Complexity of Scheduling for Minimum Power on a GMAC

Arun Padakandla
ECE Department
Indian Institute of Science
Bangalore, India
arunpr@ece.iisc.ernet.in

Rajesh Sundaresan
ECE Department
Indian Institute of Science
Bangalore, India
rajeshs@ece.iisc.ernet.in

***Abstract*— Two decision versions of a combinatorial power minimization problem for scheduling in a time-slotted Gaussian multiple-access channel (GMAC) are studied in this paper. If the number of slots per second is a variable, the problem is shown to be NP-complete. If the number of time-slots per second is fixed, an algorithm that terminates in polynomial time is provided.**

I. Introduction and Prior Work

Consider a Gaussian multiple-access channel (GMAC) with K users. User k demands reliable communication at rate $\frac{r_k}{2}$ bits per second. There are N slots every second[1] and each user transmits in at most one slot per second. We consider an overloaded system where $K \geq N$. Let S_n denote the set of users that transmit in slot n. The received signal in slot n is given by

$$Y_n = \sum_{k \in S_n} X_k + W_n$$

where X_k is the information symbol transmitted by user k. The additive noise random variables W_n are independent and identically distributed as $\mathcal{N}(0,1)$. The goal is to schedule users in each of these n slots so that users' rate requirements are met and sum power over all users is minimized.

Clearly, this problem can be posed in other settings as well. For example in a frequency-flat channel, subcarriers pertaining to an OFDM system and codes pertaining to CDMA system play the role of time slots in this paper. Our attention to scheduling in time slots is only to ease exposition.

Let us first focus on one slot, say n. Recall that S_n is the subset of users that transmit in slot n. In order to meet the rate requirements, the sum power of users in this slot should satisfy [1, Chapter 14]

$$\frac{1}{2} r(S_n) := \frac{1}{2} \sum_{k \in S_n} r_k \leq \frac{1}{2} \log \left(1 + \sum_{k \in S_n} p_k \right)$$

so that[2]

$$\sum_{k \in S_n} p_k \geq 2^{r(S_n)} - 1.$$

Furthermore, it is known that the above lower bound on sum power is achieved via a successive cancellation decoder. (See for e.g. [2, Lemma 3.4]). Thus we may assume equality in (1) for a fixed S_n.

Now given a partition $S_n : n \in [N]$, where $[N]$ denotes the set $\{1, 2, \cdots, N\}$, the minimum sum power for the partition is given by

$$\sum_{n=1}^{N} \sum_{k \in S_n} p_k = \sum_{n=1}^{N} 2^{r(S_n)} - N \tag{1}$$

The minimum is over all encoding and decoding schemes for the given partition. We pose the following question: What is the minimum power (1) over all partitions?

This is a combinatorial optimization problem. We address the complexity of the decision versions of this problem in this paper. In Section II, we introduce some notation and relevant concepts from complexity theory. In Section III-A, we show that when N is input as part of the problem instance, the problem is NP-complete, i.e., if this problem can be solved in polynomial time by a deterministic Turing machine, we will have obtained a polynomial time algorithm to several problems that have thus far resisted such solutions. In Section III-B we show that a version of the problem where N is known and fixed can be solved in polynomial time by a deterministic Turing machine.

Algorithms for power allocation to nodes in a network have been studied in different contexts. In a wireless communication setting a node's transmission range being proportional to its transmit power, the topology of the network is a function of the assigned powers. Chen and Huang [3] show the problem of minimizing sum power for full connectivity is NP-hard. Kirousis and others [4] study a simpler problem when n nodes lie along a line separated by unit distance and provide an $O(n^4)$ algorithm for power assignment. The above problems deal with power allocation for connectivity, whereas we focus on power allocation to meet certain rate requirements. Arikan [5] considers the problem of assigning powers to nodes in a packet radio network such that specific origin-destination pairs communicate at specified rates. He proves that the problem is NP-hard. Assuming that the only cause for packets to be received in error is simultaneous transmissions he schedules at most one radio at any time. In contrast we allow for multiple users per slot and compensate for simultaneous transmissions

[1] Second as unit of time has been chosen for simplicity.

[2] Note the disappearance of the factor 2 in the exponent; this is the reason for the rather strange appearance of $\frac{1}{2}$ in the rate requirement.

via larger powers and the use of a successive cancellation decoder.

II. Preliminaries

We begin with some remarks on notation. Recall that for an integer $K \geq 1$, $[K]$ denotes the set $\{1, 2 \cdots, K\}$. For $x, y \in \mathbb{Z}_+$, let $s(x) = \lfloor \log_2 x \rfloor + 1$ represent span of x when represented in binary. Let $p_S(x)$, $p_M(s(x), s(y))$ be the number of steps required to compute 2^x and compute xy respectively, where $p_s(\cdot)$, $p_M(\cdot, \cdot)$, are fixed univariate and bivariate polynomials.

