Optimization-based Design of Wireless Link Scheduling with Physical Interference Model

Shuai Fan, Lin Zhang, Wei Feng, Wenzhu Zhang, and Yong Ren

Abstract—We address the link scheduling problem in wireless multi-hop networks under the realistic physical interference model. Different from most work that adopt the protocol interference model that treats the pair-wise interference, we use the physical interference model to reflect the aggregated Signal to Interference and Noise Ratio (SINR), which is a more accurate abstraction of the real scenarios.

We first propose a centralized scheduling method based on the Integer Linear Programming (ILP) and get an approximate solution by relaxing it to Linear Programming (LP). The probability of getting the guaranteed approximate factor is given, which is distinguished from other LP based algorithms. Then, for the cases where the required global information is hard to get, a distributed scheduling algorithm is proposed. We simplify the formulation of the scheduling problem in the distributed scenarios, and calculate the optimal solution as the transmission probability. This method can be implemented on each node through the Jacobi algorithm, only relying on local channel information. Simulation results show that it converges fast to the optimal solution, and provide good throughput performance comparable to the centralized algorithm.

Index Terms—Wireless multi-hop networks, scheduling, physical interference model.

I. INTRODUCTION

In wireless networks, the problem of scheduling link transmissions in order to optimize one or more of performance objectives (e.g. throughput, delay, fairness or energy) under interference constraints has been a subject of much interests over the past decades. The concurrent transmissions that compete for the common channel may cause interference, which prevents an intended receiver from receiving the signal. But if too few nodes are allowed to transmit at the same time, valuable bandwidth is wasted and the overall throughput may suffer. Hence, the classic problem faced by the Media Access Control (MAC) layer or any scheduling protocols is to select an appropriate set of devices for concurrent transmissions, so that the interference does not cause message loss. In this paper, an important metric for wireless networks, namely, the throughput is selected to be maximized based on our proposed scheduling algorithms with the interference restriction.

In wireless communication, the signal is decoded by treating the sum of all the other concurrent signal transmissions as noise, and decoding success is an event with probability depending on the received signal strength, the strength of interference, and the thermal noise. All these factors have to be modeled with appropriate interference model to analyze and optimize the network performance. The interference model has been shown to have a major impact on the complexity and efficiency of link scheduling algorithms [1]–[4]. Two main interference models, namely, the protocol and the physical interference models [5], have been widely utilized.

In the protocol model, a communication from node $u$ to node $v$ is successful if no other node within the interference range from $u$ is simultaneously transmitting. Due to its simplicity and to the fact that it can be used to mimic the behavior of CSMA/CA networks such as IEEE 802.11, this model has been widely adopted in literature [6]–[9]. Specifically, [10], [11] established the constant-factor approximation bounds to justify the level of approximation for the protocol model. However, the protocol model only considers local, pairwise interference and neglects the aggregated interference of nodes out of the interference range.

In the physical interference model, a communication between nodes $u$ and $v$ is successful if the SINR at receiver $v$ is above some threshold defined by the radio receiver. This model is less restrictive than the protocol interference model, and thus higher network capacity can be achieved with it.

It is clear that the interference in the protocol model is a tremendous simplification of the physical reality in the wireless multi-hop networks. Particularly, the interference caused by different transmitters may accumulate and is thus not binary, i.e. interference does not vanish at any deterministic border. Moreover, when transmission powers are properly assigned, a node may successfully receive a packet even when there are other transmitting nodes in its interference range. In [1], [3], [4], it is argued that the performance of protocol model based algorithms is inferior to those with more realistic physical interference model. More recently, the work in [2], [12] show experimentally that the theoretical limits of any protocol under the protocol interference model can be easily broken by adopting a protocol under the physical interference model. Unfortunately, this does not come for free as the scheduling problem under the physical model is notoriously hard.

Only a few previous work have considered physical interference in the context [13]–[21], etc. In [13], Jain, et al. formulate the problem of scheduling under the physical interference model as an LP problem. Unfortunately, no polynomial time solution or simulation-based evaluation of scheduling is given. [14] also provides an exponential-time LP formulation. In [15], Brar et al. present a heuristic scheduling method that is based on a greedy assignment of weighted links. Although
it is based on physical interference model, the approximation factor of the algorithm is given only when nodes are uniformly distributed. [16] considers the physical interference in the minimum length scheduling problem. It is based on a power-based interference graph, which describes the interference relationship of every two links according to the SINR of the receiver. However, the power-based interference graph does not consider the accumulation effect of interference. In [17], approximation algorithms for packet scheduling to minimize end-to-end delay with the physical interference model are proposed; [18], [19] study the problem of scheduling edges with SINR constraints to ensure that some property (e.g., connectivity) is satisfied. However, they take power control, scheduling or routing together into account, which are similar with [20], [21], [22] proposes approximation algorithms for link scheduling with SINR-based models, and presents upper bound of the scheduling length. However, the approximation factor is not given. An O(log n) approximation algorithm is proposed in [23] with physical interference model, which doesn’t consider the constraints of fairness and half-duplex.

In this paper, we study the centralized link scheduling problem under the physical interference model. First, we formulate the problem as an Integer Linear Programming (ILP) problem, which is similar to the model in [24]. Since it is NP-complete [25], we formulate an approximate solution by relaxing it to be a Linear Programming (LP). The probability bound of getting the guaranteed approximate factor is given, which is distinguished from other LP based algorithms. Unlike [24], we adjust the solution of the relaxed LP based on all the constraints, and no iteration procedure is needed.

The centralized algorithm requires global information for the scheduling, including the location of all nodes in the network, the loads of the links in the time slot t, etc.. Collecting global information is demanding, especially when the network scales. In some cases, it is hardly possible so that a distributed scheduling scheme that may achieve a fraction of the globally optimal performance is acceptable. We simplified the formulation of the scheduling problem in the distributed scenarios, and establish the optimal solution in form of the transmission probability. This method can be implemented distributively by each node through the Jacobi algorithm, and only local channel information is needed.

Various distributed algorithms have been proposed for finding good approximations of the scheduling problem based on the protocol interference model (e.g. [26]–[28], etc.). Only a few previous work propose distributed algorithms based on the physical interference model. [29]–[31] propose distributed algorithm based on the physical interference model, which is lattice-throughput-optimal. But the approximation ratio is not given. The work of [32] develop a constant-time distributed random access algorithm for scheduling and give the performance bound of the algorithm. In [33], a distributed and randomized protocol is proposed, which uses physical carrier sensing to reduce message overhead. But the protocol is sensitive to the scale of the network and spends much time to learn the node density. The authors in [34] proposed both centralized and distributed approaches of random access schemes. They are optimization-based design with the physical interference model which is similar with ours. But It does not address how to find the feasible sets in which all links do not interfere with each other.

The rest of the paper is organized as follows. The network and interference models are described in Section II. In Section III, scheduling problem under the physical interference model is formulated and the centralized algorithm to get the approximate solution is presented. In section IV, we moved on to present the distributed approach. Section V presents the simulation results of the proposed algorithms and Section VI concludes the paper.

