J Comb Optim (2016) 32:1052–1067
DOI 10.1007/s10878-015-9908-4

An improved approximation algorithm for the shortest link scheduling in wireless networks under SINR and hypergraph models

Cui Wang[1] · Jiguo Yu[1] · Dongxiao Yu[2] · Baogui Huang[1] · Shanshan Yu[3]

Published online: 4 June 2015

Abstract Link scheduling is a fundamental problem in wireless ad hoc and sensor networks. In this paper, we focus on the shortest link scheduling (SLS) under Signal-to-Interference-plus-Noise-Ratio and hypergraph models, and propose an approximation algorithm SLS_{pc} (A link scheduling algorithm with oblivious power assignment for the shortest link scheduling) with oblivious power assignment for better performance than GOW* proposed by Blough et al. [IEEE/ACM Trans Netw 18(6):1701–1712, 2010]. For the average scheduling length of SLS_{pc} is $1/m$ of GOW*, where $m = \lfloor \Delta_{max} \cdot p \rfloor$ is the expected number of the links in the set V returned by the algorithm HyperMaxLS (Maximal links schedule under hypergraph model) and $0 < p < 1$ is the constant. In the worst, ideal and average cases, the ratios of time complexity of our algorithm SLS_{pc} to that of GOW* are $O(\Delta_{max}/\bar{k})$, $O(1/(\bar{k} \cdot \Delta_{max}))$ and $O(\Delta_{max}/(\bar{k} \cdot m))$,

✉ Jiguo Yu
jiguoyu@sina.com

Cui Wang
wangcui6600@163.com

Dongxiao Yu
dxyu@hku.hk

Baogui Huang
hjbaogui@163.com

Shanshan Yu
shanshanyu@sdu.edu.cn

1 School of Information Science and Engineering, Qufu Normal University, Rizhao, Shandong 276826, China

2 Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong, China

3 School of Information Science and Engineering, Shandong University, Jinan 250100, China

respectively. Where $\bar{k}$ $(1 < \bar{k} < \Delta_{max})$ is a constant called the SNR diversity of an instance G.

Keywords Wireless network · Shortest link scheduling · Approximation algorithm · Hypergraph model · SINR

1 Introduction

During the past two decades, wireless ad hoc and sensor networks have been applied in various areas, such as battled field surveillance, traffic monitoring, environment monitoring and so on, the problems of drawing dominant dataset (Cheng et al. 2015), composite event coverage (Gao et al. 2015), tracking quantiles (He et al. 2014) and scheduling (Goussevskaia et al. 2007) in wireless networks are getting people to pay attention. Compared to a wired network, a wireless network has a fundamental property of sharing the common (radio) communication channels. Therefore, the links will inevitably interfere with each other as they transmit simultaneously. Due to the inherent interference property of wireless networks, one of the major challenges is to decrease interference and maximize the throughput of the network by appropriate scheduling strategy. Link scheduling is an important approach to decrease interference among the transmission links and enhance the throughput of the network. Link scheduling problem in a wireless network can roughly be classified into two sub-problems. One is single slot scheduling (One-Slot Scheduling) problem (Goussevskaia et al. 2007), and the other is the shortest link scheduling (SLS) problem (Wan et al. 2010). The former is also known as the maximum independent set link (MISL) (Pei and Anil Kumar 2012) or maximum link scheduling problem (MaxLSP) (Pei et al. 2012). That is, for a given set of links $L = \{l_1, l_2, \ldots, l_n\}$, compute the largest possible subset of links $S \subseteq L$ that can be scheduled concurrently without conflicts. The latter is represented by $S = \{S_1, S_2, \ldots, S_t\}$, where S_i, $1 \leq i \leq t$, denotes a subset of links of L, designated to time slot i. The number t is referred as the scheduled length or the scheduled slots. SLS is to schedule all the links within the minimum slots.

In wireless networks, interference models play a very important role in the design of algorithms. The performance of a designed algorithm under different interference models may have great differences. Roughly speaking, interference models can be classified into the graph-based models and the physical interference model. The graph-based models (Hajek and Sasaki 1988), such as the unit disk graph (UDG) (Schmidt and Wattenhofer 2006), quasi unit disk graph (QUDG) (Schmidt and Wattenhofer 2006) and the protocol interference model (Gupta and Kumar 2000; Schmidt and Wattenhofer 2006) and so on, are essentially binary interference models. This implies that the transmission from the sending node s to the receiving node r is successful if and only if r is within the transmission range of s, and there is no other concurrent transmission links within the interference range of r. The graph-based models are too idealistic, and can not accurately present the interference among all the links due to it ignores the cumulative interference. There exist gaps compared with the actual application environment. For instance, the transmission from the sender s to the receiver r is failure in the actual application since there are walls or other obstacles between the two nodes while the two nodes are adjacent in a physical location. In Gupta and Kumar (2000), Gupta and Kumar first proposed Signal-to-Interference-plus-Noise-Ratio (SINR) model, a kind of physical interference model. Under SINR, the transmission from the node s to the node r is successful if and only if the SINR value of receiver r is greater than a certain threshold, whose value depends on the desired channel characteristics. Thus, the interference is a global interference rather than a binary interference. And all the links in the network have to be considered. In other words, even if the transmission from a far node may also interfere or even prevent the successful transmission of the current node.

We have known that any scheduling algorithm under the global SINR model will require coordination among all the user nodes in the network, which makes it very difficult to design distributed scheduling algorithms. It is difficult to design algorithms under the SINR, a global interference model. Li et al. proposed a new interference model called the hypergraph model in Li et al. (2008) and Li and Negi (2012). This model is different from the hypergraph proposed in Sarkar and Sivarajan (1998). The hypergraph model considers the cumulative interference, which is more accurate than the traditional binary graph models. Unlike the global SINR model, the hypergraph model allows much easier scheduling design and analysis by extending the existing rich body of work on graph-based scheduling due to the structural similarities between graph-based models and SINR model. The hypergraph model can achieve a systematic trade-off between the interference approximation accuracy and user coordination complexity during scheduling (Li and Negi 2012). As an application of the hypergraph model, Li et al. gave a simple distributed scheduling algorithm for single-slot link scheduling problems. They obtained a lower bound stability region which achieves a fixed fraction of the optimal stability region for any maximal scheduler, proved the interference approximation accuracy of hypergraph model in random networks, and showed that hypergraphs with small hyperedge sizes can model the interference accurately. Thus that, we adopt the hypergraph model in our paper.