For a problem Π, let domain $D(\Pi)$ denote the set of all valid instances of Π, and $Y(\Pi)$ the set of all yes-instances of Π. Let $\max_\Pi : D(\Pi) \to \mathbb{Z}_+$ map a valid instance I to the magnitude of the largest integer in I, or 0 if no integer occurs in I. Let $\text{Length}_\Pi : D(\Pi) \to \mathbb{Z}_+$ map a valid instance I to the length of its encoding[3].

A problem Π' is a subproblem of Π if $D(\Pi') \subseteq D(\Pi)$ and $Y(\Pi') = D(\Pi') \cap Y(\Pi)$. Note that a problem Π and a restricted domain $D(\Pi') \subseteq D(\Pi)$ define the subproblem Π' uniquely.

Let $p(\cdot)$ be a polynomial. Π_p is a subproblem of Π defined through its domain

$$D(\Pi_p) = \{I \in D(\Pi) : \text{max}_\Pi(I) \leq p(\text{Length}_\Pi(I))\}.$$

We quickly recall some basic complexity concepts. See [6, Chapter 2] for a detailed discussion. Π is said to be in class P if it can be solved by a deterministic Turing machine in polynomial time. Π is said to be in class NP if it can be solved by a non-deterministic Turing machine in polynomial time. We say problem Λ can be reduced to Π in polynomial time if there exists an $f : D(\Lambda) \to D(\Pi)$ that satisfies the following :

1) for all $I \in D(\Lambda)$, $I \in Y(\Lambda) \Leftrightarrow f(I) \in Y(\Pi)$, and
2) given I, $f(I)$ can be computed in time polynomial in $\text{Length}_\Lambda(I)$.

Π is NP-complete if $\Pi \in$ NP and every problem $\Lambda \in$ NP can be reduced to Π in polynomial time.

Definition 1: [6, p.95] A problem Π is strongly NP-complete if there exists a polynomial $p(\cdot)$ such that Π_p is NP-complete. □

Example 2: Consider the following three dimensional matching (3DM) problem.

3DM : Given disjoint sets X, Y, Z and a set $V \subseteq X \times Y \times Z$, is there $V' \subseteq V$ that forms a matching for X, Y, Z ? In other words, does every element of X, Y, Z belongs to exactly one triplet in the matching V'? □

3DM is NP-complete. It is also strongly NP-complete because no integer occurs in its description. □

[3]Any encoding scheme referred to in this paper is a *reasonable* encoding scheme. For a discussion on *reasonable* encoding schemes refer to [6, Section 2.1]

III. Slotted allocation for power minimization

Recall from Section I that the problem of minimizing total received sum power (1) needed to satisfy a set of rate requirements $\frac{r_1}{2}, \frac{r_2}{2}, \cdots, \frac{r_K}{2}$ bits/second reduces to the following combinatorial optimization problem:

Given scaled rates $r_1, r_2, \cdots, r_K$, and $N \leq K$, identify a partition $S_n : n \in [N]$ of $[K]$ that minimizes

$$\sum_{n=1}^{N} 2^{r(S_n)}.$$

We investigate the computational complexity of this problem. Two cases are of interest. In the *variable bandwidth* case the number of slots N per second is a variable that is input as part of the problem instance. In the *fixed bandwidth* case, N is assumed known and fixed. We study the complexity of both these variations by looking at their decision versions.

A. Variable Bandwidth Case

SLOTTED PMIN : Given positive integer rates $r_1, r_2, \cdots, r_K$, number of slots N per second, $N \leq K$, a positive integer power P, is there a partition $S_n : n \in [N]$ of $[K]$ such that

$$\sum_{n=1}^{N} 2^{r(S_n)} \leq P \quad ? \tag{2}$$

□

Our first result is the following.

Theorem 3: SLOTTED PMIN is NP-complete. □

Proof:

1) We first show that SLOTTED PMIN $\in$ NP by providing a polynomial time algorithm to check validity of a certificate partition. We assume without loss of generality that $r_1, r_2, \cdots, r_K$ are in increasing order.

Clearly $P > 2^{r_K}$ and therefore $s(P) > r_K$ is a necessary condition for the existence of a partition $S_n : n \in [N]$ that satisfies (2). We can compare $s(P)$ and r_K in time polynomial in the input size and reject the instance when $s(P) \leq r_K$. Hence we may focus on the instances that satisfy $s(P) > r_K$; these are fortunately instances where the rate values are bounded by the size of the input. We thus have the following algorithm. Let $(1 \ll x)$ denote the left shift operation on 1 to obtain a representation of 2^x.

