Exact and Approximate Link Scheduling Algorithms under the Physical Interference Model

Qiang-Sheng Hua
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong, P.R. China
qshua@cs.hku.hk

Francis C.M. Lau
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong, P.R. China
fcmlau@cs.hku.hk

ABSTRACT

Given n arbitrarily distributed single-hop wireless links, using the physical interference model, the objective is to minimize the scheduling length. This is an open problem (Problem 1) proposed by Locher et al. [21]. In this paper, we solve this open problem at the cost of moderately exponential time. Specifically, this paper gives two classes of exact and approximate link scheduling algorithms, one based on the somewhat straightforward link independent set covering, and the other on counting the number of set covers. Let $p(n)$ denote the time of checking whether the spectral radius of an irreducible non-negative matrix is smaller than 1 or not, then the time complexity for the set covering based exact algorithm is $O(2^{\binom{n}{n/2}})$, whereas the proposed counting based exact scheduling algorithm called ESA_MLSAT needs only time $O(3^n \cdot n \cdot \log^2 n \cdot p(n))$ with polynomial space. If exponential space is allowed, the time complexity can be further reduced to $O(2^n \cdot n \cdot \log^2 n \cdot p(n))$. The time complexity for the set covering based approximate algorithm is $O((\binom{n}{n/2}) \cdot \log n \cdot p(n))$ with approximation ratio $O(\log n)$. The time complexity of the first counting based approximation algorithm is $O(n^2 poly \log(n))$ with approximation ratio $O(n/\log n)$, the time complexity of the second counting based approximation algorithm is $O(n^{1+\log 3 \cdot \log^{k-1} n} poly \log(n))$ with approximation ratio $O(n/\log^k n)$, and the time complexity of the third counting based approximate algorithm is $O((\binom{n}{n/2}) + 3^{e^{-\varepsilon} n} \cdot n \cdot \log n) \cdot \log n \cdot p(n))$ with approximation ratio $\lceil (1+\varepsilon) \rceil$. All these approximation algorithms use polynomial space.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *wireless communication, network topology*;
F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems - *geometrical problems and computations, sequencing and scheduling*;
G.2.1 [**Discrete Mathematics**]: Combinatorics - *Combinatorial algorithms, counting problems*;
G.2.2 [**Discrete Mathematics**]: Graph Theory – *Network problems*;

General Terms

Algorithms, Theory

Keywords

Wireless, Ad hoc Networks, Sensor Networks, Link Scheduling, Coloring, Counting, Physical Interference Model, SINR, Inclusion-Exclusion Principle, Constraint Satisfaction Problem.

1. INTRODUCTION

Recently, wireless link scheduling with or without power control under the physical interference model has attracted considerable attention [2,6,7,9-11,15,16,18,19,21,23-27,29]. The physical interference model is preferred to the graph based models for two reasons: (1) graph based models tend to yield conservative performance [18], and it has been shown that scheduling algorithms based on the physical model can surpass the theoretically achievable performance of the graph based models [25]; (2) the aggregate interference effect of simultaneous transmissions may subvert a communication request which might otherwise appear successful under some graph based models. In this paper, we study the minimum length wireless link scheduling problem for arbitrary topologies under the physical interference model [15,19,21,23,27].

A wireless link comprises a transmitter and a receiver, and the problem is to schedule all such links (such that each transmitter can successfully send a packet to its corresponding receiver) with the fewest timeslots. By arbitrary topologies, we mean all the transmitters and receivers are arbitrarily located on a plane, which implies arbitrary link lengths and link densities. This is an open problem (Problem 1) recently proposed by Locher et al. in [21]. To the best of our knowledge, this present paper gives the first exact and the first non-trivial approximate link scheduling algorithms for this open problem. In addition, the power levels we assign to the transmitters are optimal in the sense that each transmitter uses the minimum power to transmit while guaranteeing that the receiver can successfully decode the packet. The main technique we use is the inclusion-exclusion principle which has many surprising applications in set partitioning, including graph coloring [4,5,20]. Although our problem can not be directly transformed to a vertex coloring problem, which is to find the minimum number of colors such that no edges connect two identically colored vertices, we can view the feasible concurrent transmissions as a link independent set, thus building a connection between our minimum length link scheduling problem

and the independent set based vertex coloring problem. To avoid confusion, we refer to these feasible simultaneous transmissions as a "**link independent set**", and those non-adjacent vertices in a graph an "**independent set**".

We propose two classes of exact and approximate link scheduling algorithms, one based on the relatively straightforward set covering, and the other on counting the number of different set covers. Throughout the paper, we let $p(n)$ denote the time of checking whether the spectral radius of an irreducible non-negative matrix is smaller than 1 or not; then the time complexity for the counting based exact link scheduling algorithm called ESA_MLSAT is $O(3^n \cdot n \cdot \log^2 n \cdot p(n))$ with polynomial space, which represents a substantial improvement over the set covering based exact scheduling which needs time $O(2^{\binom{n}{n/2}})$. If exponential space is allowed, using either the fast zeta transform [5] or the fast subset convolution [3], the time complexity can be reduced to $O(2^n \cdot n \cdot \log^2 n \cdot p(n))$. Then based on the exact coloring and the maximum link independent set finding algorithms, we present three approximate link scheduling algorithms with approximation ratios $O(n/\log n)$, $O(n/\log^k n)$ and $\lceil (1+\varepsilon) \rceil$, respectively. Here ε is an arbitrary positive value independent of n. The time complexity of the first approximation algorithm is $O(n^2 poly\log(n))$ with polynomial space, the time complexity for the second algorithm is $O(n^{1+\log 3 \cdot \log^{k-1} n} poly\log(n))$ with polynomial space, and the time complexity for the third algorithm is $O((\binom{n}{n/2} + 3^{e^{-\varepsilon}n} \cdot n \cdot \log n) \cdot \log n \cdot p(n))$ with polynomial space.

The remainder of the paper is organized as follows. We give the system model, the problem formulation and hardness analysis in Section 2. In Section 3, we discuss some closely related work that focus on joint link scheduling and power control under the physical interference model. Then we present some exact and approximate link scheduling algorithms based on link independent set covering in Section 4. In Section 5, we give the construction of the pair-wise link conflict (infeasible) graph, on top of which we build the exact link scheduling algorithm. In Section 6, based on the inclusion-exclusion principle, we give the exact coloring algorithm ESA_MLSAT through counting the number of *k*-set coverings. Building upon these results, we present three approximate link scheduling algorithms in Section 7. We conclude the paper in Section 8. Note that we will use the terms scheduling and coloring interchangeably throughout the paper.

