INTERNATIONAL CONFERENCE ON COMBINATORICS, CRYPTOGRAPHY AND COMPUTATION

Iran University of Science &Technology

Proceedings of the 2nd International Conference on Combinatorics, Cryptography and Computation (I4C2017)

Scheduling wireless links in the physical interference model

Neda Mohammadi[a], Mehdi Kadivar[b]
[a,b]Department of Computer Science,
University of Shahrekord, Shahrekord, Iran
[a]Nedamohammadi_66@yahoo.com; [b]M_Kadivar@aut.ac.ir

ABSTRACT

In this paper, we consider a computationally hard problem; Scheduling wireless links for simultaneous activation in such a way that all transmissions are successfully decoded at the receivers and moreover network capacity is maximized. Often this problem is formulated in a spatial time-division multiple access (STDMA) framework. Here we interpret this framework as the finding maximum clique in a graph.

KEYWORDS: link scheduling, physical interference model, maximum clique;

1 INTRODUCTION

In this work, we study the problem of scheduling wireless links in the physical interference model. Let $L=\{l_1,l_2,...,l_n\}$ be a set of n wireless links, where each link l_i represents a communication request from a sender s_i to a receiver r_i. Depending on the spatial disposition of such nodes, activating more than one link simultaneously creates interference that may hamper the receivers ability to decode what they receive. In the physical interference model [1], a receiver r_i successfully decodes a transmission from a sender s_i if and only if

$$SINR(l_i,A)=\frac{\frac{P}{d_{ii}^{\alpha}}}{N+\sum_{\substack{j\in S\\ j\neq i}}\frac{P}{d_{ji}^{\alpha}}}\geq\beta, \qquad (1)$$

Where $A\subseteq L$ is the set of links which are simultaneously active. P is a sender's transmission power (assumed the same for all senders), i.e., we use uniform power assignment scheme. d_{ij} is the Euclidean distance between nodes s_i and r_j. α is a constant path-loss exponent, with typical values in the range $2\leq\alpha\leq 6$. N is the noise floor; and β is the minimum signal-to-interference-and-noise-ratio (SINR)

required for a successful message decoding, we assumed that $\beta > 1$ and has the same value for all links in L.We say that $A \subseteq L$ is feasible if no two of its members share a node and $SINR(l_i, A) \geq \beta$ for all $l_i \in A$.

Several strategies have been devised to maximize network capacity, either through the self-contained scheduling of the links in L for activation or by combining link scheduling with other techniques such as graph-theoretic notions. Often the problem is formulated in a spatial time-division multiple access (STDMA) framework, that is, assuming essentially that time is divided into time slots, each one accommodating a certain number of simultaneous link activations. Now we formally define the problem to be studied in this paper.

Definition 1: A scheduling is represented by $S = \{S_1, S_2, ..., S_T\}$, where $S_t, 1 \leq t \leq T$, is a subset of links in L assigned into time slot t. We say a schedule S is feasible if and only if:

1) each $l_i \in L$ is appeared in exactly one time slot and at most one time in each S_t.

2) for each S_t and each $l_i \in S_t$, $SINR(l_i, S_t) \geq \beta$.

3) every two members of S_t share no node.

For a schedule S, we refer to $|S|$ as schedule length.

The aim of scheduling problem is finding a feasible schedule S^* such that it has the minimum schedule length among all the feasible schedules. This problem can be interpreted as the proper coloring of a graph's vertices so that every vertex gets exactly one color [2]. There is a sense in which this formulation can be interpreted as the context of finding maximum clique in a graph. A clique in an undirected graph is a subset of vertices in which every two vertices are adjacent to each other. The maximum clique problem seeks to find a clique of the largest possible size in a given graph. In the specific case of scheduling the links in L for simultaneous activation, we begin by defining a graph, denoted by C.

Definition 2: Let L be the link list. We introduce the graph $C = (V, E)$, in which $V = L$ and $(l_i, l_j) \in E$ if and only if $\{l_i, l_j\}$ is a feasible set.

