-Convex Optimization.* Kluwer Academic Publishers, 2003.

[19] H.D. Sherali and W.P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems,* chapter 8. Kluwer Academic Publishers, 1999.

[20] L. Tassiulas and A. Ephremides, "Jointly Optimal Routing and Scheduling in Packet Radio Networks," *IEEE Trans. Information Theory,* vol. 38, no. 1, pp. 165-168, Jan. 1992.

[21] M. Win and R. Scholtz, "Ultra-Wide Bandwidth Time-Hopping Spread-Spectrum Impulse Radio for Wireless Multiple-Access Communications," *IEEE Trans. Comm.,* vol. 48, no. 4, pp. 679-691, Apr. 2000.

[22] H. Zhang and J. Hou, "Capacity of Wireless Ad-Hoc Networks under Ultra Wide Band with Power Constraint," *Proc. IEEE INFOCOM '05,* pp. 455-465, Mar. 2005.

Yi Shi received the BS degree in computer science from the University of Science and Technology of China, Hefei, China, in 1998, the MS degree in computer science from the Institute of Software, Chinese Academy of Science, Beijing, in 2001, and the second MS degree in computer science and the PhD degree in computer engineering from the Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, Virginia, in 2003 and 2007, respectively. He is currently a senior research associate in the Bradley Department of Electrical and Computer Engineering, Virginia Tech. His current research focuses on algorithms and optimization for cognitive radio wireless networks, MIMO and cooperative communication networks, sensor networks, and ad hoc networks. His work has appeared in some highly selective international conferences (ACM MobiCom, ACM MobiHoc, and IEEE INFOCOM) and IEEE journals. While an undergraduate, he was a recipient of Meritorious Award in the International Mathematical Contest in Modeling in 1997 and 1998, respectively. He was a recipient of the Chinese Government Award for Outstanding PhD Students Abroad in 2006. He is active in professional services. He was a TPC member for the IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks (held in conjunction with IEEE SECON 2006) and ChinaCom. He is a TPC member of the ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing 2008 (colocated with ACM MobiHoc 2008), IEEE ICCCN 2008, IEEE PIMRC 2008, IEEE MASS 2008, and IEEE ICC 2009. He is a member of the IEEE.

Y. Thomas Hou received the BE degree in electrical engineering from the City College of New York in 1991, the MS degree in electrical engineering from Columbia University in 1993, and the PhD degree in electrical engineering from Polytechnic University, Brooklyn, New York, in 1998. Since August 2002, he has been with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, Virginia, where he is now an associate professor. His current research interests are radio resource (spectrum) management and networking for software-defined radio wireless networks, optimization and algorithm design for wireless ad hoc and sensor networks, and video communications over dynamic ad hoc networks. From 1997 to 2002, he was a researcher at Fujitsu Laboratories of America, Sunnyvale, California, where he worked on scalable architectures, protocols, and implementations for differentiated services Internet, service overlay networking, video streaming, network bandwidth allocation policies, and distributed flow control algorithms. He was a recipient of a US Office of Naval Research (ONR) Young Investigator Award (2003) and a National Science Foundation (NSF) CAREER Award (2004) for his research on algorithm design and optimizations for wireless ad hoc and sensor networks. He has published extensively in leading journals and top-tier conferences and received several best paper awards from IEEE for his publications. He holds five US patents. He is active in professional services and is currently serving as an editor of the *IEEE Transactions on Wireless Communications*, *ACM/Springer Wireless Networks*, and *Elsevier Ad Hoc Networks Journal*. He was a past associate editor of the *IEEE Transactions on Vehicular Technology*. He was a cochair of the Technical Program Committee (TPC) of the Second International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM 2007). He was the founding chair of the First IEEE Workshop on Networking Technologies for Software Defined Radio Networks in 2006. He was a cochair (with Tony Ephremides) of the NSF Workshop on Bridging the Gap between Wireless Networking Technologies and Advances at the Physical Layer in 2007. He will serve as a cochair of the TPC of IEEE INFOCOM 2009. He is a senior member of the IEEE and the IEEE Computer Society.

Hanif D. Sherali is a university distinguished professor and the W. Thomas Rice chaired professor of engineering in the Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia. His areas of research interest are in analyzing problems and designing algorithms for specially structured linear, nonlinear, and integer programs arising in various applications, global optimization methods for nonconvex programming problems, location and transportation theory and applications, and economic and energy mathematical modeling and analysis. He has published more than 200 refereed articles in various operations research journals, has (co)authored six books in this area, and serves on the editorial board of eight journals. He is an elected member of the US National Academy of Engineering.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.