2. SYSTEM MODEL AND PROBLEM DEFINITION

2.1 System Model

A link is a wireless transmission from a transmitter to a receiver. All the transmitters and receivers are arbitrarily located on a plane, and they are equipped with omni-directional antennas. For convenience, we assume the transmitters and receivers of all the links are distinct, but they can be arbitrarily close to each other. This constraint can be eliminated if we can use multi-radio and full-duplex wireless devices. But if only single-radio and half-

duplex devices are allowed, we can just add more constraints, such as a device can not transmit and receive at the same time and a transmitter can not transmit to (receive from) more than one receiver (transmitter) at the same time, to the following **link independent set** definition (Definition 2.1 in Section 2.2). Since these constraints can be checked in polynomial time, it would not affect our results. The transmitters can vary their transmission powers from zero to infinity. All the transmissions will occupy the entire bandwidth, and thus simultaneous transmissions may interfere with each other. A successful transmission is determined based on the physical interference model, i.e., the signal-to-interference-and-noise-ratio (SINR) model proposed in [16]. Only when the SINR ratio at the link receiver is above the SINR threshold can the packet be successfully transmitted. The SINR ratio at the receiver of a link i can be represented as:

$$SINR_i = \frac{g_{ii} \cdot p_i}{n_i + \sum_{j=1, j\neq i}^{Q} g_{ij} \cdot p_j} \geq \beta$$

where p_i denotes the transmission power of link i's transmitter i_s; n_i is the background noise at link i's receiver i_r; g_{ii} and g_{ij} are the link gain from i_s to i_r, and that from link j's transmitter j_s to i_r, respectively; Q denotes the number of simultaneous transmissions with link i; β is the SINR threshold which is larger than or equal to 1.

If we do not consider the fast fading and obstacles effects of wireless transmissions, the link gain can be represented by an inverse power law model of the link length, i.e., $g_{ii} = 1/d^\alpha(i_s, i_r)$ and $g_{ij} = 1/d^\alpha(j_s, i_r)$. Here $d(,)$ is the Euclidean distance function, and α is the path loss exponent, which is equal to 2 in free space, and varies between 2 and 6 in urban areas.

We then define a normalized non-negative $Q \times Q$ link gain matrix $H = (h_{ij})$ such that $h_{ij} = \beta \cdot g_{ij}/g_{ii}$, for $i \neq j$, and $h_{ij} = 0$, for $i = j$. We also define a normalized noise vector $\eta = (\eta_i)$ such that $\eta_i = \beta \cdot n_i/g_{ii}$. From this we can rewrite the SINR inequality as

$$p_i \geq \sum_{j=1}^{Q} h_{ij} \cdot p_j + \eta_i$$

Using a vector-matrix notation, the above inequality becomes $P \geq HP + \eta$, or $(I - H)P \geq \eta$. Here, $\geq$ is an element-wise operation.

Now according to the Perron-Frobenius Theorem [1,28,33], $(I - H)^{-1} > 0$ if and only if the spectral radius (the largest absolute value of all eigenvalues) of the H matrix $\rho(H) < 1$ (We use $p(n)$ to denote the time for checking this property, where n is the number of the links in the H matrix), and we say the power vector $P^* = (I - H)^{-1} \cdot \eta$ is optimal in the sense that each transmission power in this P^* vector is the minimum power that must be used to satisfy all the Q simultaneous transmissions. In other words, in order to successfully schedule all these Q links, each transmitter must use power larger than or equal to the corresponding power in P^*.

2.2 Problem Definition

We first give the original problem description below. (Note that this problem has also been studied in [15,23,27], and we will review their results in Section 3.)

Problem 1 (Locher, von Rickenbach, Wattenhofer [21]). *A communication request consists of a source and a destination, which are arbitrary points in the Euclidean plane. Given n communication requests, assign a color (time slot) to each request. For all requests sharing the same color specify power levels such that each request can be handled correctly, i.e., the SINR condition is met at all destinations. The goal is to minimize the number of colors.*

In this paper, since we will formulate this problem as a set covering problem, we need to introduce some related concepts.

DEFINITION 2.1: A set of concurrent transmissions are called a **link independent set** if there exist a positive power vector P^* (cf. Section 2.1) satisfying all the SINR constraints; otherwise it is an **infeasible link independent set**.

DEFINITION 2.2: A **maximal link independent set** is a link independent set that is not a proper subset of any other link independent set.

DEFINITION 2.3: The largest maximal link independent set is called a **maximum link independent set**.

PROPOSITION 2.4: Any superset of an infeasible link independent set is an infeasible link independent set; each subset of a link independent set is a link independent set.

Based on these definitions, we can rewrite Problem 1 as the <u>M</u>inimum-<u>L</u>ength Link <u>S</u>cheduling for <u>A</u>rbitrary <u>T</u>opologies (**MLSAT**) problem which is defined below.

The MLSAT Problem: Given n arbitrarily distributed single-hop wireless links $N = \{1..n\}$, select a minimum number of link independent sets such that each link has at least one successful transmission under the SINR constraint.

Remark 1: This problem implies that every link has uniform demand, i.e., each link has the same number of packets to transmit. The non-uniform link demands version of the **MLSAT** problem, i.e., selecting a minimum number of link independent sets such that all the packets will be successfully transmitted is our future task.

2.3 Hardness of the MLSAT Problem

With the assumption of uniform power assignments, i.e., all the links within a link independent set employ the same power, the decision version of the MLSAT problem has been proved to be NP-complete through a reduction from the partitioning problem [15]. By assuming a zero SINR threshold and all the links use zero power, the authors in [2] also prove the MLSAT problem (they call it ILSP problem) to be NP-complete through a reduction from the edge-coloring problem. For the general case (using power control and positive SINR thresholds), however, whether the MLSAT problem remains NP-complete or not is still an open problem [15]. But one thing for sure is that, compared with the uniform (constant) power assignment and linear power assignment [18,19,26], using power control, such as the optimum power assignment used in this paper and the non-linear power assignment used in [18,19,23,24,26,27], the scheduling length can

be greatly reduced. Moreover, as discussed in [19], the authors have shown that the scheduling length can be further reduced at the cost of using more power as long as we have employed the appropriate power assignment strategy. So to what extent the impact of power control would affect the NP-completeness nature of the uniform power assignment case is the key to the hardness of the problem. Intuitively, we may conjecture that the MLSAT problem remains NP-hard in the general case.

3. RELATED WORK

We must first of all distinguish our work from another two categories of link scheduling problems. In the first category of scheduling problems, fractional coloring (or multi-coloring) is allowed [17], which means that a single packet can be partitioned into certain smaller units for transmission. Hence, the minimum length scheduling problem can be formulated as a linear programming problem, and by Khachiyan's ellipsoid algorithm, it can be solved in polynomial time. In this paper, we assume the packet is the smallest unit, so it is an integer programming, and it may not admit a polynomial time algorithm. The second category of wireless link scheduling problems is studied in the context of queuing networks. In this case, the optimal throughput scheduling is equivalent to finding the maximum weighted matching of the links under certain interference models. This problem is polynomial under the 1-hop interference model and becomes NP-hard under the K-hop interference models (K>1) [30]. Here the weight of the links is the maximum backlog difference between the link's sender and receiver. And the backlog value of a wireless device means the number of packets that are waiting to transmit. Obviously, this problem can not be interpreted as our minimum length scheduling problem. In our case, we assume the links have fixed uniform demands, and therefore minimum length scheduling also means optimal throughput capacity.