In this paper, we mainly focus on the design of SLS algorithm with oblivious power assignments under the SINR model and hypergraph model. The main idea of the algorithms comes from the approximation algorithm in Goussevskaia et al. (2007) and Blough et al. (2010). Our algorithm improve the algorithm GOW* in Blough et al. (2010), since the exact SINR interference model was used by GOW*, in other words, the links are grouped according to a SNR-based criterion (Blough et al. 2010) instead of distance-based criterion (Goussevskaia et al. 2007). Firstly, since the hypergraph model can limit the interference of links around itself, the interference from far links can be negligible. Then, based on the characteristic of hypergraph and the ideas of GOW* in Blough et al. (2010), we give an improved algorithm SLS_{pc} (A link scheduling algorithm with oblivious power assignment for the shortest link scheduling) with oblivious power assignment for SLS, and prove that the schedule length of the SLS_{pc} is $O(\Delta_{max})$ in the worst case in which the links in each square S must be sequentially scheduled, where Δ_{max} is maximum number of the receivers in the partitioned squares. The average length of the scheduling is $\Theta(\Delta_{max}/m)$, where

$m = \lfloor \Delta_{max} \cdot p \rfloor$ is the expected number of the links in the set V returned by the algorithm HyperMaxLS (Maximal links schedule under hypergraph model) and $0 < p < 1$ is the constant. Compared with the algorithm GOW* in Blough et al. (2010) whose scheduling length is $O(|C_0| + \Delta_{max})$, where $|C_0|$ is number of links in the class C_0 in which the link must be sequentially scheduled. In the worst case, the scheduling length of SLS_{pc} decreases by C_0, and the average length of scheduling is $1/m$ of GOW*. In the worst case, ideal case and the average case, the ratios of the time complexity of SLS_{pc} to that of the GOW* are $O(\Delta_{max}/\overline{k})$, $O(1/(\overline{k} \cdot \Delta_{max}))$ and $O(\Delta_{max}/(\overline{k} \cdot m))$, respectively. Constant $\overline{k}$ $(1 < \overline{k} < \Delta_{max})$ is called the SNR diversity of an instance G.

The rest of the paper is organized as follows. Section 2 summarizes the related work. In Sect. 3, we introduce interference models. Section 4 presents the approximation algorithm, related analysis for SLS. In Sect. 5, we show the simulation results. We conclude this paper in Sect. 6.

2 Related work

Scheduling is an important technology in wireless networks. Many scheduling algorithms have been proposed for various problems such as data aggregation (Li et al. 2014; Guo et al. 2014), parting (Cai et al. 2015), unicast and convergecast (Ji et al. 2013) and so on. In this paper, we focus on th link scheduling problem in wireless networks. In Nelson and Kleinrock (1985), Nelson and Kleinrock first studied the classical problem of scheduling transmission, which has been investigated in isolation known as link scheduling (Goussevskaia et al. 2007), in multihop wireless networks. For the shared nature of the communication medium in a wireless network, a major challenge for link scheduling problem is the wireless interference which limits the link transmission in parallel. Several models have been considered in the literature, which can be roughly classified into graph-based model, SINR-based model. Many scheduling algorithms in the previous works such as *e.g.*, Dimakis and Walrand (2006), Sharma et al. (2006), Chaporkar et al. (2008); Afek et al. (2011), are proposed based on graph-based models since the models usually are well understood, due to its simplicity and abundant results of graph theory. However, the graph-based models have also been regarded as a rigorous model which oversimplifies the interference constrains in wireless networks, since it does not take into account the cumulative effect of interference.

The link scheduling algorithms under SINR model are more accurate than that under the graph-based models since the physical interference model can accurately describe the interference in wireless networks. In Goussevskaia et al. (2007), Goussevskaia, Oswald and Wattenhofer proved that two problems, *i.e.*, SLS and MaxLSP, under geometric SINR model, are NP-hard. For MaxLSP, they proposed an approximation algorithm of factor $O(g(L))$ with uniform powers, where $g(L)$ is the length diversity. For SLS, the authors gave an $O(g(L))$ approximation algorithm, where $g(L)$ indicates that the link length diversity in the network. From then on, many approximation algorithms under SINR model have been recognized. In Goussevskaia et al. (2009), Goussevskaia et al. proposed an $O(\log(n))$ approximation factor guaranteed algorithm for MaxLSP, where n is the total number of links. In Pei and Anil Kumar (2012), Pei and Kumar proposed a fast distributed algorithm under the SINR model, whose running time is $O(g(L)^c m)$, where $c = 1, 2$ or 3 for different problem instances, and m is the number of links. In Blough et al. (2010), Blough, Resta and Santi proved the first known true approximation algorithms for transmission scheduling under the exact SINR model, and obtained an upper bound on the length of SLS algorithm, which is $O(|C_0| + \Delta_{max})$, where $|C_0|$ is the number of links in class C_0, and Δ_{max} is the maximal number of receivers in a cell of class C_k. In Kompella et al. (2010), Kompella et al. addressed the SLS problem under SINR model in wireless networks as a cross layer optimization problem, and considered the parameters of link layer, physical layer, and dynamic power in order to get a viable matching. In Wan et al. (2010), Wan et al. proposed a polynomial $O(\beta \ln \alpha)$-approximation algorithm for SLS with power control under SINR, where α is the independence number, β is the power density. In Halldórsson and Mitra (2011), Halldórsson et al. proved a distributed $O(\log n)$—approximation algorithm for SLS, the algorithm has best ratio when compared with centralized algorithms. Efficient scheduling algorithms under the SINR model has been widely recognized, however, low complexity scheduling is still far from being solved. The reason is that the global interference, *i.e.*, the transmission of any link will be received by other links in the network, weakens its own communication quality. This means that the scheduling algorithms under the SINR model have to coordinate all the links in the network, which makes it very difficult to design the distributed algorithms.

