Algorithms for Implementing Fair Wireless Power Allocations

Yun Shi and Eric J. Friedman,
School of Operations Research and Industrial Engineering,
Cornell University, Ithaca, NY 14853

***Abstract*— We describe algorithms for fairly allocating power among wireless devices sharing a multiple access channel. This problem can be formulated as one of finding a representation of a point in a contra-polymatroid as a convex combination of a small number of its extreme points. We show that this problem is solvable in polynomial time and then describe a fast algorithms based on the Dantzig-Wolfe decomposition. We also describe a faster algorithm when rates are clustered which combines the Dantzig-Wolfe decomposition with a greedy algorithm. (We note that this greedy algorithm solves an interesting covering problem with balls in the L^1 metric.) We also present new algorithms for computing several fair allocations.**

***Index Terms*— Multi-access channel, power allocation, fairness, game theory.**

I. Introduction

In this paper we consider the problem of fairly allocating power in a multiple access channel. It is well known, that given a set of data requirements there are many possible power allocations. The simplest one is given by successive encoding, in which the devices are encoded according to some order in which each successive device chooses the optimal encoding, conditional on the previous devices encodings [1]. Unfortunately, this mechanism is extremely unfair to devices which are towards the end of the ordering.

However, any convex combination (implemented via timesharing) of these allocations is also feasible (and in fact contains all feasible and efficient allocations). As discussed in [2], [3] these "mixture" allocations can be chosen to achieve a variety of goals.

One important goal is fairness. Recently, several authors have described different fair allocations for the multiple access channel. One approach is based on the idea that groups of devices might jam the channel in order to gain more favorable allocations [4]. An alternative approach comes from networking and is based on performance guarantees (stand alone bounds) and dynamic stability [3]. It is interesting that both of these approaches lead to the "fair share" allocation, which was originally introduced in [5].

Two other well known fair allocation methods are based on max-min fairness and proportional fairness. Another interesting allocation method is based on the Shapley value [4], [6].

In this paper, we consider some practical problems involved in the implementation of any of these methods. As discussed in [2] one could implement any allocation method via a timesharing of different device orderings. Mathematically, this corresponds to taking a convex combination of extreme points of a contra-polymatroid.

However, since there are an exponential number of orderings, for such a method to be practical one needs to use only a small number of orderings in the solution. Our goal in this paper is to develop algorithms which find a small set of orderings for any of the above allocation methods.

It is well known that it is always possible to use a number of orderings equal to the number of devices. This set of orderings can be found in polynomial time; however this requires use of the ellipsoid algorithm which is quite slow in practice. Thus, we also present a practical algorithm based on the Dantzig-Wolfe decomposition. We also develop a faster hybrid Dantzig-Wolfe/greedy algorithm for the case in which the rate demands fall into a fixed number of classes. We note that this greedy algorithm solves an interesting covering problem with balls in the L^1 metric.

Lastly, we note that computing the allocations for several of these methods has not been studied, so as part of the above analysis we have developed algorithms (exact or approximate) which efficiently compute the various fair allocation methods.

II. Wireless Model

We consider a group of devices $N = \{1...n\}$ sharing a single multiple access channel. Each device chooses a data rate, r_i and is allocated a power level p_i sufficient so that the pair of rates and powers (r, p) is feasible. As is well known [1], for fixed r the set of feasible allocations is a contra-polymatroid [2]. $P(r) = \left\{p \mid \forall_{S\subseteq N} \sum_{i\in S} p_i \geq c(\sum_{i\in S} r_i)\right\}$ where $c(x) = e^{2x} - 1$, the inverse channel function where we have chosen units so that the noise level is 1.

Now, as discussed in [2], the extreme points of the contra-polymatroid correspond to orderings of the devices and can be implemented using successive encoding as follows. The first device in the ordering uses an optimal code, assuming there are no other devices. Then the second device treats the first as noise and optimally codes on top of the first device and this process continues through all the devices. The decoder then successively decodes by first decoding device n and then subtracting out its perfectly recovered signal and then continues successively to the 1st device.

Since a contra polymatroid is convex, any non-extreme point in the basepolytope of the polymatroid can be implemented by timesharing among some set of orders.

{yun,friedman}@orie.cornell.edu,www.orie.cornell.edu/∼friedman

We denote an ordering by $\sigma : N \rightarrow N$, and the related extreme point by $\psi(\sigma)$ where $\psi_i(\sigma) = \left[c(\sum_{\{j|\sigma(j)\leq\sigma(i)\}} r_j) - c(\sum_{\{j|\sigma(j)<\sigma(i)\}} r_j)\right]$ and thus our goal is to find a set $\{\sigma^1, \ldots, \sigma^k\}$ such that we can write our preferred allocation p as $p = \sum_{j=1}^{k} w_j \psi(\sigma^j)$ where $w_j \geq 0$ and $\sum_{j=1}^{k} w_j = 1$. Note that we do not explicitly note the dependence of $\psi(\sigma)$ on r except when necessary.

III. Fair Allocation Methods: Definitions and Computation

In this section, we describe several fair allocation methods, both their definitions and efficient algorithms for their computations. Let ϕ denote some fair power allocation.

A vector of powers p is said to be max-min fair if it is feasible and for each agent i, p_i can not be increased without decreasing p_j for some agent j with $p_j < p_i$ while maintaining feasibility. We construct a strongly polynomial algorithm for computing this allocation similar to that in [7].