The first joint link scheduling and power control algorithm under the SINR model was proposed in [11]. Their objective is to improve the single-hop throughput and to reduce the total power consumption. The algorithm is implemented in two phases: in the scheduling phase, they try to find a valid transmission scenario such that each link has sufficient spatial separation with other links based on a parameter D; in the second power control phase, they try to find the maximum number of admissible links from the valid transmissions such that each link would satisfy the SINR constraint while minimizing the total power consumption. This is a heuristic algorithm without performance guarantee. Also, they did not show how to adjust the value of parameter D which is a key parameter affecting the performance of this joint scheduling and power control algorithm.

Another closely related work is the integrated minimum frame length link scheduling and power control problem proposed in [2]. Based on the power based interference graph, the authors try to find a maximal link independent set using the Minimum Degree Greedy Algorithm (MDGA). This algorithm is also of heuristic nature, and may not work well on arbitrary topologies. Moreover, this algorithm may lead to $\Omega(n)$ lower bounds in the worst case since they use the step-wise maximum interference removal algorithm (SMIRA) in the pruning stage [23]. Although they

added the "maximality stage" which tries to find a feasible superset of the link independent set, the performance improvement is still unclear. For a special case where the links have superincreasing demands, i.e., when we sort the link demands in a non-increasing order, each link with a higher demand is greater than or equal to the sum of all the links with lower demands, the authors in [6] design a greedy algorithm which can construct a shortest schedule in polynomial time.

If the wireless links are built on uniformly and randomly distributed wireless devices (a special case of our problem), a polynomial time approximate link scheduling algorithm called GreedyPhysical with a sub-linear approximation ratio for non-uniform link demands is given in [7]. By taking advantage of the desirable properties of random networks, the authors propose an approximate physical interference model in which the interferences caused by the links outside some "close-in" region of the scheduled link (called "far-away" region) can be neglected, which greatly relaxes the global constraint property of the original physical interference model. This kind of approximate physical interference model, however, does not hold in our case here of arbitrary link topologies (the general case).

If the links topology is a kind of spanning tree which is constructed by iteratively connecting the nearest neighbor forests of the wireless devices on a plane, an ingenious non-linear power assignment based scheduling algorithm is proposed in [26]. By using some elegant geometrical arguments, they prove that the scheduling length is upper bounded by $O(\log^4 n)$. This is also the first paper to point out that both constant (or called uniform) and linear power assignments are inefficient in the worst case link topologies. These results have been generalized to cover wide-band networks in [18], and the authors also show that the total power consumption of the non-linear power assignments is lower bounded by $\Omega(n \cdot 2^n)$. Based on this observation, a scheduling algorithm which is a tradeoff of the scheduling and energy complexities is presented in [19], in which the authors give the first $o(n)$ scheduling length with only polynomial total power consumption. In addition, this paper also discusses the interesting "More Energy, More Power?" problem, i.e., whether we can schedule the wireless links in fewer timeslots at the cost of using more energy. An affirmative answer to this question is given but with the precondition that the right power assignment strategy is employed [19]. In a line of improvements, the scheduling length for the spanning tree topology has been reduced to $O(\log^3 n)$ in [27], then to $O(\log^2 n)$ in [24].

For arbitrary link topologies, two non-linear power assignments based link scheduling algorithms have been given in [27] and [23]. The authors in [27] prove that the scheduling length of arbitrary link topologies is upper bounded by $O(I_{in} \cdot \log^2 n)$, where I_{in} is a graph-based in-interference measure. The authors in [23] prove that the scheduling length of arbitrary link topologies is upper bounded by $O(\chi_\rho \rho^2 \log n \cdot (\log n + \rho))$, where χ_ρ is the maximum number of senders (receivers) located in the circle centered at the link's sender (receiver) with a radius as the corresponding link's length over a ρ value. These are interesting and quite useful results with an exception for the higher density topologies, i.e., the in-interference I_{in} or the χ_ρ value becomes

sufficiently large. But more importantly, these two papers build interesting bridges between the graph based models and the physical interference model. For example, the result in [27] shows that the topology control method with the aim of lowering the maximum in-interference helps to shorten the scheduling length.

With the assumption of uniform power assignments (aka constant power assignments), an approximate link scheduling algorithm is proposed in [15], which can schedule all the links whose lengths are within a factor of 2 in a constant approximation ratio over the optimum. In practice, since the number of the magnitudes of link lengths is almost constant, the approximation ratio remains constant, but in the worst case the approximation ratio becomes $O(n)$. In the context of more complicated multi-hop wireless connections, some joint link scheduling and routing problems with SINR constraints have been proposed in [9,10]. And in [29], also using the physical interference model, the authors present a randomized and distributed protocol called TWIN which can establish a constant density dominating set in $O(\log n)$ communication rounds with high probability.

4. SET COVERING BASED EXACT AND APPROXIMATE COLORINGS

From the formulation of the MLSAT problem we can see that it can be viewed as a kind of set covering problem. So in this section, we give some relatively straightforward exact and approximate link scheduling algorithms based on some traditional techniques such as the generation and test method, the backtracking search and the greedy set covering. Compared to the counting based exact and approximate link scheduling algorithms given in Section 6, we will see that these traditional methods are inferior in terms of either the running time or the approximation ratio.

LEMMA 4.1: The number of maximal link independent sets in arbitrary link topologies is at most $\binom{n}{n/2}$.

PROOF: In graph theory, it is known that the number of maximal independent sets in a graph is at most $3^{n/3}$ [22]. In our scenario, due to the cumulative interference effect of the SINR model, the number of maximal link independent sets may exceed $3^{n/3}$, since in graph theory, if all the subsets of a set of vertices are independent sets, then this set is also an independent set. But this is not true in our setting. According to Proposition 2.4, we can easily obtain the result.

4.1 Set Covering based Exact Coloring

Since there are at most $O(2^n)$ link independent sets, a naive brute force optimal covering (such as the generation and test method) takes time $O(2^{2^n})$. An improvement is to consider only the maximal link independent sets, but some post processing is needed since some links may then be scheduled more than needed. All the maximal link independent sets can be found in $O(2^n \cdot p(n))$, and from Lemma 4.1, the optimal set covering takes time $O(2^{\binom{n}{n/2}})$.

4.2 Set Covering Based Approximate Coloring

This approximation algorithm proceeds as follows: In each timeslot, we find a maximum link independent set among the unscheduled wireless links; then we delete the maximum link independent set and continue until all the links have been scheduled. Actually, this is equivalent to the standard greedy set covering algorithm which is to select a set to maximize the uncovered elements, and the approximation ratio is $O(\log n)$. The decision version of the maximum link independent set finding problem in arbitrary link topologies has been shown to be NP-complete in [1], and an obvious brute force algorithm takes time $O(2^n \cdot p(n))$. For example, we can just enumerate all the k-combinations (k is from n to 1) of the n links, and then check whether they are link independent sets. If yes, we just stop there and output the k links [33]. But according to Proposition 2.4, and with the help of binary search, we can give an exact maximum link independent set finding algorithm which takes time $O(\binom{n}{n/2} \cdot \log n \cdot p(n))$. From the Stirling's approximation for large factorials, the above complexity becomes $O(2^n \cdot \log n / \sqrt{n} \cdot p(n))$. This algorithm works as follows: we first check whether there exists a link independent set in all the $n/2$-combinations of the n links; if yes, we check the $3n/4$-combinations, otherwise we check $n/4$-combinations. This continues until we find the maximum combination. In Section 7.3, we will give another exponential time approximation algorithm with a much better approximation ratio and without increasing the running time.