CheckCertificate$(r_k : k \in [K], N, S_n : n \in [N], P)$:

```
if (s(P) ≤ r_K)
        RETURN Certificate is invalid
else {
        for (n = 1, 2, ..., N){
                r(S_n) = Σ_{k∈S_n} r_k
                P_n = (1 ≪ r(S_n))
        }
        P_tot = Σ_{n=1}^{N} P_n
        if (P_tot ≤ P)
                RETURN Certificate is Valid
        else
```

RETURN *Certificate is Invalid*

}

Since $r(S_n) \leq Kr_K \leq Ks(P)$, number of operations needed to compute P_n (via left shifts) is $p_S(Ks(P))$, and the span of P_n is at most $Ks(P)$. The span of P_{tot} is thus at most $Ks(P)\log N$. This is multiplied by N because of the for loop. Since $N \leq K \leq |I|$, the time needed to compute P_{tot} and compare it with P is thus $\mathrm{O}(|I|^4)$. *CheckCertificate* runs in polynomial time.

2) We next show that a subproblem of a strongly NP-complete problem 4-PARTITION can be reduced in polynomial time to SLOTTED PMIN.

4-PARTITION : Given positive integers $a_1, a_2, \cdots, a_{4N}$ such that $\sum_{k=1}^{4N} a_k = NB$ where B is a positive integer, and $\frac{B}{5} < a_k < \frac{B}{3}$ for every $k \in [K]$, is there a partition $S_n : n \in [N]$ of $[4N]$ such that $a(S_n) = B$ for all $n \in [N]$? □

This is termed 4-PARTITION because if a partition exists, every set in the partition will have exactly 4 elements on account of $\frac{B}{5} < a_k < \frac{B}{3}$. Observe that B and N need not be directly input as part of the problem instance. As 4-PARTITION is strongly NP-complete [6, Theorem 4.3], there exists $p(\cdot)$, a polynomial, such that 4-PARTITION$_p$ is NP-complete.

2a) Consider the transformation $f : D(\text{4-PARTITION}_p) \rightarrow D(\text{SLOTTED PMIN})$ defined as follows

$$\begin{array}{rcl} \text{SLOTTED PMIN} & \leftarrow & \text{4-PARTITION}_p \\ r_k & := & a_k \text{ for } k = 1, 2, \cdots, 4N \\ N & := & N \\ P & := & N2^B \end{array}$$

This is a polynomial time reduction because of the following. For any instance $I \in D(\text{4-PARTITION}_p)$, $B < 5a_k < 5p(\text{Length}_\Pi(I))$. Since $N < \text{Length}_\Pi(I)$, P can be computed in at most $\mathrm{O}(p_M(s(N), 5p(\text{Length}_\Pi(I))))$ proving the polynomial complexity of the reduction.

2b) We now prove $I \in Y(\text{4-PARTITION}_p)$ if and only if $f(I) \in Y(\text{SLOTTED PMIN})$. It is easy to see that is I is a yes-instance for 4-PARTITION with partition $S_n : n \in [N]$, then $f(I)$ is a yes-instance of SLOTTED PMIN with the same partition. In fact equality holds in (2). Conversely, if $S_n : n \in [N]$ is a desired partition for SLOTTED PMIN for a yes-instance $f(I)$, we then have

$$\begin{aligned} P = N2^B & \geq \sum_{n=1}^{N} 2^{r(S_n)} \\ & \geq N \left(\prod_{n=1}^{N} 2^{r(S_n)} \right)^{\frac{1}{N}} \qquad (3) \\ & = N2^B \end{aligned}$$

where (3) follows from the arithmetic mean - geometric mean inequality. Consequently, all inequalities are equalities leading to $r(S_n) = B$ for all $n \in [N]$. Thus $S_n : n \in [N]$ is a desired partition for 4-PARTITION and I is a yes-instance of 4-PARTITION. This proves SLOTTED PMIN is NP-complete. ■

B. Fixed Bandwidth

We now look at the case when the number of slots N is fixed.

N-SLOTTED PMIN : Given positive integer rates $r_1, r_2, \cdots, r_K$, where $N \leq K$, a positive integer power P, is there a partition $S_n : n \in [N]$ of $[K]$ such that

$$\sum_{n=1}^{N} 2^{r(S_n)} \leq P \quad ? \tag{4}$$

□

Our second result is the following.