The proportional allocation rule ϕ^{PR} maximizes $\min_i p_i/r_i$ over all feasible allocations. In our setting, the proportional rule is simply given by $\phi_i^{PR} = \frac{r_i c(|r|)}{|r|}$ [3]. This is easy to compute in strongly polynomial time.

Shapley [8] proposed the value in 1953. The Shapley value is the expectation over extreme points given by $\phi_i = \frac{1}{n!}\sum_{\sigma\in\Sigma} \psi_i(\sigma) \quad \forall i \in N$. We can approximate it in polynomial time using statistical sampling.

In [3], [4] the fair-share (FS) allocation rule is proposed for both fairness and robustness reasons. It is given by the following formula, when $r_1 \leq r_2 \leq \cdots \leq r_n$, $\phi_i(r) = \sum_{j=1}^{i} q_j$ where $q_i = [c(\sum_{j=1}^{i-1} r_j + (n-i+1)r_i) - \sum_{j=1}^{i-1}(n-j+1)q_j]/(n-i+1)$. It is efficient to compute the allocation directly from this formula.

IV. Extreme Point Representations

Our main goal is to find a small set of orders that will give us a fair allocation in their convex hull, as this will allow us to efficiently encode any of the fair allocations. Generically this will require n such orders.

Problem 1 (GCH): Given a rate vector $r \in \Re_+^n$, find a set of orders $\sigma^1, \ldots, \sigma^n$ and weights $w_1, \ldots, w_n \geq 0$ with $\sum_{j=1}^{n} w_j = 1$ such that $\phi(r) = \sum_{j=1}^{n} w_j \psi(\sigma^j)$.

First we note, that (ignoring some issues in rational approximation) that this problem can be solved in polynomial time as shown by [9]; however, this algorithm relies on the ellipsoid method as a subroutine and is too slow to use in practice [10].

In order to solve this problem efficiently, we note that it can be posed as an integer program:

$$\begin{array}{c} \min \sum_{i=1}^{n!} s_i \\ s.t. \;\; \Psi w = \phi, \quad \sum_{i=1}^{n!} w_i = 1 \\ s_i \geq w_i, \quad w_i \geq 0, \quad s_i \in \{0,1\} \end{array}$$

where Ψ is the matrix which has one column $\psi(\sigma)$ for each order σ and note that s_i is the indicator of whether $w_i > 0$.

However, if one solves the linear relaxation of this problem (i.e., does not enforce the integrality constraints) using a simplex like method, the solution will be basic [11] and thus a solution to the Integer program.

The problem with this approach is that the linear program that must be solved has an exponential number of variables. However, we can decompose this problem using the Dantzig-Wolfe decomposition [11] in which case we can use the structure of the polymatroidal constraints to efficiently solve the subproblems. In numerical simulations, we can easily solve problems with ten to twenty devices in fractions of a second on a slow computer.

In practice, one expects to have clusters of demands, where similar devices (or applications) have the same requirements on data rate. We would expect that the number of clusters is not too large, while the number of devices in each cluster could be quite large.

We have developed the following approach to solving such problems efficiently. First, we solve for an allocation by clusters using the Dantzig-Wolfe method described above, then we compute the detailed allocation using a greedy-combinatorial approach. The subproblem that we need to solve is essentially the problem of covering a lattice of points with a set of balls that minimize the sum of the L^1 diameters of the balls. We have developed a greedy algorithm for this problem which computes this in time which is logarithmic in the total number of devices and linear in the number of clusters, so it is quite efficient in practice. The number of vertices it gives is linear in the number of devices and exponential in the number of clusters. For example, in simulations we can typically solve problems with 5 clusters and hundreds of users in a fraction of a second.

Acknowledgments

We would like to thank Sergio Servetto and Eva Tardos for helpful comments. Work supported in part by National Science Foundation Grant No. ANI–9730162.

References

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

[2] D. Tse and S. Hanly, “Multi-access fading channels: Part i: Polymatroid structure, optimal resource allocation and throughput capacities,” *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796–2815, 1998.

[3] E. J. Friedman, “Fairness and stability in allocating wireless bandwidth,” 2003, technical Report, Cornell ORIE.

[4] R. La and V. Anantharam, “A game-theoretic look at the gaussian multiaccess channel,” 2002, to appear in the proceedings of the March 2003 DIMACS workshop on Network Information Theory.

[5] S. Shenker, “Making greed work in networks: A game-theoretic analysis of switch service disciplines,” *IEEE/ACM Transactions on Networking*, vol. 3, pp. 819–831, 1995.

[6] M. Shubik, “Incentives, decentralized control, the assignment of joint costs and internal pricing,” *Management Science*, vol. 8, no. 3, pp. 325–343, 1962.

[7] J. Kleinberg, E. Tardos, and Y. Rabani, “Fairness in routing and load balancing,” in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 1999, p. 568.

[8] L. Shapley, “A value for n-person games,” in *Contributions to the theory of games*, H. Kuhn and A. Tucker, Eds. Princeton, NJ: Princeton University Press, 1953.

[9] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer-Verlag, 1988.

[10] R. G. Bland, D. Goldfarb, and M. J. Todd, “The ellipsoid method: A survey,” *Operations Research*, vol. 29, no. 6, pp. 1039–1091, 1981.

[11] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1997.