Remark 2: Although there has been a long history of finding the maximum independent set in a graph with faster exponential time, such as $O(1.2209^n)$ using a measure and conquer approach in [13], we suspect that similar techniques can not be applied in our setting, since besides the observation in Proposition 2.4, we can not take advantage of many useful structural properties of graphs without checking all the subsets that are upper bounded by $\binom{n}{n/2}$ in each searching step. Further study of more efficient maximum link independent set finding algorithms will be of independent interest. Also we note that the well-known divide and conquer based "grid shifting strategy" [12] which is to find a PTAS for the maximum independent set under the disk graph interference model can not be applied in our setting.

5. PAIR-WISE CONFLICT GRAPH

In this section, we give the construction of the pair-wise link conflict graph (or infeasible graph) called G_{pair}. Our exact scheduling algorithm is built upon it.

The conflict graph is constructed based on the following fact.

FACT 5.1 [18]: Given two links i and j, if $g_{ii} \cdot g_{jj} \le \beta^2 \cdot g_{ij} \cdot g_{ji}$, then these two links can not form a link independent set, and we call them conflict (or infeasible) links; otherwise, there always exists a positive power assignment to allow them to be scheduled simultaneously.

From this fact, we can build G_{pair} by just adding an edge between any two infeasible links. This is done in time $O(n^2)$. And for each link i, let $N(i)$ denote the number of links which conflict with i, i.e., the number of neighbors of node i in G_{pair}. Then due to Proposition 2.4, we have the following straightforward theorem.

THEOREM 5.2: The number of colors for the **MLSAT** problem is lower bounded by the clique number $\omega(G_{pair})$.

This trivial lower bound can be improved by applying similar constraint propagation techniques in [8].

Remark 3: Instead of just creating a pair-wise conflict graph, according to the Perron-Frobenius Theorem introduced in Section 2.1, one can also build other k-conflict graphs for each k links, where k is up to a constant value. But this construction does not help too much in our scheduling problem. There are at least four reasons. First, the complexity of computing these k-conflict graphs grows exponentially with the number k. Second, by viewing the scheduling algorithm as a graph coloring problem, we can not guarantee a correct scheduling under the physical interference model without constructing $O(2^n)$ number of conflict graphs. Third, all these k-conflict graphs can become trivial graphs (isolated vertices) no matter what the value of k is. Fourth, our exact link scheduling algorithm (**ESA_MLSAT** in Section 6.4) removes one link in every scheduling round, so considering k-conflict ($k>1$) graphs does not help. As a result, the pair-wise conflict graph suffices.

6. COUNTING BASED EXACT COLORING

6.1 The Inclusion-Exclusion Principle

FACT 6.1 [5]: Let B be a finite set with subsets $A_1, A_2, ..., A_n \subseteq B$, and with the convention that $\bigcap_{i \in \varnothing} A_i = B$, then we know the number of elements in B which lie in none of the A_i is

$$|\bigcap_{i=1}^{n} \overline{A_i}| = \sum_{X \subseteq \{1,...,n\}} (-1)^{|X|} \cdot |\bigcap_{i \in X} A_i|. \qquad (6.1)$$

Now let's define $S = \{S_1, S_2, ..., S_i, ...\}$ where $i \le 2^n$ as the set of the link independent sets, B as the set of k-tuples $(S_1, ..., S_k)$, and $A_i \subseteq B$ as the set of those k-tuples whose union does not include link i; then the left hand side of Equation 6.1 can be interpreted as the number of k-tuples in B which cover all the links from $N = \{1, ..., n\}$. On the right hand side of Equation 6.1, for each X, $|\bigcap_{i \in X} A_i|$ means the number of k-tuples whose union does not include all the links in X.

6.2 Counting the Number of k-Set-Coverings

Here we define a k-set-covering as a set covering in which each covering consists of k link independent sets. Also we use $c_k(S)$ to denote the number of different k-set-coverings. We define $S(X) \subseteq S$ as the set of the link independent sets whose union does not include the links in X, which

means $\bigcup_i S_i = N - X$, where $S_i \in S(X)$. And we use $s(X) = |\{S_i \in S : S_i \bigcap X = \varnothing\}|$ to denote the number of link independent sets in $S(X)$. Then the following lemma holds.

LEMMA 6.2: The number of different k-set coverings satisfies

$$c_k(S) = \sum_{X \subseteq N} (-1)^{|X|} \cdot (s(X))^k \qquad (6.2)$$

PROOF: With $s(X)$ denoting the number of link independent sets in $S(X)$, $(s(X))^k$ stands for the number of different ways to choose k link independent sets from $S(X)$. (Note that the link independent sets in a k-set-covering may be non-distinct and non-disjoint.) Now combining the analyses in Section 6.1, we have $c_k(S) = |\bigcap_{i=1}^{n} \overline{A_i}|$, which is the left hand side of Equation 6.1, and $(s(X))^k = |\bigcap_{i \in X} A_i|$, which is the right hand side of Equation 6.1. This completes the proof.

THEOREM 6.3: Counting k-set coverings can be solved in time $O(3^n \cdot p(n))$ and using polynomial space.

PROOF: According to Equation 6.2, we can see that the computational complexity is dominated by computing $s(X)$, i.e., to count the number of link independent sets in $S(X)$. For each X , we can enumerate all the combinations of the links in set $N - X$, which will take time $2^{n-|X|} \cdot p(n-|X|)$, because each testing of the link independent set takes time $p(n)$. Now combining with Equation 6.2 and the binomial theorem, the running time of counting k-set coverings is $T(c_k(S)) = \sum_{m=0}^{n} \binom{n}{m} \cdot 2^{n-m} \cdot p(n-m) = O(3^n \cdot p(n))$.

Here $p(n)$ subsumes the time of raising each $s(X)$ to the k-th power. For the space complexity, since we compute $s(X)$ anew for each X, the occupied space is definitely polynomial. This finishes the proof.

Remark 4: There has also been a long history of counting the independent sets in graphs with moderately exponential time. For example, the fastest algorithm so far for counting the independent set takes time $O(1.2461^n)$, which is achieved by a reduction from the #2-SAT problem [14]. Unlike counting the independent sets in graphs where each constraint contains only two variables (binary constraint), testing the link independent set involves any subset of the n links (global constraint). This leads us to believe that it is unlikely that we can enumerate all the link independent sets in $N - X$ faster than checking all the subsets in it which takes time at least $\Omega(2^{n-|X|} \cdot p(n))$. So we are inclined to believe that $3^n \cdot p(n)$ is also a time lower bound.

THEOREM 6.4: Counting k-set coverings can be solved in time $O(2^n \cdot p(n))$ with exponential space $O(2^n \cdot n^2)$.