Lemma: Finding S^* is equivalent to finding a feasible maximum clique in C.

2 EXACT ALGORITHM FOR FINDING MAXIMUM CLIQUE

The maximum clique problem is a well-Known NP-hard problem [3]. In [4] a exact branch-and-bound algorithm for the maximum clique problem is presented. If we attempted to schedule the links in L by finding the maximum cliques in C iteratively and using each of the resulting cliques as the set of links to be scheduled in each time slot, clearly some clique A might turn up as part of the solution such that $SINR(l_i, A) < \beta$ for some link $l_i \in A$; so we must forbid any maximum clique that is not feasible. We reconstruct the algorithm in [4] to provide a feasible maximum clique for scheduling problem, Algorithm 1. In this algorithm, the n vertices of the input graph $G = (V, E)$ are identified as $\{v_1, v_2, ..., v_n\}$. The set of vertices adjacent to a vertex v_i, the set of its neighbours, is denoted by $N(v_i)$ and the degree of the vertex v_i, the cardinality of $N(v_i)$, is denoted by $d(v_i)$. A key element of this algorithm is that during the search for the largest feasible clique containing a given vertex, vertices that cannot form cliques larger than the current maximum clique are pruned, in a hierarchical fashion. The variables max and *Cm* store the size of the maximum clique and the maximum clique found thus far, respectively. Initially, max is set to be

equal to the lower bound on clique lb provided as an input parameter. It gives the maximum clique when the algorithm is terminated.

```
Procedure MaxClique(G, lb)
   max = lb
   for i : 1 to n do
      if deg(v_i) ≥ max then
         U = ∅
         for each v_j ∈ N(v_i) do
            if j > i then
               if deg(v_j) ≥ max then
                  U = U ∪ {v_j}
         Clique(G, U ,1, {v_i} , {v_i})
```

```
Procedure Clique(G, U , size, S , Cm)
   if U = ∅ then
      if size > max then
         max = size
         S = Cm
      return
   while |U |> 0 do
         if size + |U | ≤ max then
            return
         Select any vertex u from U
         U = U \ {u}
         if Cm ∪ {u} is feasible
            Cm = Cm ∪ {u}
            N'(u) = {w | w ∈ N(u), deg(w) ≥ max}
            Clique(G, U ∩ N'(u) ,size+1, S , Cm)
            Cm = Cm \ {u}
```

Algorithm 1 Algorithm for finding the feasible maximum clique of a given graph $G = (V, E)$.

3 S^* CONSTRUCTION

We now introduce the Algorithm 2 to determine S^*, iteratively. The main idea is finding a feasible maximum clique in C. For this purpose, first S_1 is computed by finding feasible maximum clique in C and then S_2 is computed by finding feasible maximum clique in $C \setminus S_1$ and so on till C become empty. The following is the general outline of the algorithm.

```
Procedure Scheduling(C )
   k =1
   while C ≠ ∅
     S_k = Maxclique(C ,0)
    C = C \ {S_k}
    k = k +1
```

Algorithm 2 Algorithm for finding S^*.

4 CONCLUSION

In this paper, we presented a new algorithm for the scheduling wireless links problem. this problem is interpreted in the context of finding maximum clique in a graph.

5 REFERENCES

1. Gupta, P. and Kumar, P. R. (2000),"The capacity of wireless networks," IEEE Trans. Inf. Theory, 46, pp 388-404.
2. Vieira, FR. J., Rezende, JF. and Barbosa, VC. (2016),"Scheduling wireless links by vertex multicoloring in the physical interference model," Computer Networks, 99, pp 125-133.
3. Garey, M. R. and Johnson, D. S. (1979), "A guide to the theory of NP-Completeness," Computers and Intractability, New York, NY, USA: W. H. Freeman & Co.
4. Pattabiraman, B., Patwary, Md. M. A., Gebremedhin, A. H., Liao, Wk. and Choudhary, A. (2015),"Fast algorithm for the maximum clique problem on massive graphs with applications to overlapping community detection," Internet Mathematics, 11, pp 421-448.