II. NETWORK AND INTERFERENCE MODELS

A wireless multi-hop network is represented by a directed graph G(V, E), where V is the set of vertices denoting the nodes and E is the set of directed edges between vertices representing wireless links. Let e_{ij} ∈ E denotes the edge existing between v_i and v_j.

Each node is equipped with a single transceiver working in the half-duplex way, and all nodes share a common channel. The antennas of all nodes are assumed to be omnidirectional.

Assuming that the network working in the Time Division Multiple Access (TDMA) mode, and time is divided into slots of fixed length, and are grouped into frames. To increase the capacity, spatial reuse TDMA (STDMA) [35], allowing multiple transmissions in one time slot when interference constraint is satisfied.

We use the physical interference model [5] to describe the interferences between active links, where the successful reception of a transmission depends on the received signal strength, the interference caused by nodes transmitting simultaneously, and the ambient noise level. The received power P_r(s_i) of a signal transmitted by sender s_i at an intended receiver r_i is

\[ P_r(s_i) = P(s_i) \cdot G(s_i, r_i), \]

where P(s_i) is the transmission power of s_i and G(s_i, r_i) is the propagation attenuation (link gain). Given a sender-receiver pair (s_i, r_i), we use the notation I_r(s_j) = P_r(s_j) for any other sender s_j concurrent to s_i. The total interference I_r experienced by receiver r_i is the sum of the interference power values created by all nodes in the network transmitting simultaneously (except the intending sender s_i), that is, \[ I_r := \sum_{s_j \in V \setminus \{s_i\}} I_r(s_j). \] Finally, let N denote the ambient noise power level. Then, r_i receives s_i transmission if and only if

\[ SINR(r_i) = \frac{P_r(s_i)}{N + \sum_{s_j \in V \setminus \{s_i\}} I_r(s_j)} \leq \beta, \]

where \( \beta \) is the minimum \( SINR \) threshold required for a successful reception.
A. Problem formulation

We denote by $V_T \subseteq V$ and $V_R \subseteq V$ the set of transmitting and receiving nodes respectively. The time is divided into slots of fixed duration and a frame is composed of constant over $t \in [1, T]$, there exists at least one edge being scheduled to transmit. We use the sum rate of all transmitting links to measure throughput, and denote by $b_{ij}$ the total traffic rate through link $e_{ij} \in E$. Here, it is assumed that $b_{ij}$ is constant over $t$ in a frame, and it can vary in different frames. And different links can set different link rates $b_{ij}$. The goal of the scheduling method is to transmit all the edges in a frame to gain the maximum throughput. So the objective of the formulation is as follows:

$$\max \sum_{t=1}^{T} \sum_{e_{ij} \in E} b_{ij} x_{ij}^t.$$  \hfill (1)

To formally formulate the problem, the boolean variables $x_{ij}^t$ is also introduced which are defined as follows,

$$x_{ij}^t = \begin{cases} 1 & \text{if link } e_{ij} \text{ is scheduled to be transmit in slot } t \\ 0 & \text{otherwise} \end{cases}$$

Fairness is very important for wireless networks. There are several fairness criteria in the literature, e.g., max-min fairness, proportional fairness, etc. These fairness criteria have nonlinear optimization objectives, which makes the problem hard to resolve. To remediate the problem, we add a constraint to guarantee the fairness, and keep the objective function linear. The constraint guarantees that each active edge should be scheduled at least once. It is described as

$$\sum_{t=1}^{T} x_{ij}^t \geq 1, \forall e_{ij} \in E.$$  \hfill (2)

For simplicity, it is assumed that each node has only one transceiver, so each node can only send or receive signal to or from one another node. The constraints are described as follows:

$$\sum_{v_i \in V_T} x_{ij}^t \leq 1, \forall v_i \in V_R, \forall t,$$  \hfill (3)

$$\sum_{v_j \in V_R} x_{ij}^t \leq 1, \forall v_i \in V_T, \forall t.$$  \hfill (4)

There are cases in which each node has multi-transceivers. The only differences in the constraint (3) and (4) are that the right side of the inequalities are the numbers of transceivers equipped in each node. Each node can not send and receive at the same time because it works in a half-duplex way. As a result,

$$\sum_{v_j \in V_R} x_{ij}^t + \sum_{v_k \in V_T} x_{kj}^t \leq 1, \forall v_i \in V_T \cap V_R, \forall t.$$  \hfill (5)

III. THE CENTRALIZED ALGORITHM

In this section, we first formulate the link scheduling problem under the physical interference model as an Integer Linear Programming problem, then give an approximate algorithm using randomized rounding method, which can be done in polynomial time.

A. Problem formulation

The only differences in the constraint (3) and (4) are that the constraint (3) is satisfied. But after the rounding procedure, some of the constraints may no longer be satisfied. For example, suppose $\hat{x}_{ij}$ is close to one, it is likely that the link will be included in the current time slot, and vice versa. Specifically, the rounding procedure can be more formally described as follows,

$$x_{ij}^t = \begin{cases} 1, & \text{with probability } \hat{x}_{ij} \\ 0, & \text{with probability } 1 - \hat{x}_{ij} \end{cases}$$  \hfill (9)

To express the required SINR threshold that should be satisfied for a successful reception at the receiver, we have the following constraint.

$$P(v_i)G(v_i, v_j) + (1 - x_{ij}^t)\Delta \geq \beta, \forall e_{ij} \in E, \forall t.$$  \hfill (6)

The item $(1 - x_{ij}^t)\Delta$ ensures that the inequality is also satisfied when link $e_{ij}$ is not scheduled in time slot $t$ (i.e. $x_{ij}^t = 0$), for a sufficiently large value of $\Delta$. The following constraint guarantees that the variables $x_{ij}^t$ are boolean:

$$x_{ij}^t \in \{0, 1\}, \forall e_{ij} \in E, \forall t.$$  \hfill (7)

With the above constraints, the link scheduling problem under the physical interference model is formulated as an ILP problem, which comprises of objective function (1) and constraints (2) - (7).

B. Approximate algorithm with randomized rounding

Since ILP problems are NP-complete, there is no efficient algorithm known for solving them in bounded time (and there can not exist any unless $P = NP$). The above formulation does not help us solve the scheduling problem. However, it does guide us to a natural relaxation which helps us find a good approximate algorithm.