PROOF: We need to introduce the zeta transform of a function f, where f is an indicator function of the link independent set. Specifically, the zeta transform [5] $\widehat{f}$ on the subset lattice $(2^N, \subseteq)$ of f is defined by

$$\widehat{f}(X) = \sum_{S \subseteq X} f(S) \ for \ X \subseteq N. \qquad (6.3)$$

Now since $s(N - X) = \sum_{S \subseteq X} f(S) = \widehat{f}(X)$, we can compute a table containing $s(N - X)$ for all $X \subseteq N$, and using a fast zeta transform introduced in [5], we can compute all $\widehat{f}(X)$ with time $O(2^n \cdot p(n) + 2^n n) = O(2^n \cdot p(n))$. So according to Equation 6.2 and by subsuming the time of raising each $s(X)$ to the k-th power into $p(n)$, the time complexity of computing $c_k(S)$ is reduced to $O(2^n \cdot p(n))$. For the space complexity, since we have stored $O(2^n \cdot n)$ number of interim values for calculating $\widehat{f}(X)$ (including all $s(X)$), and since the value of $s(X)$ can be up to 2^n , the space complexity is $O(2^n \cdot n^2)$. This ends the proof.

6.3 Computing the Minimum Number of Colors

LEMMA 6.5: The **MLSAT** problem can be solved with k colors if and only if $c_k(S) > 0$.

PROOF: On one hand, if all the links can be scheduled with k colors, there must exist a valid k-set covering, which means $c_k(S) > 0$; on the other hand, if $c_k(S) > 0$, there must exist a coloring such that all the links can be scheduled at least once in k timeslots (colors).

Now we use $\chi(N)$ to denote the minimum number of colors to schedule all the $N = \{1,...,n\}$ links. Combining with Lemma 6.5, we have the following corollary.

COROLLARY 6.6: $\chi(N) = \min\{k : c_k(S) > 0\}$.

With the help of binary search, the time for computing $\chi(N)$ becomes $\log n \cdot T(c_k(S))$. So according to Theorems 6.3 and 6.4, we have the following corollaries.

COROLLARY 6.7: If we only allow polynomial space, the minimum number of colors $\chi(N)$ can be computed in time $O(3^n \cdot \log n \cdot p(n))$.

COROLLARY 6.8: If exponential space is allowed, the minimum number of colors $\chi(N)$ can be computed in time $O(2^n \cdot \log n \cdot p(n))$.

6.4 The Exact Scheduling Algorithm

Although we have computed the minimum number of colors to schedule all the links, we have not constructed a practical schedule yet. In this section, we present an algorithm called **ESA_MLSAT** for scheduling each link at least once while guaranteeing the minimum number of colors. (To demonstrate the

use of this algorithm, we give a detailed illustrating example in the Appendix.)

ESA_MLSAT: Exact Scheduling Algorithm for MLSAT with Optimum Power Assignments

---

Input: A set of arbitrarily distributed single-hop wireless links $N = \{1,...,n\}$.

Output: A successful scheduling of all the links under the SINR constraint such that the number of colors is minimized.

1: Construct the pair-wise conflict graph G_{pair} on N;

2: Compute $\chi(N)$, i.e., the minimum number of colors of N.

3: Pick the most constrained link i which has the maximum node degree in the conflict graph, and list all the links in G_{pair} not incident on i. These links form a set $\{j_1, j_2,..., j_m\}$. We construct new pair-wise conflict graphs called $G_{pair}(k)$ ($1 \le k \le m$) on top of G_{pair} by adding the edges between link i and links j_k where $1 \le k \le m$. Let $V(G_{pair})$ and $E(G_{pair})$ denote all the nodes and edges in G_{pair}, then $V(G_{pair}(k)) = V(G_{pair})$ and $E(G_{pair}(k)) = E(G_{pair}) \bigcup \{ij_1,...,ij_k\}$;

4: Let $S(k)$, $1 \le k \le m$ denote the set of all the link independent sets in N but excluding any link independent set containing link pairs incident on link i in $G_{pair}(k)$; similar to Corollary 6.6, we have $\chi(G_{pair}(k)) = \min\{k' : c_{k'}(S(k)) > 0\}$; and from Proposition 2.4, we have $\chi(G_{pair}(k-1)) \le \chi(G_{pair}(k)) \le \chi(G_{pair}(k-1)) + 1$;

5: If $\chi(G_{pair}) = \chi(N) = \chi(G_{pair}(m))$, then we know the color of link i must be different from those of all the other links in some optimal coloring. So we give it a new color number and assign the sender of this link (or the senders of all the actual links if i is a virtual link) based on the optimal power vector P^* (cf. Section 2.1), then we remove i from N. Otherwise, we can find the smallest k using binary search such that $\chi(G_{pair}(k)) = \chi(G_{pair}) + 1$.

In this case, we can deduce that link i must have the same color with link j_k in some optimal coloring (otherwise $\chi(G_{pair}(k)) = \chi(G_{pair})$). We now replace link i and link j_k with a new virtual link p_{ij_k}, and the neighbors of p_{ij_k} in the conflict graph become $N(p_{ij_k}) = N(i) + N(j_k)$.

6: Repeat step 2 to step 5 until all links have been scheduled (colored).

6.5 Correctness Analysis

We call step 2 to step 5 in the **ESA_MLSAT** algorithm a scheduling round. In each scheduling round, we remove one link, either directly giving it a new color or "contracting" two links (step 5). Since in each scheduling round, our link removals are based on the computed minimum number of colors of all the remaining links, and combining with the analyses in step 4 and step 5, we can guarantee the output is optimal, i.e., the number of colors we actually obtain is minimized.

Also, we need to emphasize the computation of the minimum number of colors $\chi(G_{pair}(k))$. Unlike the computation of $\chi(N)$,

which is based on the set of all the link independent sets (e.g., the set S in Sections 6.1 and 6.2), the computation of $\chi(G_{pair}(k))$ is based on $S(k)$ (the set of the link independent sets in step 4). In addition, we must note that, if there are some virtual links in the conflict graph, due to the aggregate interference effect, all the actual links in these virtual links must be taken into account for checking whether the supersets of these links are link independent sets.

6.6 Time Complexity Analysis

Let some exponential function $T(\chi(n))$ denote the time of computing the minimum number of colors of scheduling n links. Since each scheduling round causes at most $O(\log n)$ computations of computing the minimum number of colors, and there are n scheduling rounds, the overall optimal scheduling takes time $O(T(\chi(n)) \cdot n \cdot \log n)$. So from Corollaries 6.7 and 6.8, if only polynomial space is allowed, the time complexity becomes $O(3^n \cdot n \cdot \log^2 n \cdot p(n))$, and if exponential space is allowed, the time complexity of the exact scheduling algorithm is $O(2^n \cdot n \cdot \log^2 n \cdot p(n))$.