In Li et al. (2008) and Li and Negi (2012), proposed a hypergraph model based on the graph-based and SINR models, and pointed out that the hypergraph model can combine the advantages of both graph-based models and the SINR model while avoiding their drawbacks. Namely, the hypergraph model can emulate cumulative interference constraints as *hyperedge*, where each hyperedge is a set of links that are not allowed to transmit simultaneously. Thus, the hypergraph model can avoid the defects of graph-based models which overlook the cumulative interference. Moreover, since a major portion of the total interference is caused by only a few nearby transmitting links in special wireless networks, the hypergraph can approximate the SINR locally with very good accuracy. Therefore, the hypergraph model can restrict the interference to local.

The algorithms mentioned above adopt uniform power assignment. Many algorithms for link scheduling with non-uniform power assignment are also addressed by people in the literature (ElBatt and Ephremides 2004; Kozat and Koutsopoulos 2006; Kesselheim 2011; Chaorkar and Proutiere 2013; Wan et al. 2013). In Kozat and Koutsopoulos (2006), Kozal et al. proved the problem of power control and scheduling to be NP-complete by using a reduction from integer programming under the assumption that the values of gain matrix can be chose arbitrarily. In Kesselheim (2011), Kesslheim studied the capacity maximization problem with power control under SINR model in a wireless network. They presented the algorithm which selected a subset of links and chosen a power level for each link with the objective of maximizing the number of simultaneous communications, and proved that the algorithm can achieve $\Omega(n)$, $O(\log n)$ approximation in a fading metric and a general metric space, respectively, where n is the number of links. In Chaorkar and Proutiere (2013), based on a combination of simple and efficient power allocation strategy and randomization techniques, Chaporkar and Proutiere presented the first distributed scheduling algorithms under SINR model, they claimed that the algorithms are optimal under the SINR model, and do not require any message passing. In this paper, we design an SLS algorithm SLS_{pc} under the hypergraph and SINR model with oblivious power assignment.

3 Interference models

Let $L = \{l_1, l_2, \ldots, l_n\}$ denote the set of links, where each link l_i represents a communication request from a sender s_i to a receiver r_i, $d_{ij} = d(s_i, r_j)$ denotes the Euclidean distance between s_i and r_j. Thus, the length of link l_j is d_{jj}.

Let $P_{l_i} = c \cdot d_{ii}^{\rho\alpha}$ denote the transmission power of link l_i, where $c > 0$ and $0 \leq \rho \leq 1$ are constants, the path loss factor α $(2 < \alpha < 6)$ is a constant. If each link uses the same power *i.e.*, $\rho = 0$, the power assignment is called uniform power assignment or fixed power assignment. When $\rho = 1/2$ and $\rho = 1$, the power assignment is called mean power assignment and linear power assignment, respectively.

We use the path loss model as energy consumption model. That is,

$$P_{r_j}(s_i) = \frac{P_{l_i}}{d_{ij}^{\alpha}}. \tag{1}$$

where $P_{r_j}(s_i)$ is the received power of the signal from s_i to r_j.

We adopt the *SINR* model, in which a node r_i successfully receives the message from a sender s_i if only if the following condition holds

$$SINR(r_i) = \frac{P_{r_i}(s_i)}{N + I_{r_i}} = \frac{P_{r_i}(s_i)}{N + \Sigma_{l_j \in W \backslash \{l_i\}} P_{r_i}(s_j)} = \frac{P_{l_i}/d_{ii}^{\alpha}}{N + \Sigma_{l_j \in W \backslash \{l_i\}} P_{l_j}/d_{ji}^{\alpha}} \geq \beta. \tag{2}$$

where N is the ambient noise, I_{r_i} is the sum of the interference power experienced by the receiver r_i of link l_i, W is the set of concurrently scheduled links in the same slot, and $\beta \geq 1$ represents the threshold value, *i.e.*, the minimum SINR required for the message to be successful received. We call a link set W is valid if the SINR of any transmitting link in W satisfies (2).

A *hypergraph interference model* (Li et al. 2008; Li and Negi 2012) is a hypergraph $H = (L, \varepsilon)$, where L is a link set, and ε is the set of hyperedges such that each hyperedge $e = (\{l_{i_1}\}, \{l_{i_2}, l_{i_3}, \ldots, l_{i_{k-1}}\}) \in \varepsilon$ consists of a subset of links, in which the links are not allowed to be transmitted together. For any two link sets $S, T \subseteq L$, $e = (S, T) \in \varepsilon$ if and only if S and T satisfy

(1) When the links in $S \cup T$ are scheduled, all the links in S fail and none of T fails.
(2) If any link in $S \cup T$ is removed, no failure occurs if only the remaining links are scheduled.

The condition (1) is validity, and (2) is minimality. For instance, the set of links $\{\{l_1\}, \{l_2, l_3\}\}$ is a hyperedge, and while the set $\{\{l_1\}, \{l_2, l_3, l_4\}\}$ can not form a hyperedge. The reason is that, by referring the condition 2, we can delete $\{l_4\}$ from $\{\{l_1\}, \{l_2, l_3, l_4\}\}$, then $\{\{l_1\}, \{l_2, l_3\}\}$ can be scheduled simultaneously. However, by the condition $\{\{l_1\}, \{l_2, l_3\}\}$ is a hyperedge, $\{l_1\}$ can not be scheduled with $\{l_2, l_3\}$ when we scheduled $\{\{l_1\}, \{l_2, l_3\}\}$, which is a contradiction.

4 The approximation algorithm for SLS

In this section, we give the shortest link scheduling algorithm SLS_{pc} (A link scheduling algorithm with oblivious power assignment for the shortest link scheduling) with a better performance by improving the algorithm GOW* in Blough et al. (2010).

In order to understand the the differences between algorithm SLS_{pc} and GOW*, we first review the main implementation of algorithm GOW*. In GOW*, the links are partitioned into classes $C_0, C_1, \ldots, C_{\lfloor log_{1+\epsilon}(P/\beta N)\rfloor}$ according to SNR-based criterion, with the property that C_i is the set of link l_i such that $D_{i+1} = (\frac{P}{(1+\varepsilon)^{i+1}\beta N})^{1/\alpha} < L_i \leq (\frac{P}{(1+\varepsilon)^{i}\beta N})^{1/\alpha} = D_i$, where $\epsilon \geq \frac{1}{7}$. The links in C_0 are called *black and gray links* whose SNR is below $(1+\varepsilon)\beta$ and greater than β. GOW* sequentially schedules all the links in class C_0. Then, for each class C_i, where $i \geq 1$, it is defined that a proper square partitioning of the deployment region. Squares are 4-colored in such a way that no two adjacent squares have the same color. Finally, links are greedily scheduled in successive slots, with the property that only links with the same color whose receivers are in different squares are assigned to the same slot.