1) Randomized rounding procedure: We can use randomized rounding method to find efficient and near optimal solutions. In the LP-relaxation each of the variables $x_{ij}^t$ will be in the range from zero to one and therefore converted from integer variables to fractional ones:

$$x_{ij}^t \in [0, 1], \text{ for } \forall e_{ij} \in E, \forall t.$$  \hfill (8)

Equation (1) - (6) and (8) make up the linear programming-relaxation (LP-relaxation) problem. The optimal solution of the LP-relaxation problem can be treated as a probability vector with which we choose to include a link to a specific time slot. Denote the optimal solution as $\hat{x}_{ij}$. In that sense, if $\hat{x}_{ij}$ is close to one, it is likely that the link will be included in the current time slot, and vice versa. Specifically, the rounding procedure can be more formally described as follows,

$$x_{ij}^t = \begin{cases} 1, & \text{with probability } \hat{x}_{ij} \\ 0, & \text{with probability } 1 - \hat{x}_{ij} \end{cases}$$  \hfill (9)

However, after the rounding procedure, some of the constraints may no longer be satisfied. For example, suppose in constraint (3), for $\forall e_{ij} \in V_R, \forall t$, there are two nodes $v_1, v_2 \in V_T$. If $\hat{x}_{ij} = 0.3$ and $\hat{x}_{ij} = 0.7$, constraint (3) is satisfied. But after the rounding procedure, $x_{ij} = 1$ with probability 0.3, and $x_{ij} = 1$ with probability 0.7. So $\sum_{v_i \in V_T} x_{ij} = 2$ with probability 0.21. Therefore, with probability 0.21, constraint (3) is violated. To overcome this problem, the work in [24] uses an iteration method. The rounding procedure would not stop until a feasible solution is found, which costs a very long time to converge. Even worse, it may not find any feasible solution in the end. Unlike [24], we adjust the results of the rounding procedure based on all the constraints, and no iteration procedure is needed.
2) Approximate algorithm: we denote by \( \hat{\phi}^t = \{ e_{ij} : x^t_{ij} = 1 \} \) the set of edges that will be transmitted after the rounding procedure at time slot \( t \). As discussed above, the constraints of the ILP programming problem may not be satisfied. So the solution of the rounding procedure will be adjusted in Algorithm 1 to satisfy all the constraints. As the solution of the relaxed LP problem \( \{ \hat{x}^t_{ij} : \forall e_{ij} \in E, t \in T \} \) is the probability for the wireless links to transmit, we reorder the elements in \( \hat{\phi}^t \) according to the probabilities in the non-decreasing order, and denote the new set by \( \phi^t \).

Algorithm 1 proceeds in two phases. In Phase 1, the integer programming problem is relaxed to be a linear programming problem which is solved by the randomized rounding procedure. As mentioned before, there exist cases that the rounding solution \( \{ x^t_{ij} : \forall e_{ij} \in E, t \in T \} \) may not satisfy all the constraints in the ILP problem. In Phase 2, the solution of the relaxed LP \( \{ \hat{x}^t_{ij} : \forall e_{ij} \in E, t \in T \} \) is viewed as the probability of the transmission of the links. The links are checked in a non-decreasing order of \( \hat{x}^t_{ij} \). In each slot, if a link doesn’t satisfy any of the constraints of (2), (3) or (4), it will not transmit. After all links in \( \hat{\phi}^t \) are checked, constraint (5) is finally checked because it is correlated to all the other transmission schedules. Constraint (1) guarantees that each active link should be scheduled at least once in a frame. In some cases, all \( x^t_{ij} \) may be rounded, or adjusted from 1 to 0, leading to violation to constraint (1). When this happens, find the one with the largest probability to transmit.

3) Performance analysis: In this part, the performance of Algorithm 1 is analyzed, including the complexity and the approximate ratio. Denote the throughput of each slot. [23] shows that the per-slot throughput of each slot is \( T_h \text{opt} \) and that of Algorithm 1 by \( T_h \). The approximation ratio of Algorithm 1 is

\[
1 - \theta = \frac{T_h}{T_h \text{opt}}.
\]

We say Algorithm 1 is a \( 1 - \theta \)-approximation algorithm.

The complexity of phase 1 in Algorithm 1 depends on the algorithm solving the LP-relaxed problem. It is \( O(P) \) because it is a linear programing problem, where \( P \) is polynomial. Let \( n \) be the total number of the links in \( E \). It can be easily seen that the complexity from line 5 to 15 in Algorithm 1 is \( O(nT) \), because there are two loops. Similarly, the complexity from line 16 to 22 is \( O(n^2T) \). Thus the complicity of the whole algorithm is \( O(max(P, n^2T)) \), which is polynomial.

The approximate ratio of Algorithm 1 is due to the following theorem:

**Theorem 1:** For \( 0 < \theta < 1 \), the probability of Algorithm 1 being \( (1 - \theta) \)-approximate to the optimization is lower bounded by \( 1 - e^{-\frac{\Delta A}{12}} \), where \( \bar{A} \) is the throughput calculated by the LP-relaxation, and \( \Delta A \) is the variation of the throughput in phase 2 of Algorithm 1.

**Proof:** See Appendix A.

IV. DISTRIBUTED ALGORITHM

The centralized algorithm discussed in section III uses the physical interference model to achieve good performance. But it needs global information for the scheduling, including the location of every node, the knowledge of which links need to be scheduled in a certain time slot \( t \), etc. Collecting these information may waste a lot of bandwidth, and in some cases, even impossible.

In this section, we propose a fully distributed algorithm which also adopts the physical interference model into account. We simplified the optimization problem in the distributed scenario, and achieve the optimal solution in terms of the transmission probability for each node. This method can be implemented in a distributed manner on each node where only local channel information is present.

A. Simplified formulation of the distributed algorithm

In the centralized algorithm, the objective function is to maximize the average throughput in each frame. In the distributed algorithm we will instead maximize the expected throughput of each slot. [23] shows that the performance gap between the single slot optimization and the frame-wise optimization is \( O(\log n) \). But the former is much easier for distributed setups, so it is rational use the single

---

**Algorithm 1: Centralized Algorithm**

**Phase 1:**
1. Consider all links in \( E \);
2. Relax the integer linear programming problem to be a linear programming problem (LP-relaxation);
3. Find the solutions of the relaxed LP \( \{ \hat{x}^t_{ij} : \forall e_{ij} \in E, t \in T \} \);
4. Do randomized rounding and get the rounding solution \( x^t_{ij} : \forall e_{ij} \in E, t \in T \) from equation (9);

**Phase 2:**
5. for each \( t \in T \) do
6. Get the set \( \hat{\phi}^t = \{ e_{ij} : x^t_{ij} = 1 \} \);
7. Consider all links \( e_{ij} \in \hat{\phi}^t \) in non-decreasing order of \( \hat{x}^t_{ij} \); if constraint (3) or constraint (4) or constraint (5) is not satisfied then
8. \( x^t_{ij} = 0; \)
9. Move \( e_{ij} \) away from \( \hat{\phi}^t \);
10. end
11. Consider all links \( e_{ij} \in \hat{\phi}^t \) in non-decreasing order of \( \hat{x}^t_{ij} \); if constraint (5) is not satisfied then
12. \( x^t_{ij} = 0; \)
13. Move \( e_{ij} \) away from \( \hat{\phi}^t \);
14. end
15. end
16. Consider all links \( e_{ij} \in E \);
17. for each link \( e_{ij} \in E \) do
18. if \( \sum_{t=1}^{T} x^t_{ij} < 1 \) then
19. Consider the biggest element \( \hat{x}^t_{ij} \) for all \( t \in T \);
20. \( x^t_{ij} = 1, \hat{x}^t_{ij} = 1; \)
21. execute from line 5 to 15;
22. end

---

Copyright (c) 2011 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.
slot optimization at the price of very small performance deterioration.