7. COUNTING BASED APPROXIMATE COLORINGS

7.1 Polynomial Time Approximation

This approximation algorithm is implemented by clustering. We first partition all the n links into $n/\log n$ groups, each group containing $\log n$ links. Then we use the exponential space version of the **ESA_MLSAT** algorithm to compute the minimum number of colors of each group. Let Opt stand for the minimum number of colors, then the actual number of colors we get is at most $n/\log n \cdot Opt$, and so the approximation ratio is $O(n/\log n)$. Since the time complexity of our exact scheduling algorithm is $O(2^n \cdot n \cdot \log^2 n \cdot p(n))$, and the space complexity is $O(2^n \cdot n^2)$, the time complexity of our approximate scheduling algorithms is bounded by $O(n^2 \cdot poly\log(n))$. The space complexity is $O(n^2 \cdot \log n)$.

7.2 Quasi-polynomial Time Approximation

Obviously, we can also partition all the n links into $n/\log^k n$ groups, each group containing $\log^k n$ links. Then we use the polynomial space version of the **ESA_MLSAT** algorithm to compute the minimum number of colors of each group. The approximation ratio is $O(n/\log^k n)$. But the time complexity becomes $O(n^{1+\log 3 \cdot \log^{k-1} n} poly\log(n))$, which is a quasi-polynomial time complexity, i.e., the complexity with the form $O(\exp((\log n)^{O(1)}))$. The space complexity is still polynomial.

7.3 Exponential Time Approximation

We have given an exponential time approximate link scheduling algorithm in Section 4.2, which is based on repeatedly finding the maximum link independent set on the remaining links. This is equivalent to a standard greedy set covering method with approximation ratio $O(\log n)$. In this section, we will present another exponential time approximation algorithm which is also

based on finding the maximum link independent set. But in this algorithm, when the number of the remaining links equals $e^{-\varepsilon}n$, we do not repeat the maximum link independent set finding algorithm. Instead we use our polynomial space version of the exact link scheduling algorithm **ESA_MLSAT** since the number of the remaining links has become small enough.

THEOREM 7.1: The approximation ratio of this polynomial space approximate link scheduling algorithm is $\lceil (1+\varepsilon) \rceil$, and the time complexity of this algorithm is $O(((\binom{n}{n/2}) + 3^{e^{-\varepsilon}n} \cdot n \cdot \log n) \cdot \log n \cdot p(n))$.

PROOF: The proof is adapted from [4,5].

Let *Opt* be the minimum number of colors for scheduling all the links. And let *s* be the number of maximum link independent sets we have removed. If we use $\chi(e^{-\varepsilon}n)$ to denote the minimum number of colors we have obtained to schedule the remaining $e^{-\varepsilon}n$ links, then the total number of colors we have used is $\chi(e^{-\varepsilon}n) + s$. Since $\chi(e^{-\varepsilon}n) \leq Opt$, we only need to prove that $s \leq \lceil \varepsilon \rceil \cdot Opt$.

Since we remove the maximum link independent set in each step, so after at most *t* steps, the number of remaining links is smaller than or equal to $n \cdot (1-1/Opt)^t$, and due to a standard inequality, we have $n \cdot (1-1/Opt)^t \leq n \cdot e^{-t/Opt}$. So if $\varepsilon \cdot Opt \leq t \leq s \leq \lceil \varepsilon \rceil \cdot Opt$, then the number of remaining links is at most $e^{-\varepsilon}n$. By plugging into the time complexity result of the maximum link independent set finding algorithm in Section 4.2 and the polynomial space version of the exact scheduling algorithm in Section 6.4, we finish the proof.

8. CONCLUSIONS

In this paper, using the physical interference model (SINR model), we have presented both exact and approximate minimum length wireless link scheduling algorithms for arbitrary link topologies. The exact scheduling algorithm called ESA_MLSAT solves an open problem (Problem 1) proposed in [21] at the cost of moderately exponential time. And the proposed approximate scheduling algorithms solve the same problem with polynomial, quasi-polynomial and exponential time, respectively. Based on the inclusion-exclusion principle, the exact link scheduling algorithm is realized by computing the minimum number of colors through counting the number of *k*-set coverings. Depending on whether exponential space is allowed or not, the time complexity of the exact link scheduling algorithm can be reduced from $O(3^n \cdot n \cdot \log^2 n \cdot p(n))$ (polynomial space) to $O(2^n \cdot n \cdot \log^2 n \cdot p(n))$ (exponential space). The approximation ratios of the polynomial time, quasi-polynomial time and exponential time approximation algorithms are $O(n/\log n)$, $O(n/\log^k n)$ and $\lceil 1+\varepsilon \rceil$, respectively. All these approximation algorithms use polynomial space. These results show that the counting versions of the exact and approximate link scheduling algorithms outperform the somewhat simpler set covering algorithms in terms of both the time complexity and the approximation ratio.

Obviously there are limitations in our algorithms. Firstly, the exponential time exact scheduling algorithm is not that practical when the number of the links is large. But when there are not too many links to be scheduled, the moderately exponential time may

be acceptable. Furthermore, just like what is done in the polynomial time approximation algorithm in Section 7.1, for large number of links, we can split them into some small number of groups, and use our exponential time exact scheduling algorithm to handle the links in each group. In addition, in practice, the optimum power assignment can be realized by distributed iterative power control [11,1], where each link only needs to know its own transmission power and the measured SINR value. In each iteration, the transmitter can increase its transmission power if the measured SINR is small, and reduce its power otherwise. It has been shown that this local power control algorithm can converge exponentially to the optimum power vector [11]. Secondly, we have assumed the transmitters can vary their transmission powers even to infinity. If however the maximum powers are limited, we can use the constrained power control methods in [1], or something similar. Thirdly, we need to check whether the spectral radius of a large non-negative matrix is smaller than 1 or not every time, this can be realized by using the technique presented in [32]. Moreover, we do not need to find the exact spectral radius value; instead, we can easily compute an always convergent upper bound in some constant steps, thus greatly lowering the computation overhead and it can be done in polynomial time ($p(n)$).

There are some possible directions that are worth further investigations.

First and foremost, to completely solve the MLSAT problem and make it practical, a distributed and polynomial time approximation algorithm with good approximation ratio is needed. Although we have proposed a polynomial time approximation algorithm in Section 7.1, it is a centralized algorithm which makes implementation difficult in practice.

Second, as mentioned in Section 2.3, a rigorous NP-completeness proof of the MLSAT problem is needed, or otherwise we need to propose a polynomial time exact scheduling algorithm.

Third, the time complexity of our exact scheduling algorithm is determined by the time of counting the link independent sets. Due to the aggregate interference effect of the SINR model (global constraint), we believe that it is very unlikely that we can list all these link independent sets faster than checking all the subsets one by one. So breaking the $\Omega(2^n)$ barrier will certainly be a breakthrough.

Fourth, since our exponential time approximate link scheduling algorithms are implemented by finding the maximum link independent set, and although we have designed an $o(2^n)$ algorithm, a more efficient maximum link independent set finding algorithm can be expected.

Finally, we have assumed the links have uniform demands, so extending our work to non-uniform demands is a natural extension. A possible idea is to view the latter as a set multi-cover problem [31], and thus we may be able to use some related techniques to tackle the non-uniform case.

9. ACKNOWLEDGMENTS

10. REFERENCES

[1] M. Andersin, Z. Rosberg and J. Zander. Gradual Removals in Cellular PCS with Constrained Power Control and Noise. *Wireless Networks*, 2(1): 27-43, 1996.