Theorem 4: N-SLOTTED PMIN $\in$ P. □

Proof: We first show that number of partitions of $[K]$ that need to be checked is polynomial in the size of the input. We then argue that computing $\sum_{n=1}^{N} 2^{r(S_n)}$ for each of these partitions $S_n : n \in [N]$ of $[K]$ can be done in polynomial time. Subsequently, we provide a polynomial time algorithm that solves N-SLOTTED PMIN.

1) Associate the N-length vector $(r(S_n) : n \in [N])$ with the partition $S_n : n \in [N]$ of $[K]$. In order to solve N-SLOTTED PMIN we may focus on partitions whose associated vectors have components with values at most $s(P)$. This is because $s(P) > r(S_n)$ for every $n \in [N]$ is a necessary condition for partition $S_n : n \in [N]$ to satisfy (4). Let $T^{(K)}$ be set of vectors associated with all such partitions. Thus we have $|T^{(K)}| \leq (1 + s(P))^N$.

2) Assume without loss of generality $r_1, r_2, \cdots, r_K$ is in increasing order. Observe that $s(P) > r_K$ is a necessary condition for $T^{(K)}$ to be nonempty. As before we declare *No* if $s(P) \leq r_K$ in polynomial time and therefore focus on those instances with $s(P) > r_K$. As in algorithm *CheckCertificate*, $r(S_n), P_n, n \in [N]$, and P_{tot} can be computed in polynomial time. (See discussion on algorithm *CheckCertificate*).

3) We now provide a dynamic programming algorithm *NSlottedPMIN* to solve N-SLOTTED PMIN. Let e_n denote the unit vector with 1 in the n^{th} component and 0 elsewhere. *NSlottedPMIN* computes $T^{(k)}$, the set of vectors associated with partitions of $[k]$, recursively from $T^{(k-1)}$. The set of vectors obtained by adding r_k to the n^{th} component of vectors in $T^{(k-1)}$ i.e.,

$$\begin{aligned} & T^{(k-1)} \oplus_{s(P)} r_k e_n := \\ & \qquad \left\{ t + r_k e_n : t \in T^{(k-1)}, t_n + r_k \leq s(P) \right\} \end{aligned} \tag{5}$$

are the vectors associated with partitions of $[k]$ with $k \in S_n$. Performing a union over n, of sets in (5) we obtain $T^{(k)}$.

NSlottedPMIN$(r_k : k \in [K], P)$:

if $(s(P) \leq r_k)$

RETURN *No*

```
else {
        T^(0) ← {(0,0,··· ,0)} ⊆ Z_+^N
        for (k = 1,2,··· ,K)
              T^(k) ← ∪_{n=1}^N (T^(k−1) ⊕_{s(P)} r_k e_n)
        for (t ∈ T^(K)) {
              for (n = 1,2,··· ,N)
                    𝒫_n = (1 ≪ t_n)
              P_tot = Σ_{n=1}^N 𝒫_n
              if (P_tot ≤ P)
                    RETURN Yes
        }
        RETURN No
}
```

We now analyze complexity of *NSlottedPMIN*. We observe $|T^{(k-1)}| \leq (1+s(P))^N$ for every $k \in [K]$. Therefore, computing $T^{(k)}$ from $T^{(k-1)}$ requires at most $(1+s(P))^N$ additions and as many comparisons. The values involved in these operations are at most $s(P)$. Therefore $T^{(k)}$ can be computed in polynomial time. Since the components of vectors in $T^{(K)}$ are bounded in value by $s(P)$, computation of $\mathcal{P}_n : n \in [N]$, computation of P_{tot}, and its comparison with P, can all be done in polynomial time. Thus *NSlottedPMIN* runs to completion in polynomial time. ■

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 1991.

[2] D. N. C. Tse and S. Hanly, “Multiaccess fading channels - Part I: Polymatroid structure, optimal resource allocation and throughput capacities,” *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 2796–2815, Nov. 1998.

[3] W.-T. Chen and N.-F. Huang, “The strongly connecting problem on multihop packet radio networks,” *IEEE Transactions on Communications*, vol. 37, pp. 293–295, March 1989.

[4] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, “Power consumption in packet radio networks,” in *Lecture Notes In Computer Science; Vol. 1200 archive, Proceedings of the 14th Annual Symposium on Theoretical Aspects of Computer Science*, Nice, France, 1997.

[5] E. Arikan, “Some complexity results about packet radio networks,” *IEEE Transactions on Information Theory*, vol. 30, pp. 681–685, July 1984.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability - A Guide to the Theory of NP-Completeness*, 1st ed. New York: W.H Freeman and Company, 1979.