The maximization of the throughput of each slot is then:

\[
\max \sum_{e_{ij} \in E} b_{ij} x_{ij}.
\]

In the distributed protocol, if a node has already sent a packet in a frame, it will set the transmission probability to zero for the rest of the frame, unless it observes no other nodes are occupying the channel, when the sending probability will be reset to the value computed by the distributed algorithm. As a result, constraint (2) is naturally satisfied. Furthermore, each receiving node can select one sending randomly from the sending nodes set, and compute corresponding sending probability using its signal strength. The sending probability of other nodes in the sending set will be set to be 0. As a result, constraint (3) is satisfied, and similarly is constraint (4). For constraint (5), the analysis is the same as that for constraint (3). The difference is to select the active sending node set and the receiving nodes set respectively. As a result, the scheduling problem can be simplified as follows in the distributed scenario:

\[
\max \sum_{e_{ij} \in E} b_{ij} x_{ij} + \frac{P(v_i)G(v_i, v_j) + (1 - x_i^T \Delta) N}{N + \sum_{e_{ij} \in E \setminus \{e_{ij}\}} P(v_k)G(v_k, v_j)x_k} \geq \beta, \forall e_{ij} \in E, \forall t.
\]

\[
x_{ij} \in \{0, 1\}, \forall e_{ij} \in E
\]

Obviously, it is still an ILP problem. As discussed in the centralized algorithm, the ILP can be relaxed to be LP, and the solution of the ILP can be viewed as the probability of scheduling the link to transmit. Because it is in a distributed situation, each node needs to calculate the probability individually based on its local information. Therefore, how to compute the above problem in a distributed way is the key issue.

Suppose there are \(n\) links \([e_1, e_2, \ldots, e_n]\) to be scheduled. Denote the link rate vector by \(b = [b_1, b_2, \ldots, b_n]^T\), and the vector form of the relaxed variables by \(x = [x_1, x_2, \ldots, x_n]^T\).

The constraint matrix is defined as:

\[
A_n = \begin{pmatrix}
\Delta - PG_{11} & \beta PG_{12} & \ldots & \beta PG_{1n} \\
\beta PG_{12} & \Delta - PG_{22} & \ldots & \beta PG_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\beta PG_{1n} & \beta PG_{2n} & \ldots & \Delta - PG_{nn}
\end{pmatrix},
\]

where \(G_{ij}\) is the link gain between node \(i\) and \(j\), e.g., \(G_{ij} = G(s_i, r_j)\). The optimization can then be rewritten as:

\[
\max b^T x
\]

\[
s.t. \ A_n x + \beta N - \Delta \leq 0
\]

\[
0 \leq x \leq 1
\]

Definition 1: A \(m \times m\) matrix \(M\) is strictly row diagonally dominant, if \(|a_{ii}| > \sum_{j \neq i} |a_{ij}|\) for all \(i = 1, \ldots, m\), where \(a_{ij}\) denotes the entry in the \(i\)-th row and \(j\)-th column. Matrix \(M\) is strictly column diagonally dominant, if \(M^T\) is strictly row diagonally dominant, where \(M^T\) is the transposed matrix of \(M\).

The properties of the matrix \(A_n\) defined in equation (11) are described in the following lemmas:

**Lemma 1**: The matrix \(A_n\) is invertible.

**Proof**: See Appendix B.

**Lemma 2**: The determinant of matrix \(A_n\), \(\det(A_n)\), is strictly positive.

**Proof**: See Appendix C.

### B. Optimal solution of the simplified formulation

To get the optimal solution of above simplified problem, the Lagrangian duality is used to augment the objective function with a weighted sum of the constraint functions. Define \(c = [c_1, \ldots, c_n]^T = [\Delta - \beta N, \ldots, \Delta - \beta N]^T\), and \(g_n(x) = [g_1(x), g_2(x), \ldots, g_n(x)]^T = A_n x + \beta N - \Delta = A_n x - c\). The Lagrangian associated with problem (12) is defined as:

\[
L(x, \omega, \lambda, \nu) = -bx^T + \omega^T g_n(x) + \lambda^T x - \lambda - \nu^T x
\]

where vector \(\omega = [\omega_1, \ldots, \omega_n]^T\) is referred to as the Lagrange multiplier associated with the first constraint in problem (12); \(\lambda = [\lambda_1, \ldots, \lambda_n]^T\) and \(\nu = [\nu_1, \ldots, \nu_n]^T\) are the Lagrange multipliers associated with the second constraint in problem (12).

According to the Karush-Kuhn-Tucker (KKT) conditions [38], the optimal solution must satisfy the following conditions:

\[
\nabla_x L(x, \omega, \lambda, \nu) = 0
\]

\[
g_i(x) \leq 0, \quad \forall 1 \leq i \leq n
\]

\[
\omega_i g_i(x) = 0, \quad \forall 1 \leq i \leq n
\]

\[
\lambda_i (x_i - 1) \leq 0, \quad \forall 1 \leq i \leq n
\]

\[
\nu_i x_i = 0, \quad \forall 1 \leq i \leq n
\]

\[
0 \leq x_i \leq 1, \quad \forall 1 \leq i \leq n
\]

Based on equation (13), condition (14) can be rewritten as:

\[
A_n \omega = b - \lambda + \nu.
\]

According to Lemma 1, matrix \(A_n\) is invertible, we have:

\[
\omega = A_n^{-1}(b - \lambda + \nu).
\]

Therefore we have the following Theorem:

**Theorem 2**: Let \(\hat{x}\) be the solution of \(g_n(x) = 0\), and the optimal solution of the LP problem (12) be \(x^*\). For \(\forall 1 \leq i \leq n\), if \(\hat{x}_i\) satisfies condition (19), there exists a dual optimal point \((\omega^*, \lambda^*, \nu^*)\) which guarantees that the optimal solution \(x^* = \hat{x}\).

**Proof**: If \(g_n(x) = 0\), it is obvious that (15) and (16) hold. \(\forall 1 \leq i \leq n\), let \(\lambda_i^* = 0\) and \(\nu_i^* = 0\), and condition (17) and (18) are satisfied.
Next we will show that there exists an unique $\omega^* > 0$ that satisfies condition (22). Denote the entry in the $i$-th row and $j$-th column of matrix $A_n$ by $a_{ij}$, and the entry in the $i$-th row and $j$-th column of $A_n^{-1}$ by $a_{ij}$. Since the diagonal entries of $A_n^{-1}$ is calculated as $\alpha_{ii} = (-1)^i \text{det}(A_{ii})/\text{det}(A)$, according to lemma 2 and deduction, we have $\alpha_{ii} > 0$. Similar to the proof of lemma 1, it can be shown that matrix $A_n$ is a strictly column diagonally dominant matrix. So $A_n^{-1}$ is a strictly row diagonally dominant matrix [37]. Then according to the definition of the diagonally dominant matrix, for all $i = 1, \ldots, n$, it follows that $|\alpha_{ii}| > \sum_{j \neq i} |\alpha_{ij}|$. Because $\alpha_{ii} > 0$, we can get
\[
\sum_{j=1}^{n} \alpha_{ij} > 0.
\]
(23)

For simplicity, suppose that all links transmit at the same rate, i.e. vector $b$ has equal entries $b$ (in the case that the links do not transmit at the same rate, the same conclusion can reached). So according to condition (22), for $\forall 1 \leq i \leq n$, it can be deduced that
\[
\omega^*_i = \sum_{j=1}^{n} \alpha_{ij}b_j = b(\sum_{j=1}^{n} \alpha_{ij}) > 0,
\]
in which inequality (24) follows because of (23). So there exists a dual optimal point $(\omega^*, \lambda^*, \nu^*)$ which guarantees that all the Karush-Kuhn-Tucker (KKT) conditions [38] are satisfied, and the optimal solution $x^* = \hat{x}$.