[2] A. Behzad and I. Rubin. Optimum Integrated Link Scheduling and Power Control for Multihop Wireless Networks. *IEEE Transactions on Vehicular Technology*, 56(1):194-205, 2007.

[3] A. Björklund, T. Husfeldt, P. Kaski and Mikko Koivisto. Fourier Meets Möbius: Fast Subset Convolution. In *Proc. 39th Annual ACM Symposium on Theory of Computing (STOC)*, San Diego, California, US, June 2007.

[4] A. Björklund and T. Husfeldt. Inclusion--Exclusion Algorithms for Counting Set Partitions. In *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Berkeley, California, US, Oct. 2006.

[5] A. Björklund, T. Husfeldt, and M. Koivisto. Set Partitioning via Inclusion--Exclusion. *SIAM Journal on Computing*, to appear.

[6] S. A. Borbash and A. Ephremides. Wireless Link Scheduling With Power Control and SINR Constraints. *IEEE Transactions on Information Theory*, 52(11): 5106-5111, 2006.

[7] G. Brar, D. Blough, and P. Santi. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In *Proc. 12th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, Los Angeles, CA, USA, Sept. 2006.

[8] M. Caramia and P. Dell'olmo. Constraint Propagation in Graph Coloring. *Journal of Heuristics*, 8(1):83-107, 2002.

[9] D. Chafekar, V. S. A. Kumar, M. V. Marathe, S. Parthasarathy and A. Srinivasan. Approximation Algorithms for Computing Capacity of Wireless Networks with SINR Constraints. In *Proc. 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Phoenix, AZ, US, April 2008.

[10] D. Chafekar, V. S. A. Kumar, M. V. Marathe, S. Parthasarathy and A. Srinivasan. Cross-Layer Latency Minimization in Wireless Networks with SINR Constraints. In *Proc. 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Montreal, Canada, Sept. 2007.

[11] T. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad-hoc Networks. In *Proc. INFOCOM*, New York, US, June 2002.

[12] T. Erlebach, K. Jansen and E. Seidel. Polynomial-Time Approximation Schemes for Geometric Intersection Graphs. *SIAM Journal on Computing*: 34(6): 1302-1323, 2005.

[13] F. V. Fomin, F. Grandoni and D. Kratsch. Measure and Conquer: a Simple O(2^0.288n) Independent Set Algorithm. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Miami, Florida, US, Jan. 2006.

[14] M. Fürer and S. P. Kasiviswanathan. Algorithms for Counting 2-Sat Solutions and Colorings with Applications. In *Proc. 3rd International Conference on Algorithmic Aspects in Information and Management (AAIM)*, Portland, OR, US, June 2007.

[15] O. Goussevskaia, Y. A. Oswald, and R.Wattenhofer. Complexity in Geometric SINR. In *Proc. MOBIHOC*, Montreal, Canada, Sept. 2007.

[16] Piyush Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388-404, 2000.

[17] B.E. Hajek and G.H. Sasaki. Link Scheduling in Polynomial Time. *IEEE Transactions on Information Theory*, 34(5): 910-917, 1988.

[18] Q.-S. Hua and F. C. M. Lau. The Scheduling and Energy Complexity of Strong Connectivity in Ultra-Wideband Networks. In *Proc. 9th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Torremolinos, Malaga, Spain, Oct. 2006.

[19] Q.-S. Hua and F.C.M. Lau. Balancing the Scheduling and Energy Complexities in Wireless Networks. Submitted to MSWiM'08.

[20] M. Koivisto. An O*(2^n) Algorithm for Graph Coloring and Other Partitioning Problems via Inclusion--Exclusion. In *Proc. FOCS*, Berkeley, California, US, Oct. 2006.

[21] T. Locher, P. von Rickenbach, and R. Wattenhofer. Sensor Networks Continue to Puzzle: Selected Open Problems (Invited paper). In *Proc. 9th International Conference on Distributed Computing and Networking (ICDCN)*, Kolkata, India, Jan. 2008.

[22] J.W. Moon and L. Moser. On Cliques in Graphs. *Israel Journal of Mathematics*, Vol. 3, pp. 23-28, 1965.

[23] T. Moscibroda, Y. A. Oswald, and R. Wattenhofer. How Optimal are Wireless Scheduling Protocols? In *Proc. INFOCOM*, Anchorage, Alaska, US, May 2007.

[24] T. Moscibroda. The Worst-Case Capacity of Wireless Sensor Networks. In *Proc. 6th International Conference on Information Processing in Sensor Networks (IPSN)*, Cambridge, Massachusetts, US, April 2007.

[25] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol Design Beyond Graph-Based Models. In *Proc. 5th Workshop on Hot Topics in Networks (HotNets)*, Irvine, California, US, Nov. 2006.

[26] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *Proc. INFOCOM*, Barcelona, Spain, April 2006.

[27] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies. In *Proc. MOBIHOC*, Florence, Italy, May 2006.

[28] S. U. Pillai, T. Suel, and S. Cha. The Perron-Frobenius Theorem and Some of its Applications. *IEEE Signal Processing Magazine*, 22(2):62-75, 2005.

[29] A.Richa, C.Scheideler, P.Santi, An O(log n) Dominating Set Protocol for Wireless Ad Hoc Networks under the Physical Interference Model. In *Proc. MOBIHOC*, Hong Kong, China, May 2008.

[30] G. Sharma, R.R. Mazumdar and N.B. Shroff. On the Complexity of Scheduling in Wireless Networks. In *Proc. MOBICOM*, Los Angeles, CA, USA, Sept. 2006.

[31] V. V.Vazirani. Approximation Algorithms. Berlin: Springer, 2003.

[32] R. J. Wood and M. J. O'Neill. An Always Convergent Method for Finding the Spectral Radius of an Irreducible Non-Negative Matrix. *The ANZIAM Journal*, V45:C474-C485, 2004.

[33] J. Zander. Performance of Optimum Transmitter Power Control in Cellular Radio Systems. *IEEE Transactions on Vehicular Technology*, 41(1):57-62, 1992.

11. APPENDIX: A Simple Example

Suppose there are five links $N = \{1,2,3,4,5\}$, and all the maximal link independent sets have been computed: $\{\{1,3\},\{2,4\},\{3,5\},\{1,2,5\}\}$. Recall that X stands for any subset of N, and $S(X)$ represents the set of all the link independent sets in $N - X$ and $s(X)$ means the number of link independent sets in $S(X)$. For clarity of presentation, we use a simpler notation to denote the link independent sets in $S(X) \subseteq S$ (Table 1); for example, we use *1* to denote the link independent set *{1}*, and *125* to denote the link independent set *{1,2,5}*.