The differences between algorithm SLS_{pc} and GOW* are as follows.

(1) The outer for loop of algorithm GOW*, which defines a proper square partitioning of the deployment region for each class Ci, can be omitted. We divide the network deployment region and set the size of squares at the beginning of the algorithm only once since the links are not classified. (2) the *black and gray links* in Blough et al. (2010), whose length is equal or near to the maximum transmission range of the sender, must be sequentially scheduled. In our paper, the *black and gray links* can be eliminated by improving their transmitting power. (3) in Blough et al. (2010), the transmitting power is uniform, which is a simple power assignment. In SLS_{pc}, we adopt the oblivious power assignment, which only depends on the length of links. (4) we improve the way of choosing the links in GOW*, and select multiple links instead of only one link in GOW*. We expect to choose as more links at each time slot as possible. Therefore, the number of slots for scheduling all links will be decreased, and the scheduling length will be reduced accordingly.

Assume that the minimum length and maximum length of links in the networks are $d_{l_{min}}$ and $d_{l_{max}}$, respectively. Divide the deployment region into squares of width $\mu(d_{l_{min}}+d_{l_{max}})/2$, where constant μ is defined as $\mu = 2[\frac{8k\beta(\beta+\tau)(\alpha-1)}{\tau(\alpha-2)}]^{\frac{1}{\alpha}}$, the constant $\tau > 0$, $k = \Delta_{max} + 1$ is the upper bound of the links in the set V returned by HyperMaxLS, and Δ_{max} is the maximal number of receivers in a square (Blough et al. 2010).

The algorithm needs to call a sub-algorithm HyperMaxLS in order to schedule all the links in the minimum slots. Now, we describe SLS_{pc} with pseudo-codes.

Algorithm 1 A link scheduling algorithm with oblivious power assignment for the shortest link scheduling (SLS_{pc})

Input A set L containing n links with the length of shortest link and the length of longest link are $d_{l_{min}}$ and $d_{l_{max}}$, respectively.
Output A feasible schedule $S_1, S_2, \ldots, S_t$.

```
1: t = 0;
2: Partition network deployment region into squares of width μ(d_lmin + d_lmax)/2 and 4−color the squares
   such that no two adjacent squares have the same color.
3: for j = 1, 2, 3 and 4 do
4:     repeat
5:     For each square A of color j, in which composes the set L', assign power P_li = c · d_ii^{ρα}
6:     for each link l_i in the set L'. Implement the algorithm HyperMaxLS on the set L', S'
7:     is the set V returned by algorithm HyperMaxLS;
8:     L' = L' \ S';
9:     L_j = L_j ∪ S';
10:     t = t + 1;
11:     S_t = L_j;
12:     until all the links in selected squares are scheduled;
13: end for
14: return S_1, S_2, ..., S_t
```

We call the set S_i is *valid (feasible)*, if each link l_i in the set S_i satisfies

$$\frac{P_{l_i}/d_{ii}^{\alpha}}{N + \sum_{j \in e_i \setminus \{l_i\}} P_{l_j}/d_{ji}^{\alpha}} \geq \beta + \tau. \tag{3}$$

The set T_i is *valid (feasible)*, if each link l_j in the set T_i satisfies

$$\frac{P_{l_j}/d_{jj}^{\alpha}}{N + \sum_{\iota \in T_i \setminus \{l_j\}} P_{l_\iota}/d_{\iota j}^{\alpha}} \geq \beta + \tau. \tag{4}$$

where $e_i = S_i \cup T_i$ in (3) and (4), and $\tau > 0$ is an arbitrarily small constant.

Next, we give a simple and intuitive scheduling algorithm HyperMaxLS (Maximal links schedule under Hypergraph model), in which we choose the links from a given set of links. Firstly, we choose any link as the key link, and find out a hyperedge which contains the maximum links. Then, according to the definition of *active hyperedge*, a hyperedge $e_i = (\{l_i\}, \{l_{i_1}, l_{i_2}, \ldots, l_{i_{k-1}}\}) \in \varepsilon$ with respect to link l_i is *active* if all links in e_i except l_i can be scheduled in Li et al. (2008) and Li and Negi (2012), the algorithm HyperMaxLS can find out the maximal set of feasible scheduling by forming the active hyperedge for each link. The following pseudo-codes describe the algorithm HyperMaxLS.

Algorithm 2 Maximal links schedule under hypergraph model (HyperMaxLS)

Input A set $L^{'}$ of links located arbitrarily in the Euclidean plane.
Output A maximal set of feasible links.
1: **Initially** $\varepsilon = \phi$, $V = \phi$, $S_i = \phi$, $Ti = \phi$;
2: $L_i^{'} = L^{'}$;
3: Choose any link $l_i \in L_i^{'}$, assign power $P_{l_i} = c \cdot d_{ii}^{\rho\alpha}$ for l_i and $l_j \in L_i^{'} \setminus l_i$ with power $P_{l_j} = c \cdot d_{jj}^{\rho\alpha}$;
4: $S_i = \{l_i\}$;
5: **while** $(L_i^{'} \neq \phi)$
6: $T_i = T_i \cup \{l_j\}$;
7: **if** (any hyperedge $e \subseteq S_i \cup T_i$)
8: **else** **if**(S_i is invalid)
9: **if** (T_i is valid)
10: $e_i = S_i \cup T_i$ will join the ε;
11: **end if**
12: **else** **if** (T_i is valid)
13: /* in order to compensate for the deprived links in the line 18 */
14: $T_i = T_i \cup \{l_j\}$;
15: **end if**
16: **end if**
17: **end if**
18: $T_i = T_i \setminus \{l_j\}$;
19: $L_i^{'} = L_i^{'} \setminus \{l_j\}$;
20: **end while**
21: **return** $V = \arg\max_{T_i \in e_i \in \varepsilon} |T_i|$

Based on the Algorithm 1 and Algorithm 2, We will give the prove the correctness and the time complexity, determine the scheduling length of the algorithm SLS_{pc} by the following theorems.