Theorem 2 gives the optimal solution of the problem (12) when the solution of $g_n(x) = 0$ satisfies condition (19). However, there exists the case in which condition (19) is not satisfied. A natural way of dealing with this situation is to restrict related optimization variables at the border of the constraint set, refer to Algorithm 2 for details.

Algorithm 2 proceeds in an iterative way. Firstly the algorithm calculates the solution $\hat{x}$ of $g_n(\hat{x}) = 0$. For some $1 \leq i \leq n$, if $\hat{x}_i > 1$, restrict it to be 1. At the same time, delete the $i$-th row and the $i$-th column of matrix $A_n$, and form a low dimension matrix $A_{n-1}$ which corresponding to $g_{n-1}(x)$. Suppose there are $j$ variables of $\hat{x}$ violating condition (19). After all the variables belonging to $\hat{x}$ checked, we get $A_{n-j}$ and $g_{n-j}(x)$. Then let $g_{n-j}(x) = 0$, and the iteration procedure continues. It will not stop until $\hat{x}$ satisfies condition (19). Actually, the solution $\hat{x}$ satisfies $\hat{x} > 0$ (see Lemma 3 in the following). So Algorithm 2 only deals with the case of $\hat{x}_i > 1$.

Lemma 3: The solution of $g_n(x) = 0$ satisfies $x > 0$.

Proof: See Appendix D.

We can claim the optimality of the solution of Algorithm 2 to the problem (12) by the following theorem.

Theorem 3: The solution $x^*$ by Algorithm 2 is optimal to problem (12).

Proof: See Appendix E.

Algorithm 2: Optimal Solution Calculation

<table>
<thead>
<tr>
<th>Input: $g_n(x): A_n, c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: optimal solution $x^*$</td>
</tr>
<tr>
<td>1 $(x^<em>, k(x)) = \text{Calculate}(g_n(x), x^</em>)$</td>
</tr>
<tr>
<td>2 while $x^*$ doesn’t satisfy condition (19) do</td>
</tr>
<tr>
<td>3 $(x^<em>, k(x)) = \text{Calculate}(k(x), x^</em>)$</td>
</tr>
<tr>
<td>4 end</td>
</tr>
<tr>
<td>5 $x^*$ is the optimal solution of problem (12);</td>
</tr>
</tbody>
</table>

Function: Calculate $(k(x), x^*)$

| 6 Denote the dimension of $k(x)$ by $m$; |
| 7 Let $k(x) = 0$, and get the solution as $\hat{x}$; |
| 8 for $i = 1$ to $m$ do |
| 9 if $\hat{x}_i > 0$ then |
| 10 $\hat{x}_i = 1$; |
| 11 for $j = 1$ to $m$ do |
| 12 $c_j = c_j - \beta PG_j$ // Updating $c$; |
| 13 end |
| 14 Delete the $i$-th row and $i$-th column of matrix $A_m$ // Corresponding $k(x)$ also changes; |
| 15 end |
| 16 end |
| 17 Let the corresponding entries of $x^*$ be those of $\hat{x}$; |
| 18 Return $k(x), x^*$ |

C. Distributed algorithm

The optimal solution of the scheduling problem (12) can be calculated by Algorithm 2. A key step in the algorithm is to calculate the solution of $g_n(x) = 0$, i.e. $A_n x = c$. Traditionally, all the entries of matrix $A_n$ need to be known to solve the linear equation. It implies that each node should know the channel gains of all the links (including the interfering links). However, this is too strong an assumption for distributed cases, in this section, we will propose a method to solve the linear equation in an iterative merely with the local channel information.

1) Distributed computation of the optimal solution: We assume that each receiving node can measure the power received from all sending nodes, meaning that each node knows the entries of the corresponding row of matrix $A_n$, with which the corresponding sending probability can be computed. The Jacobi algorithm [39] is used for each node to compute corresponding sending probability, which will be described in detail below.

Based on equation $A_n x = c$, for $\forall 1 \leq i \leq n$, we have
\[
\sum_{j=1}^{n} a_{ij}x_j = c_i,
\]
Because $a_{ii} \neq 0$, we have
\[
x_i = -\frac{1}{a_{ii}}\left(\sum_{j \neq i} a_{ij}x_j - c_i\right). \quad (25)
\]
If all components $x_j, j \neq i$, of the solution of $A_n x = c$ are known, the remaining component $x_i$ can be determined from equation (25). Naturally, Based on approximation of $x_j$,
$j \neq i$, we can also use equation (25) to obtain an estimate of $x_i$. This can be done for each component of $x$ simultaneously by using the most recent history values (Jacobi algorithm): starting with some initial vector $x(0) \in \mathbb{R}^n$, evaluate $x(t)$, $t = 1, 2, \ldots$, using the iteration

$$x_i(t + 1) = \frac{1}{a_{ii}} \left( \sum_{j \neq i} a_{ij} x_j(t) - c_i \right). \quad (26)$$

The Jacobi algorithm produces an infinite sequence $x(t)$. If this sequence converges to a limit $x$, then we have

$$\lim_{t \to \infty} x_i(t + 1) = \frac{1}{a_{ii}} \left( \sum_{j \neq i} a_{ij} \lim_{t \to \infty} x_j(t) - c_i \right) = x_i,$$

which is equivalent to $x$ being a solution of $A_n x = c$. The convergence of the Jacobi algorithm is guaranteed by the following theorem:

**Theorem 4:** Let $x^*$ satisfy $A_n x^* = c$, and let $x(t)$ be the sequence generated by equation (26). Then $\lim_{t \to \infty} x(t) = x^*$.