The 1st step of the **ESA_MLSAT** algorithm is to construct the pair-wise conflict graph G_{pair}, which is shown in Fig.1(a). The 2nd step is to compute the minimum number of colors $\chi(N)$. According to Table 1, we have $S=\{1,2,3,4,5,12,13,15,24,25,35,125\}$, and we can calculate that $c_1(S) = c_2(S) = 0$, and $c_3(S) = \sum_{X \subseteq N} (-1)^{|X|} \cdot (s(X))^3 = 96 > 0$, and so we know that $\chi(N) = 3$. In the 3rd step we pick the most

constrained link *4* and add new edges (additional constraints) between link *4* and all the other links which are not incident on it. In this example, only one link $j_1 = 2$ (link *2*) is not incident on link *4*, so we add a new edge between them ($G_{pair}(1)$ as shown in Fig. 1(b)). In the 4th step, by removing all the link independent sets containing links *4* and *2*, we can achieve $S(1) = \{1,2,3,4,5,12,13,15,25,35,125\}$, and since $c_1(S(1)) = c_2(S(1)) = 0$ and $c_3(S(1)) = 30 > 0$, we conclude that $\chi(G_{pair}(1)) = 3$. Then we go to the 5th step, since we have known that $\chi(G_{pair}) = \chi(N) = \chi(G_{pair}(1)) = 3$, we can deduce that, in some optimal coloring, link *4* must have a different color with all the other links, and so we give it a new color number and remove it from *N*. Now we have finished the first scheduling round; we then repeat step 2 to step 5 until all links have been colored. We now briefly give the following scheduling rounds below.
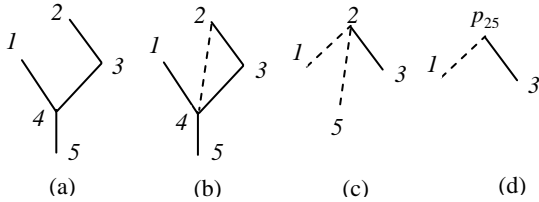

(a)   (b)   (c)   (d)

Figure 1(a): The original pair-wise conflict graph G_{pair} for the five links *N={1,2,3,4,5}*;

Figure 1(b): A new conflict graph $G_{pair}(1)$ constructed on G_{pair};
Figure 1(c): A new conflict graph $G_{pair}(2)$ constructed on the remaining links *N = {1,2,3,5}*;

Figure 1(d): A new conflict graph $G_{pair}(1)$ constructed on the remaining links *N = {1, p_{25}, 3}*.

The 2nd scheduling round: In the 2nd step, similar to Table 1, we can construct another table for *N = {1,2,3,5}*, and we have *S={1,2,3,5,12,13,15,25,35,125}*. Then we can calculate that $c_1(S) = 0$ and $c_2(S) = 10 > 0$, and so we know that $\chi(N) = \chi(G_{pair}) = 2$. In the 3rd step, we pick link *2* as the most constrained link, and add new edges between link *2* and links $j_1 = 1$ and $j_2 = 5$ ($G_{pair}(2)$ as shown in Fig. 1(c)). Note that by reducing the edge between links *2* and *5* from $G_{pair}(2)$, we can get $G_{pair}(1)$. In the 4th and 5th steps, we have $S(2) = \{1,2,3,5,13,15,35\}$ andsince $c_2(S(2)) = 0$ and $c_3(S(2)) = 36 > 0$, we know $\chi(G_{pair}(2)) = 3 > \chi(N)$, and then we continue to find that $S(1) = \{1,2,3,5,13,15,25,35\}$ and since $c_2(S(1)) = 2 > 0$, we get $\chi(G_{pair}(1)) = 2 = \chi(N)$. So in this case, we conclude that $k = 2$ (corresponding to link j_2) is the smallest *k* to satisfy $\chi(G_{pair}(k)) = \chi(G_{pair}) + 1$. We then deduce that link *2* must have the same color with link *5* in this optimal coloring. So we contract these two links into a new link p_{25}. Then we go to the 3rd scheduling round.

The 3rd scheduling round: In the 2nd step, also similar to Table 1, we can construct another table for *N = {1, p_{25}, 3}*, and we

have $S = \{1, p_{25}, 3, 1p_{25}, 13\}$, and then we can calculate that $c_1(S) = 0$ and $c_2(S) = 6 > 0$, and so we know that $\chi(N) = \chi(G_{pair}) = 2$. In the 3rd step, we pick link p_{25} as the most constrained link, and add a new edge between link p_{25} and link $j_1 = 1$ ($G_{pair}(1)$ as shown in Fig. 1(d)). In the 4th and 5th steps, we have $S(1) = \{1, p_{25}, 3, 13\}$ and since $c_2(S(1)) = 2 > 0$, we obtain $\chi(G_{pair}(1)) = 2 = \chi(N)$, and so we conclude that link p_{25} must have a different color with link *1* and link *3* in this optimal coloring. Then we give it a new color and remove it from *N*. Now we finish the 3rd scheduling round and can proceed to the 4th scheduling round.

The 4th scheduling round: We can easily find that links *1* and *3* must have the same color in this optimal coloring (the interested reader can do the checking). So we give them a new color and we finish the scheduling of all the links. Also the transmission powers of all the links are based on the optimal power vector P^* in Section 2.1. The final result is we have used three colors for the link independent sets *{4},{2,5}* and *{1,3}*. Of course, this is only one of the optimal colorings. By choosing different coloring strategies or through choosing different orders of the links in step 3 of the **ESA_MLSAT** algorithm, we may obtain different optimal colorings.

Table 1: For each subset *X* of *N = {1,2,3,4,5}*, the number of link independent sets $s(X)$ in $S(X) = N - X$

X	$S(X)$	$s(X)$
$\varnothing$	*{1,2,3,4,5,12,13,15,24,25,35,125}*	12
{1}	*{2,3,4,5,24,25,35}*	7
{2}	*{1,3,4,5,13,15,35}*	7
{3}	*{1,2,4,5,12,15,24,25,125}*	9
{4}	*{1,2,3,5,12,13,15,25,35,125}*	10
{5}	*{1,2,3,4,12,13,24}*	7
{1,2}	*{3,4,5,35}*	4
{1,3}	*{2,4,5,24,25}*	5
{1,4}	*{2,3,5,25,35}*	5
{1,5}	*{2,3,4,24}*	4
{2,3}	*{1,4,5,15}*	4
{2,4}	*{1,3,5,13,15,35}*	6
{2,5}	*{1,3,4,13}*	4
{3,4}	*{1,2,5,12,15,25,125}*	7
{3,5}	*{1,2,4,12,24}*	5
{4,5}	*{1,2,3,12,13}*	5
{1,2,3}	*{4,5}*	2
{1,2,4}	*{3,5,35}*	3
{1,2,5}	*{3,4}*	2
{1,3,4}	*{2,5,25}*	3
{1,3,5}	*{2,4,24}*	3
{1,4,5}	*{2,3}*	2
{2,3,4}	*{1,5,15}*	3
{2,3,5}	*{1,4}*	2
{2,4,5}	*{1,3,13}*	3
{3,4,5}	*{1,2,12}*	3
{1,2,3,4}	*{5}*	1
{1,2,3,5}	*{4}*	1
{1,2,4,5}	*{3}*	1
{1,3,4,5}	*{2}*	1
{2,3,4,5}	*{1}*	1
{1,2,3,4,5}}	$\varnothing$	0