Theorem 1 *The set V constructed by algorithm HyperMaxLS is feasible.*

Proof From the line 8–16 of HyperMaxLS, we can know that if and only if both S_i and T_i are valid, the link l_j can join the set T_i. Moreover, according to the definition of the hyperedge that all links in the set T_i can be scheduled together and $T_i \in e_i \in \varepsilon$ (the line 21 of HyperMaxLS), we know that the set T_i is valid. In other words, the algorithm can confirm that each SINR value of links in the set T_i is no less than $\beta + \tau$. Thus that, the set V is feasible. □

Theorem 2 *The schedule computed by algorithm SLS_{pc} is feasible under the SINR model.*

Proof Assume that the maximum interference experienced by the link l_i in the set V is I, namely, $\frac{P_{l_i}/d_{ii}^{\alpha}}{N+I} \geq \beta + \tau$, $k = \Delta_{max} + 1$ is the upper bound of the links in the set V. Now, we calculate the upper-bounded of the total interference I_r contributed by the links in the squares except S and experienced by the link l_i which receiver in a square S. The inner frame contains 8 squares, the second frame contains 16 squares, and in general the h^{th} frame will contains $(2h+1)^2 - (2h-1)^2 = 8 \cdot h$ squares. The generic receiver contained in the h^{th} frame will be at least $(2h-1)\mu(d_{l_{min}} + d_{l_{max}})/2$ apart from r_i. Considering the minimum distance between r_i and a sender relative to frame h is $d_{l_{min}}((2h-1)\mu-1)$. Thus, the total interference I_r experienced by r_i can be bounded by

$$
\begin{aligned}
I_r &< \sum_{h=1}^{\infty} \frac{8P_{l_{max}}k \cdot h}{d_{l_{min}}^{\alpha}((2h-1)\mu-1)^{\alpha}} \\
&\le \frac{8P_{l_{max}}k}{d_{l_{min}}^{\alpha}} \sum_{h=1}^{\infty} \frac{h}{(1/2)^{\alpha}((2h-1)\mu)^{\alpha}} \\
&= \frac{8P_{l_{max}}k}{(1/2)^{\alpha} d_{l_{min}}^{\alpha} \mu^{\alpha}} \sum_{h=1}^{\infty} \frac{h}{(2h-1)^{\alpha}} \\
&\le \frac{8P_{l_{max}}k}{(1/2)^{\alpha} d_{l_{min}}^{\alpha} \mu^{\alpha}} \sum_{h=1}^{\infty} \frac{1}{h^{\alpha-1}} \\
&\le \frac{8P_{l_{max}}k}{(1/2)^{\alpha} d_{l_{min}}^{\alpha} \mu^{\alpha}} \cdot \frac{\alpha-1}{\alpha-2},
\end{aligned}
$$

where the second inequality holds because $x-1 > x/2$ for $x > 2$ and indeed $(2h-1)\mu$ is always greater than 2, and the last inequality holds from on Riemanns zeta function.

Sequentially, we need to prove that any link l_i in the set V returned by HyperMaxLS is valid after being experienced the interference I_r. That is,

$$
\frac{P_{r_i}(s_i)}{N+I+I_r} \ge \beta. \tag{5}
$$

In order to prove that (5) holds, we may prove $\frac{N+I+I_r}{P_{r_i}(s_i)} \le 1/\beta$. Note that, $P_{r_i}(s_i) = P_{l_i}/d_{ii}^{\alpha}$, we have,

$$
\begin{aligned}
&\frac{N+I+I_r}{P_{r_i}(s_i)} \\
&= \frac{N+I}{P_{r_i}(s_i)} + \frac{I_r}{P_{r_i}(s_i)} \\
&< \frac{1}{\beta+\tau} + \frac{8P_{l_{max}}k}{(1/2)^{\alpha} d_{l_{min}}^{\alpha} \mu^{\alpha}} \cdot \frac{\alpha-1}{\alpha-2} \cdot \frac{1}{P_{r_i}(s_i)} \\
&= \frac{1}{\beta+\tau} + \frac{8P_{l_{max}}k}{(1/2)^{\alpha} d_{l_{min}}^{\alpha} \mu^{\alpha}} \cdot \frac{\alpha-1}{\alpha-2} \cdot \frac{d_{ii}^{\alpha}}{P_{l_i}} \\
&= \frac{1}{\beta+\tau} + \frac{8(c \cdot d_{l_{max}}^{\rho\alpha})k}{(1/2)^{\alpha} d_{l_{min}}^{\alpha} \mu^{\alpha}} \cdot \frac{\alpha-1}{\alpha-2} \cdot \frac{d_{ii}^{\alpha}}{(c \cdot d_{ii}^{\rho\alpha})} \\
&\le \frac{1}{\beta+\tau} + \frac{8(c \cdot d_{l_{max}}^{\rho\alpha})k}{(1/2)^{\alpha} \mu^{\alpha}} \cdot \frac{\alpha-1}{\alpha-2} \cdot \frac{1}{(c \cdot d_{ii}^{\rho\alpha})}
\end{aligned}
$$

$$\leq \frac{1}{\beta+\tau} + \frac{8k}{(1/2)^{\alpha}} \cdot \frac{\alpha-1}{\alpha-2} \cdot \frac{1}{\left(2\left[\frac{8k\beta(\beta+\tau)(\alpha-1)}{\tau(\alpha-2)}\right]^{\frac{1}{\alpha}}\right)^{\alpha}}$$

$$\leq \frac{1}{\beta+\tau} + \frac{\tau}{\beta(\beta+\tau)}$$

$$= \frac{1}{\beta}.$$

Therefore, $\frac{P_{r_i}(s_i)}{N+I+I_r} \geq \beta$.
Thus that, the schedule given by SLS_{pc} is feasible. □

Theorem 3 *The time complexity of algorithm HyperMaxLS is* $O(\Delta_{max})$.