**Proof:** Let $D$ be a diagonal matrix whose diagonal entries equal to the corresponding diagonal entries of $A_n$, and let $B = A_n - D$, so that $B$ is zero along the diagonal. According to equation (26), we have

$$x(t + 1) = -D^{-1} B x(t) + D^{-1} c.$$

Denote $D^{-1} B$ by $H$, and let $h_{ij}$ be the entries of $H$. Then,

$$h_{ij} = \begin{cases} -a_{ij}/a_{ii} & i \neq j \\ 0 & i = j \end{cases}$$

Considering the row-sum norm of $H$, it follows that

$$\|H\|_1 = \max_i \left( \sum_{j=1}^n |h_{ij}| \right) = \max_i \left( \frac{1}{a_{ii}} \sum_{j \neq i} |a_{ij}| \right) < 1, \quad (27)$$

The inequality (27) follows because matrix $A_n$ is a strictly row diagonally dominant matrix. According the proposition in [39], the spectral radius $\rho(H)$ of matrix $H$ satisfies $\rho(H) \leq \|H\|_1 < 1$, and Theorem 4 follows.

2) **Implementation of the distributed algorithm:** To calculate the optimal transmission probability, each node should measure the receiving power from all the other nodes, and know the sending probabilities of other sending nodes. To get a more concrete sense of the implementation of the distributed algorithm, we depict in Fig. 1 an example of the frame structure. A frame is composed of fixed length time slots. There are two kinds of slots, namely, P-slots and D-slots. Each P-slot is divided into 2 phases, namely, probing phase and data sending phase. The probing phase is used to exchange probability information, and the data sending phase is used to send data. D-slot is also used to transmit data. We will present the details of the approach as following.

The probing phase of a P-slot is divided into $k$ sub-slots. At the beginning of each P-slot, each sending node randomly selects a sub-slot and sends the probability calculated in the previous sub-slot. In other sub-slots, the node will probe the channel, and receive the probability information which sent by other nodes. The probability signal has two effects: one is to broadcast the sending probability of the node; the other is that other nodes can get the corresponding channel gain by measuring the receiving power of the probability signal. After the probing phase, each link can calculate its sending probability using the Jacobi algorithm. The iteration procedure proceeds for every P-slot. If any two links transmit simultaneously in any of the sub-slots, it is not possible for the nodes of other links to get the sending probability information and measure the receiving power of the probability signal correctly. So the iteration procedure of the Jacobi algorithm will not be executed in this P-slot, and the convergence speed of the algorithm will suffer. It can reduce the probability of collision by randomly selecting sub-slots to transmit. There are several issues about the protocol parameters that worth note:

- **The power of sending the probability information.** Because in multi-hop networks, the distance between nodes may be very long, the probability information may not be obtained by other nodes because of fading. A solution is to use higher power level to transmit the probability information in the probing phase of each P-slot, which can be several times of the normal ones. But in the data sending phases of the P-slots and in the D-slots, a normal power level is used.

- **The number of sub-slots in the probing phase of each P-slot.** Denote the number of sub-slots in the probing phase of each P-slot by $k$. As mentioned before, the convergence speed of the algorithm will be slowed down if any two links transmit simultaneously in any of the sub-slots. To reduce the probability of collision, $k$ needs to be larger. But with the increase of $k$, the overheads increase too, causing the decrease of the throughput. So there exits a tradeoff by selecting proper value of $k$. Actually, $k$ is selected to be larger than $n$ in practical situation. The exact relationship between $n$ and the optimal $k$ is out of the scope of this paper and could be considerer as future work.

- **The number of P-slots.** As described above, the iteration proceeds in the P-slots, so the number of P-slots influences the convergence speed of the distributed algorithm. If there are a large number of P-slots in a frame, some of the P-slots will be wasted after the algorithm con-
verges, and the overhead also increases. As a result, the throughput will drop. On the other hand, if the number of P-slots in a frame is very small, the algorithm will converge after many frames, and the throughput will drop too. So selecting proper number of P-slots in each frame is important. The influence to the throughput can be seen in simulation.

V. SIMULATION

In this section, we present the simulation results that verify the performance of our proposed algorithms in this paper.

A. Simulation scenario

We create a scenario where 25 wireless nodes are uniformly distributed in a $350m \times 350m$ square, and the average distance between adjacent nodes to be 70 meters. For simplicity, all nodes are supposed to transmit at the same power of $9dBm$. We also assume $N = -90dBm$, $\beta = 10dB$, which are similar with the work of [3] and [29]. It is assumed that the link gain follows Rayleigh distribution. The gain $G_{ij}$ of link $e_{ij} \in E$ is a complex, independent and zero mean Gaussian random variable and it follows that $E(|G_{ij}|^2) = d_{ij}^\alpha$ in which $\alpha$ is the path-loss exponent. The flat fading in a frame is also assumed, and the channel gain does not change in a frame. In the simulation, $n$ communication pairs are randomly selected. Assume each selected link has the same traffic rate which is normalized to be 1.

B. Centralized algorithm

We first compare the throughput performance of our proposed algorithm to that of several other centralized algorithms. The algorithm in [9] is based on protocol interference model and a polynomial-time coloring algorithm is used. To express conveniently, we call it Algorithm Protocol Model (PM). In [15], the algorithm is based on physical interference model, which is the same as ours. It is a heuristic scheduling method that is based on a greedy assignment of weighted links. We call it Algorithm Physical Greedy (PG). The authors in [16] construct the conflict graph using the physical interference model, which does not consider the accumulation effect either. It is called Algorithm Physical Conflict Graph (PCG). The throughput comparison of these algorithms is shown in Fig.2, where the results are average throughput over 10000 simulation runs.

It shows that our proposed APP algorithm outperforms the protocol interference model based algorithm (PM algorithm) and other physical interference model based heuristic algorithms (PG and PCG algorithms). It performs even better when more nodes begin to transmit. The PM algorithm performs the worst, because it uses the protocol interference model, which is less precise than the physical interference model.

Then we evaluate the influence of the parameter $\Delta A/\bar{A}$ (defined in Theorem 1) to the $(1 - \theta)$-approximate lower bound. Here we denote $\Delta A/\bar{A}$ by $\delta$ in the legends of Fig. 3(a), 3(b) and 3(c) for simplicity. It can be seen from Fig.3(a) that the lower bound decreases while $\delta$ decreases, which is from the fact that more adjustments are made in the phase 2 of the Algorithm 1. In our simulation, $\delta$ is rarely smaller than -0.3, implying that the approximation ratio is larger than 0.5 with probability larger than 0.7, and the approximation ratio larger than 0.4 with probability larger than 0.9.

When more links begin to transmit, more adjustments are made in Algorithm 1, i.e., $\Delta \bar{A}$ decrease. The influence of the network scale $n$ to the performance can be seen in Fig.3(b) and 3(c). We can get the following conclusions when $n$ increases:

1) If $\delta$ doesn’t decrease, the probability lower bound increases. In Fig.3(b), $\delta$ is fixed to be 0.1. The probability lower bound increases while more links begins to transmit. If $\delta$ increases, the probability lower bound also increases (see plot $(n = 15, \delta = -0.3)$ and $(n = 20, \delta = 0)$ in Fig.3(c)).

2) If $\delta$ decrease, the probability lower bound decreases. It can be seen in Fig.3(c) (see plot $(n = 10, \delta = -0.1)$ and $(n = 20, \delta = 0)$).