Proof The maximum number of links in a square is Δ_{max}, which means that the maximum number of links in the set $L^{'}$ is Δ_{max}. Any link $l_i \in L^{'}$ executes the algorithm. In the process of implementation, the remaining links need to be determined whether it continues to join a set T_i or not until all links are checked. Then the algorithm returns the set V. Therefore, the time complexity of HyperMaxLS is $\Delta_{max} - 1$, *i.e.*, $O(\Delta_{max})$. □

Theorem 4 *Algorithm* SLS_{pc} *can teminate in* $O(n \cdot \sharp c \cdot \Delta_{max}^2)$. *In the ideal case, the complexity of* SLS_{pc} *is* $O(n \cdot \sharp c)$. *The average time complexity of the algorithm is* $O(n \cdot \sharp c \cdot \Delta_{max}^2/m)$, *where* $m = \lfloor \Delta_{max} \cdot p \rfloor$ *is the expected number of the links in the set* V *returned by the algorithm HyperMaxLS,* $0 < p < 1$ *is the constant and* $\sharp c$ *is the maximum number of squares in a partitioning of in the line* 2 *of algorithm* SLS_{pc}.

Proof The deployment region is divided into squares and all the squares have been 4 -colored. For each of the four colors considered, all the squares of current color are scanned $O(\sharp c)$ times. The time complexity of the algorithm HyperMaxLS is $O(\Delta_{max})$. In the worst case, we have to select only one link in each square since any two links in one square are not scheduled together. Therefore, choosing the links in a square from the first to the last have to execute $O(\Delta_{max})$ times, then the repeat-until loop up to n. Thus, time complexity of the algorithm SLS_{pc} is $O(n \cdot \sharp c \cdot \Delta_{max}^2)$. In the ideal case, the repeat-until loop up to $\lceil n/\Delta_{max} \rceil$ times since all the links in one unit can be scheduled simultaneously, so the time complexity of the algorithm SLS_{pc} is $O(n \cdot \sharp c)$. Therefore, the expected number of the links in the set V returned by HyperMaxLS is $m = \lfloor \Delta_{max} \cdot p \rfloor$. Thus, the time complexity of SLS_{pc} is $O(n \cdot \sharp c \cdot \Delta_{max}^2/m)$. □

Note The time complexity of GOW* is $O(n \cdot \overline{k} \cdot \sharp c \cdot \Delta_{max})$, where $\sharp c$ is the maximum number of squares in a partitioning of the deployment region, and constant $\overline{k}$ is called the SNR diversity of instance G. In this paper, we consider $1 < \overline{k} < \Delta_{max}$ since we mainly select the links in one square. In the worst case, ideal case and the average case, the ratios of the time complexity of our algorithm to GOW* are $O(\Delta_{max}/\overline{k})$, $O(1/(\overline{k} \cdot \Delta_{max}))$ and $O(\Delta_{max}/(\overline{k} \cdot m))$, respectively.

Theorem 5 *The upper bound of the scheduling length computed by algorithm* SLS_{pc} *is* $O(\Delta_{max})$. *The average length of the scheduling is* $\Theta(\Delta_{max}/m)$, *where* $m =$

$\lfloor \Delta_{max} \cdot p \rfloor$ *is the expected number of the links in the set* V *returned by the algorithm HyperMaxLS and* $0 < p < 1$ *is the constant.*

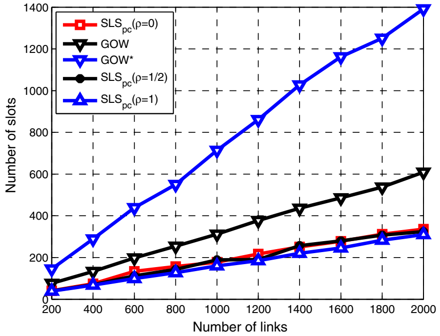
Proof Theorem 2 shows that links whose receivers are in different squares with the same color can be scheduled simultaneously. Firstly, we consider a link from node s to node r in the square S, and assume that r is exactly at the border of maximum transmission range of s, $i.e.$, the SNR at node r is exactly $\beta + \tau$, therefore, there are only one link in the square S is scheduled in one slot. Therefore, all links in the square S must be sequentially scheduled. In this case, the length of scheduling all the links in the square S is Δ_{max}. To multiply the number of colors, the upper bound of the scheduling length computed by SLS_{pc} is $4 \cdot \Delta_{max}$, that is, $O(\Delta_{max})$. Secondly, in general, there are two or more links in the square S are scheduled simultaneously, by implementing HyperMaxLS, the links can be scheduled together. Assume that the excepted number of links in the set V returned by HyperMaxLS is $m = \lfloor \Delta_{max} \cdot p \rfloor$. Thus, the average length of scheduling all the links is $\Theta(\Delta_{max}/m)$. □

Note The scheduling length computed by algorithm GOW* is $O(|C_0| + \Delta_{max})$, where $|C_0|$ is the number of links in class C_0. The upper bound of the scheduling length computed by algorithm SLS_{pc} is $O(\Delta_{max})$. The average scheduling length of SLS_{pc} is $1/m$ of GOW*. We have known that the lower bound on the length of the optimal schedule is $\Omega(\Delta_{max})$ (Blough et al. 2010). The approximation ratio of the optimal schedule is $\frac{O(\Delta_{max})}{\Omega(\Delta_{max})}$ Therefore, we have an approximation ratio with respect to optimal of $O(1)$.

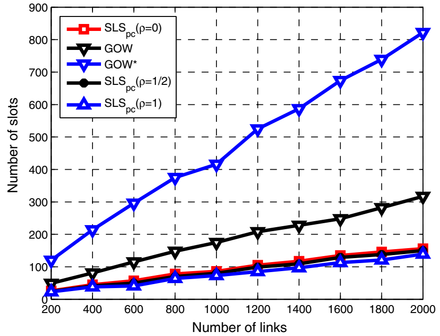
5 Simulation results

In this section, we compare our algorithm SLS_{pc} with the algorithm GOW* proposed by Blough et al. (2010) and GOW proposed by Goussevskaia, Oswald and Wanttehofer in Goussevskaia et al. (2007). Assume that all links are distributed in a 800×800 square region randomly. The number of links that are chosen in simulation is 200, 400, ..., 2000, respectively. We set $\alpha = 3, 4, 5$, the SINR threshold $\beta = 3, 4, 5$, $\tau = 2, 4, 6, 8$, the ambient noise $N = 0.001(mW)$, $\rho = 0$, $1/2$ and 1, $c = P_{l_{max}}$, $P_{l_{max}} = 80(mW)$ is the transmission power of the maximum link length. Suppose that the minimum link length $d_{l_{min}} \geq 1$ since the received power should be little than the sending power. The maximum link length is $P_{l_{max}}/((\beta \cdot N)^{1/\alpha})$. To make the results more realistic, we run simulation 30 times and average the results as the final results.

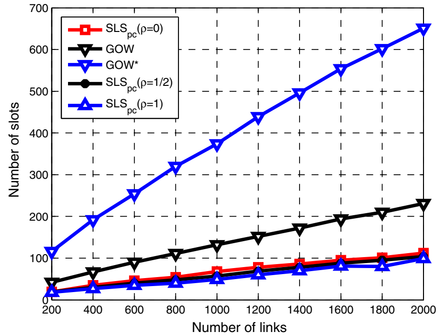
Figure 1 shows the simulation results with $\alpha = 3, 4$ and 5, $\beta = 3$, and $\tau = 2$, respectively. As shown in Fig. 1a–c, SLS_{pc} has a better performance than that of GOW* and GOW. Compared to the number of time slots scheduled all the links in the deployment region, with the increase of α, the scheduled length of those algorithms decrease. It can be also observed with the increase of the number of links, the growth rate of the scheduled length of SLS_{pc} and slower than GOW and GOW*. For the scheduled length, SLS_{pc} has a better performance than GOW* and GOW when the values of β is determined. SLS_{pc} has the similar performance when $\rho = 0$, $1/2$ and 1 for the fixed arguments.



(a) $\alpha = 3$.

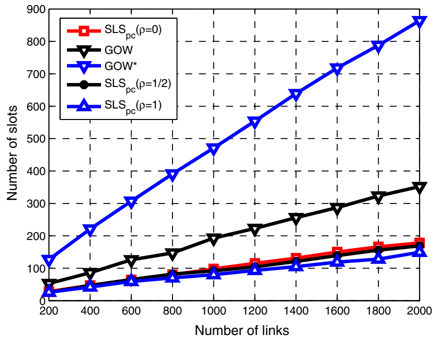


(b) $\alpha = 4$.

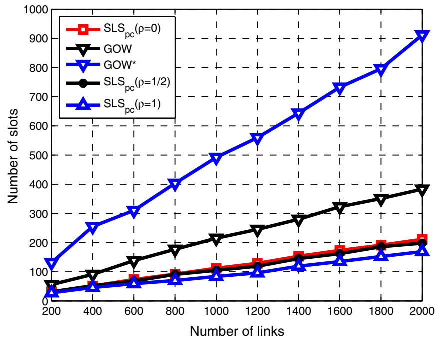


(c) $\alpha = 5$.

Fig. 1 α influences the scheduling performance with $\beta = 3$, $\tau = 2$



(a) $\beta = 4$.



(b) $\beta = 5$.

Fig. 2 β influence the scheduling performance with $\alpha = 4$, $\tau = 2$

As shown in Figs. 1b and 2a, b, we observe that, with the increase in the value of β, the scheduled length of these algorithms increase. The reason is that according to the Sect. 4, we have know that SLS_{pc} has to call a sub-algorithm HyperMaxLS. When the value of τ is determined, with the increase of β, the number of links in

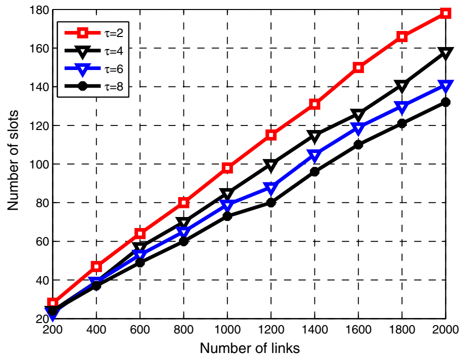


Fig. 3 τ influence the schedule performance with $\beta = 4$, $\alpha = 4$, and $\rho = 0$

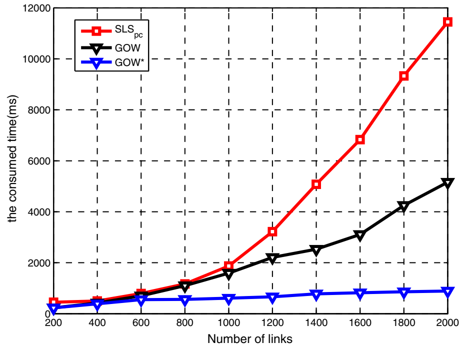


Fig. 4 The consumed times with $\beta = 3$, $\tau = 2$, $\alpha = 4$, and $\rho = 0$

the set V returned by HyperMaxLS decreases slightly, while this contributes to the large scheduled length of SLS_{pc}. Thus, the number of time slots of SLS_{pc} in Figs. 1b and 2a, b increase gradually. Moreover, we also can know that GOW have a better performance than GOW* since GOW neglects the ambient noise and GOW* has to sequentially schedule the *black and gray links*. For the scheduled length, SLS_{pc} has a better performance than GOW* and GOW when the values of α is determined.

For SLS_{pc}, Fig. 3 shows that $\tau = 2, 4, 6$ and 8 with $\beta = 4$, $\alpha = 4$, $\rho = 0$, respectively. We observe that, with the increase in the value of τ, the number of slot scheduled all the links decrease. Since there many more links in one square can be scheduled together with the increase of τ. Therefore, the performance of algorithm SLS_{pc} is going to increase.

Figure 4 presents the consumed times when $\beta = 3$, $\alpha = 4$, and $\rho = 0$. We can know that as the *black and gray links* are *easy to schedule* (Blough et al. 2010), the time complexity of GOW* is lower than GOW. It can also be seen that compared with GOW* and GOW, the time complexity of SLS_{pc} becomes higher. The reason is that we have known SLS_{pc} has to call a sub-algorithm HyperMaxLS in each colored square until all the links in selected square are scheduled. Thus, the time consumed of algorithm SLS_{pc} is higher than GOW* and GOW. This is the problem we have to consider in the further work.