In section III, Theorem 1 gives the probability lower bound of getting the guaranteed approximation ratio for the cen-
tralized algorithm. Fig. 3(d) presents the theoretical lower bound computed by Theorem 1, and compares it with the experimental result by using our centralized algorithm. Similar results are generated when different number of communication links are chosen, and we take $n = 10$ for example in Fig. 3(d). It shows that the centralized algorithm has a good approximation ratio lower bound. For example, the approximation ratio is larger than 0.5 with probability larger than 0.7, and the approximation ratio larger than 0.4 with probability larger than 0.9. It also shows that the probability of getting the guaranteed approximation ratio computed by our algorithm is larger than the theoretical lower bound when the approximation ratio $1 - \theta$ is large. When the approximation ratio $1 - \theta$ is small, the experimental probability equals 1, which is the same to the theoretical lower bound.

C. Distributed algorithm

The simulation configuration is the same as the case of the centralized algorithm. In Fig. 4(a), the throughput of the centralized and the distributed algorithms are compared. It shows that the distributed algorithm approaches the centralized algorithm well when less communication pairs are selected. Note that the distributed algorithm outperforms the centralized algorithms PM in Fig. 2. It reflects the importance to use the physical interference model to design scheduling mechanism. It also outperforms algorithm PG in Fig. 2, because our distributed algorithm finds the optimal solution of the LP-relaxation, and the performance can be guaranteed.

It also shows that there exists certain performance gap between the two algorithms. The reason is that we optimize the throughput on per basis in the distributed approach. [23] shows that the performance gap between the single slot optimization manner and the frame optimization manner is $O(\log n)$. Furthermore, we relaxed several constraints of the ILP problem in the distributed algorithm as described in section IV. If one node has to communicate with several other nodes, it will select one randomly. The random selection procedure is another factor that influences the performance.

Fig. 4(b) presents the convergence property of the distributed approach. Here we do not take the overheads into account which are caused by the probing phases of the P-slots, and only consider the convergence property of the main algorithm. Since similar results can be got when $n$ is assigned different values, with $n = 10$ selected as an example. The notation $OPT$ stands for the optimal solution of the relaxed ILP problem (12), and $Distributed\ algorithm$ stands for the distributed approach proposed in the paper. It can be seen that the throughput of the distributed approach will converge to the optimal one quickly. Because the iteration procedure is used to compute the transmission probability of each node, the solution does not need to converge to the optimal one with complete accuracy. Here, we set certain accuracy error for the transmission probability, which does not influence the throughput much and speed up the convergence procedure. The throughput gap between the optimal and our distributed algorithm is about 2% in this example.

Taking into account the overheads which are caused by the probing phases of the P-slots, the influence of the number of the P-slot to the throughput can be seen in Fig. 4(c). It shows clearly that the throughput increases when $m$ increases. This is because the algorithm converges to optimality quickly, and a better throughput is achieved. But when $m$ grows larger than a certain value, the throughput begins to drop, due to the increasing overheads caused by P-slots. The number of the P-slot needs to be carefully configured according to different networks environment.

VI. CONCLUSION

In this paper we study the link scheduling problem with the physical interference model, which is more accurate than the protocol interference model but notoriously hard to handle with. By relaxing the formulated ILP problem to be a LP problem, we propose a polynomial time centralized algorithm to solve the scheduling problem. The probability bound is achieved to get the guaranteed approximation ratio. In the distributed scenario, the formulation of the scheduling problem is simplified to solve the optimal transmission probability, which can be implemented distributively by each node through Jacobi algorithm using local channel information. Simulation results show that our proposed centralized algorithm outperforms several other scheduling method and the performance bound can be achieved. The performance of the distributed algorithm approaches the centralized algorithm and can converge fast to the optimal solution.

REFERENCES


Copyright (c) 2011 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.
Appendix A

Proof of Theorem 1

Let $x_{ij}^*$ be the solution of the ILP problem (equation (1) - (7)) using Algorithm 1, and the throughput of Algorithm 1 be $A$. Denote by $A_{OPT}$ the optimal solution of the ILP problem. It is also assumed that the optimal solution of the LP-relaxation problem (Equation (1) - (6) and (8)) is $\hat{A}$, and the throughput of the LP-relaxation is $\hat{A}$. It follows that $A = \sum_{t=1}^{T} \epsilon_{ij} x_{ij}^*.$

Let $A_{RAND}$ be total throughput calculated by the randomized rounding procedure in Phase 1, i.e., $A_{RAND} = \sum_{t=1}^{T} \epsilon_{ij} x_{ij}.$ Obviously, it shows that $A_{RAND} \leq A_{OPT} \leq \hat{A}$.

From equation (9), we can get $E(x_{ij}^*) = x_{ij}^*$. Thus,

$$E(A_{RAND}) = E\left(\sum_{t=1}^{T} \sum_{\epsilon_{ij} \in E} b_{ij} x_{ij}^*\right) = \sum_{t=1}^{T} \sum_{\epsilon_{ij} \in E} E(b_{ij} x_{ij}^*) = \hat{A}$$

According to equation (29) and Chernoff Bound [36], we can get the following bound,

$$Pr(A_{RAND} \geq (1-\delta)\hat{A}) = 1 - Pr(A_{RAND} < (1-\delta)\hat{A}) \geq 1 - e^{-\frac{\delta^2 \hat{A}}{2}}$$

Let $A$ be the final solution of Algorithm 1. In Phase 2, the randomized rounding solution is checked whether it satisfies all the constraints in the ILP problem, and some adjustments
are made, leading that the throughput changes. Suppose the throughput changes by $\Delta A$ after Phase 2, and it follows that
\[
A = A_{\text{rand}} + \Delta A
\]  
(31)
Substituting equation (31) into (30), we can derive
\[
Pr (A \geq (1 - \delta + \frac{\Delta A}{A})\hat{A}) \geq 1 - e^{-\frac{(\theta + \frac{\Delta A}{A})^2}{2}},
\]
(32)
where $-1 + \frac{\Delta A}{A} \leq \theta \leq 1 + \frac{\Delta A}{A}$.

In Phase 2, the worst case is that all links are adjusted not to transmit, leading the throughput $A = 0$, i.e. $A_{\text{rand}} + \Delta A = 0$; while the best case is that after the adjustment, the solution approaches to the optimal, leading $A = A_{\text{rand}} + \Delta A \leq \hat{A}$. Thus we can get the variation range of $\Delta A$ as follows,
\[
-A_{\text{rand}} \leq \Delta A \leq \hat{A} - A_{\text{rand}}
\]
(33)
Let $\theta = \delta - \frac{\Delta A}{A}$. Equation (32) can be written as follows,
\[
Pr (A \geq (1 - \theta)\hat{A}) \geq 1 - e^{-\frac{(\theta + \frac{\Delta A}{A})^2}{2}},
\]
(34)
where $-1 + \frac{\Delta A}{A} \leq \theta \leq 1 + \frac{\Delta A}{A}$.