6 Conclusion

In this paper, we present a link schedule algorithm SLS_{pc} with oblivious power assignment under the SINR and hypergraph models, show that the upper bound of the scheduling length constructed by the algorithm SLS_{pc} is $O(\Delta_{max})$, the average length of scheduling is $\Theta(\Delta_{max}/m)$, where $m = \lfloor \Delta_{max} \cdot p \rfloor$ is the expected number of the links in the set V returned by the algorithm HyperMaxLS (Maximal links schedule under hypergraph model) and $0 < p < 1$ is the constant. Compared with the algorithm GOW* in Blough et al. (2010) whose scheduling length is $O(|C_0| + \Delta_{max})$, where $|C_0|$ is number of links in the class C_0 in which the link must be sequentially scheduled. In the worst case, the scheduling length of SLS_{pc} decreases by $|C_0|$, and the average length of scheduling is $1/m$ of GOW*. In the worst case, ideal case and the average case, the ratios of the time complexity of our algorithm SLS_{pc} to that of GOW* are $O(\Delta_{max}/\overline{k})$, $O(1/(\overline{k} \cdot \Delta_{max}))$ and $O(\Delta_{max}/(\overline{k} \cdot m))$ respectively, and $1 < \overline{k} < \Delta_{max}$. Link scheduling algorithm designed with the low time complexity under SINR and Hypergraph models is challengeable for the further study.

Acknowledgments The work was partially supported by National Natural Science Foundation of China for contract 61373027, Natural Science Foundation of Shandong Province for contract ZR2012FM023.

References

Afek Y, Alon N, Barad O, Hornstein E, Barkai N, Bar-Joseph Z (2011) A biological solution to a fundamental distributed computing problem. Science 331(6014):183–185

Blough DM, Resta G, Santi P (2010) Approximation algorithms for wireless link scheduling with SINR-based interference. IEEE/ACM Trans Netw 18(6):1701–1712

Cai Z, Duan Y, Bourgeios A (2015) Delay efficient opportunistic routing in asynchronous multi-channel cognitive radio networks. J Comb Optim 29(4):815–835

Chaporkar P, Kar K, Luo X, Sarkar S (2008) Throughput and fairness guarantees through maximal scheduling in wireless networks. IEEE Trans Inf Theory 54(2):572–594

Chaorkar A, Proutiere P (2013) Optimal distributed scheduling in wireless networks under SINR interference model. arXiv:1305.038 [cs.IT]

Cheng S, Cai Z, Li J, et al. (2015) Drawing dominant dataset from big sensory data in wireless sensor networks. In: Proceedings of the IEEE INFOCOM 2015

Dimakis A, Walrand J (2006) Sufficient conditions for stability of longest queue first scheduling: second order properties using fluid limits. Adv Appl Probab 38(2):505–521

ElBatt T, Ephremides A (2004) Joint scheduling and power control for wireless ad hoc networks. IEEE Trans Wireless Commun 3(1):74–85

Gao J, Li J, Cai Z, Gao H (2015) Composite event coverage in wireless sensor networks with heterogeneous sensors. In: Proceedings of the IEEE INFOCOM 2015

Goussevskaia O, Oswald YV, Wattenhofer R (2007) Complexity in geometric SINR. Proc. ACM MobiHoc 2007:100–109

Goussevskaia O, Wattenhofer R, Hallorsson MM, Welzl E (2009) Capacity of arbitrary wireless networks. Proc. IEEE INFOCOM 2009:1872–1880

Guo L, Li Y, Cai Z (2014) Minimum-latency aggregation scheduing in wireless sensor network. J Comb Optim. doi:10.1007/s10878-014-9748-7

Gupta P, Kumar PR (2000) The capacity of wireless networks. IEEE Trans Inf Theory 46(2):388–404

Halldórsson MM, Mitra P (2011) Nearly optimal bounds for distributed wireless scheduleing in the SINR model. Proc. ICALP 2011:625–636

Hajek B, Sasaki G (1988) Link scheduling in ploynomial time. IEEE Trans Inf Theory 34(5):910–917

He Z, Cai Z, Cheng S, Wang X (2014) Approximate aggregation for tracking quantiles in wireless sensor networks. In: Proceedings of the COCOA 2014. LNCS 8881, pp 161–172

Ji S, Uluagac A, Beyah R, Cai Z (2013) Practical unicast and convergecast scheduling schemes for cognitive radio networks. J Comb Optim 26(1):161–177

Kesselheim T (2011) A constant-factor approximation for wireless capacity maximization with power control in the SINR model. Proc. SODA 2011:1549–1559

Kozat UC, Koutsopoulos I (2006) Cross-layer desgin for power efficiency and Qos provisioning in multi-hop wireless networks. IEEE Trans Wireless Commun 5(11):3306–3315

Kompella S, Wieselthier JE, Ephremides A, Sherali HD (2010) On optimal SINR-based scheduling in multihop wireless networks. IEEE/ACM Trans Netw 18(6):1713–1724

Li D, Zhu Q, Du H, Li J (2014) An improved distributed data aggregation scheduling in wireless sensor networks. J Comb Optim 27(2):221–240

Li Q, Kim G, Negi R (2008) Maximal scheduling in a hypergraph model for wireless networks. Proc IEEE ICC 2008:3853–3857

Li Q, Negi R (2012) Maximal scheduling in wireless ad hoc networks with hypergraph interference models. IEEE Trans Veh Technol 61(1):297–310

Nelson R, Kleinrock L (1985) Spatial-TDMA: a collison-free multihop channel access protocol. IEEE Trans Commun 33(9):934–944

Pei G, Kumar A, Vullikanti S (2012) Distributed algorithms for maximum link scheduling under the physical interference model. Proc DISC 2012:407–408

Pei G, Anil Kumar VS (2012) Efficient algorithms for maximum link scheduling in distributed computing models with SINR contraints. arXiv:1208.0811v2[cs.DC]16, Nov (2012)

Sarkar S, Sivarajan KN (1998) Hypergraph models for cellular mobile communication systems. IEEE Trans Veh Technol 47(2):460–471

Schmidt S, Wattenhofer R (2006) Algorithmic models for sensor networks. In: Proceedings of the IEEE IPDPS 2006

Sharma G, Mazumdar R, Shroff N (2006) On the complexity of scheduling in wireless networks. Proc ACM MobiCom 2006:227–238

Wan P, Xu X, Frieder O (2010) Shortest link scheduling with power control under physical interference model. Proc IEEE MSN 2010:74–78

Wan, P, Wang, L, Ma, C, Wang, Z, Xu, B, L, M (2013) Maximizing wireless network capacity with linear power: breaking the logarithmic barrier. In: Proceedings of the IEEE INFoCOM 2013, pp 135–139