Because $(0, 1) \subset [-1 + \frac{\Delta A}{A}, 1 + \frac{\Delta A}{A}]$, according to equation (28)(34), we can get
\[
Pr (A \geq (1 - \theta)A_{\text{opt}}) \geq Pr (A \geq (1 - \theta)\hat{A}) \geq 1 - e^{-\frac{(\theta + \frac{\Delta A}{A})^2}{2}}
\]
(35)
where $\theta \in (0, 1)$.

APPENDIX B
PROOF OF LEMMA 1

As described in Section III, $\Delta$ is a sufficiently large value. Here we assume it is greater than the total aggregate interference produced in the whole network, i.e. $\Delta > \beta (N + \sum_{c_{ij} \in E} PG_{ij})$. So it can be deduced that $\Delta - \beta PG_{ii} > \beta (N + \sum_{c_{ij} \in E \setminus \{c_{ii}\}} PG_{ij})$. Because $\beta$ is usually bigger than 1, it follows that $\Delta - PG_{ii} > \beta (N + \sum_{c_{ij} \in E \setminus \{c_{ii}\}} PG_{ij})$. Based on Definition 1 it can be seen that matrix $A_i$ is a strictly row diagonally dominant matrix. According to Levy-Desplanques theorem [40], matrix $A_i$ is non-singular and invertible.

APPENDIX C
PROOF OF LEMMA 2

This lemma can be proven by mathematical induction. Assume $n = 1$, then $A_1 = [a_{11}]$. It is obvious that $det(A_1) = a_{11} > 0$, and lemma 2 follows. Suppose lemma 2 follows when $n = k - 1 (2 \leq k \leq n)$, i.e. $det(A_{k-1}) > 0$. When $n = k$, denote the submatrix of $A_k$ with row $j$ and column $i$ deleted by $A_{kj}$. It can be got that
\[
det(A_k) = \sum_{j=1}^{k} a_{kj}(-1)^{j+k} det(A_{kj})
\]
\[
\geq a_{kk}(-1)^{k+k} det(A_{kk}) = a_{kk} det(A_{k-1})
\]
Since $a_{kk} > 0$ and $det(A_{k-1}) > 0$, we can get $det(A_k) > 0$, and lemma 2 follows.

APPENDIX D
PROOF OF LEMMA 3

According to the definition of $g_n(x)$, we get $x = A_n^{-1}c$. Since matrix $A_n$ is a strictly column diagonally dominant matrix, $A_n^{-1}$ is a strictly row diagonally dominant matrix [37]. Then according to Definition 1, for all $i = 1, \ldots, n$, it follows that $|\alpha_{ii}| > \sum_{j \neq i} |\alpha_{ij}|$. Because $\alpha_{ii} > 0$, we can get $\sum_{j=1}^{n} \alpha_{ij} > 0$. Therefore, it can be deduced that
\[
x_i = \sum_{j=1}^{n} \alpha_{ij}(\Delta - \beta N) > 0, \forall x_i \in \mathbb{R}.
\]

APPENDIX E
PROOF OF THEOREM 3

Without loss of generality, we suppose that after a few recursion operations, there are $i$ variables of $\bar{x}$ not satisfying condition (19), denoted by $\bar{x}_i$. They are restricted to 1, and denoted by $x_i^* = [x_1, \ldots, x_i]^T$. The constraint matrix becomes $A_{n-i}$, the vector form becomes $g_{n-i}(x)$ and the rate vector becomes $b_{n-i}$. It is also supposed that the solution of $g_{n-i}(x) = 0$ is $x_{n-i}^*$, and $x_{n-i}^*$ satisfies condition (19) (if condition (19) is not met, the recursion will continue; the extreme case is that all the variables of $\bar{x}$ do not meet the condition). Next we prove that there exists a dual optimal point $(\omega^*, \lambda^*, \nu^*)$ which guarantees that $x^* = [x_i^*, x_{n-i}^*]$ is optimal.

• Case $x_{n-i}^*$

Since the constraint matrix $A_{n-i}$ is a submatrix of $A_n$, it is also a strictly column diagonally dominant matrix. According to the proof of theorem 2, it can be deduced that there exists a point $(\omega_{n-i}^*, \lambda_{n-i}^*, \nu_{n-i}^*)$ which guarantees that $x_{n-i}^*$ is optimal to $g_{n-i}(x) = 0$, and satisfy KKT conditions (15)~(21). Specially, we can get the following equation from condition (21):
\[
A_{n-i} \omega_{n-i} = b_{n-i},
\]
(36)
which follows because $\lambda_{n-i}^* = \nu_{n-i}^* = 0$ (see the proof of theorem 2).

• Case $x_i^*$

Let $\omega_i^* = \nu_i^* = 0$, it can be deduced easily that $\forall x_j^* \in x_i^*$, thus KKT condition (16)~(20) are satisfied. Next we prove that KKT condition (15) is also satisfied.

Since $g_n(x) = A_n x - c$, $\forall x_j^* \in x_i^*$, we can get the following equation:
\[
g_j(x_i^*) = \sum_{k=1}^{n} a_{kj}x_i^* - c_j.
\]
(37)
It can be easily seen from equation (37) that $g_j(x)$ is an increasing function of $x$. So $g_j(x_j^*) < g_j(x_j) = 0$, and $\forall x_j^* \in x_i^*$ satisfies KKT condition (15). We next prove that there exists $\lambda^* \geq 0$ that meets KKT condition (21).
Denote the $j$th row of $A_n$ by $A_n^j$. Suppose $x^*_{j} \in x^*_{k-i}(i + 1 \leq j \leq n)$, and we can get

$$A_n^j \omega = \sum_{m=1}^{n} a_{mj} \omega_m$$

$$= \sum_{m=i+1}^{n} a_{mj} \omega_m$$

$$= a_{jj} \omega_j + \sum_{m=i+1, m \neq j}^{n} a_{mj} \omega_m$$

$$= A_{n-i} \omega_{n-i}$$

$$= b,$$

in which equation (38) follows because $\omega_i = [\omega_1, \ldots, \omega_i]^T = 0$, and (40) follows because of (36). Similarly, suppose $x^*_k \in x^*_i (1 \leq k \leq i)$, and it follows that

$$A_n^k \omega = \sum_{m=1}^{n} a_{mk} \omega_m$$

$$= \sum_{m=i+1}^{n} a_{mk} \omega_m.$$  \hspace{1cm} (41)

By comparing equation (39) and (41), the difference is that equation (39) has an item $a_{jj} \omega_j$. Because $a_{jj}$ is the diagonal entry of matrix $A_n$ which is strictly row diagonally dominant, it can be easily deduced that $A_n^j \omega < A_n^j \omega = b$. So there exist $\lambda^*_k \geq 0$ that makes $A_n^k \omega = b - \lambda^*_k$, i.e., condition (21) is met.

In a word, there exist Lagrange multipliers $\omega^* = [\omega^*_1, \omega^*_n-i]^T$, $\lambda^* = [\lambda^*_1, \lambda^*_n-i]^T$ and $\nu^* = [\nu^*_1, \nu^*_n-i]^T$ which make $x^*$ satisfies all the KKT conditions, and theorem 3 follows.