Fairness and Optimality in Congestion Games

Deeparnab Chakrabarty* Aranyak Mehta† Viswanath Nagarajan‡ Vijay Vazirani§

ABSTRACT

We study two problems, that of computing social optimum and that of finding fair allocations, in the congestion game model of Milchtaich[8] Although we show that the general problem is hard to approximate to any factor, we give simple algorithms for natural simplifications. We also consider these problems in the symmetric network congestion game model [11, 4], and show hardness results and approximate solutions.

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

Keywords

Fairness, Congestion games, Nash equilibrium

1. INTRODUCTION

Congestion games are a special class of non-cooperative games first introduced by Rosenthal [10]. In this setting, the cost faced by a player employing a certain strategy is determined only by the number of other players who employ the same or overlapping strategies. Rosenthal showed that if the cost function is same for all the players, then these games possess a rich structure, in particular they always have a Nash equilibrium in pure strategies. In [8] Milchtaich extended the definition to allow player-specific cost functions, i.e. when different players have different costs at the same congestion, and showed that even these games have a pure Nash equilibrium, if the strategies are not allowed to overlap.

Many real life problems like that of load balancing, bandwidth allocation, network routing etc. can be modelled as congestion games of some sort. In such settings, there may be a central authority who decides the allocations in order to optimize certain global costs. This is the motivation for our paper.

REMARK: In this paper, we shall denote strategies as bins. When we say a player is in a bin, we imply that the player is employing the corresponding strategy. We call this the bin-player model

In this paper, we look at these games (in Milchtaich's setting) from a *centralized* viewpoint. One of the problems that we study is of *social optimality.* In this, we wish to assign strategies to players such that the total cost of all the players is minimized. The other problem is that of finding *fair allocations.* We consider the standard model of minmax fairness model [5, 7]. We call an allocation of strategies to players *minmax fair* if the cost of any player cannot be decreased without increasing the cost of a player who was facing an already higher cost. Its easily seen that such an allocation minimizes the maximum cost faced by a player in an allocation, and hence the name.

We observe that in these settings, the problems of social optimum and fairness are harder than that of finding Nash equilibria. While Milchtaich [8] gave a polytime algorithm for finding pure Nash equilibria in the bin-player model, we show that its hard to approximate both the social optimum and the minmax cost to any factor (Theorem 4.1).

A congestion game model that has recently come under study is the network congestion game ([11, 4]). In the single commodity model the congestion game is on a single source-sink network with all players assumed to be at the source. The paths from source to sink are strategies, and number of players using an edge denotes the congestion on that edge. The cost faced by a player in this case is the sum of the costs of edges in his path. Fabrikant etal. [4] gave a polynomial time algorithm to compute the Nash equilibrium. We investigate the problem of finding social optimum and fair allocations when the costs are linear.

Our Results

The results of this paper are

- We show that the problem of computing fairness and social optimum, is $\mathcal{NP}$-hard in the general model. Our

*College of Computing, Georgia Tech. Email: deepc@cc.gatech.edu

†College of Computing, Georgia Tech. Email: aranyak@cc.gatech.edu

‡Tepper School of Business, CMU Email:viswa@cmu.edu Supported by NSF ITR grant CCR-0122581 (The ALADDIN project)

§College of Computing, Georgia Tech. Email: vazirani@cc.gatech.edu. Supported by NSF Grants 0311541 and 0220343

EC'05, June 5–8, 2005, Vancouver, British Columbia, Canada.

reduction also shows that the problem is hard to approximate to any factor. However, we note that if the number of bins is a constant, then there is a polynomial time algorithm to solve both the problems of fairness and social optimal. (Section 4)

- If all the strategies/bins are similar, which we call the *Symmetric Bins* case, then we give an algorithm to find the fair allocation under certain restrictions. However, the social optimality problem appears hard, although we were not able to prove a hardness. Nevertheless, we show that if the cost functions satisfy certain properties (e.g. if they are linear) then again, there is an algorithm to find the social optimal (Section 2).
- We show that if all the users are similar, facing same costs at same congestions, then its very simple to find both the social optimal and fair allocations. We shall call this case as the *Symmetric Players* case. We show that the problem of finding fair allocations is hard in the Symmetric Network Congestion games, even when the costs are linear. We then go on to give approximate solutions. In particular, we show that the social optimum, which can be found efficiently by a simple modification of the [4] algorithm, is a 3-*prefix-sum* approximation to the fair solution, when the costs are linear (Section 3).

Related Work

Congestion games were first studied by Rosenthal [10] where he showed that if all players faced similar costs (symmetric players case), then there exists a pure Nash equilibrium. Milchtaich [8] extended the result to the case with general cost functions but non-overlapping strategies. In fact, the Nash Equilibrium can be found in polynomial time from any given allocation. There has also been some recent work on computing the equilibrium in the incomplete information model [2].

In computer science, the most common congestion games to be studied are the network routing problem [6, 12, 4] and the load balancing problem [5, 1]. There has also been some research on modelling bandwidth allocation in P2P systems as congestion games [13]. The social optimal in congestion games was also studied by Milchtaich [9], who showed that under certain constraints on the congestion function, there is a socially optimal Nash equilibrium.

Notations and Problem Statement

We are given k bins (strategies) and n players who are to be associated with these bins. Each player i has a cost $c_{i,j}(l)$ when he is in the j^{th} bin with l players (including himself) in that bin. The function $c_{i,j}$ is non decreasing in the congestion. In its whole generality, the input to this problem can be represented as a three dimensional $n \times k \times n$ matrix A, where $A[i,j,l] = c_{i,j}(l)$. An *allocation* is an assignment of players to bins. Given an allocation, the *congestion vector* is a k-dimensional vector where the j^{th} coordinate represents the number of players in bin j. The *cost allocation* vector C is an n-dimensional vector with each coordinate representing the cost faced by that player in the allocation. We will assume, by renaming players, that this vector is non-increasing in its coordinates.

An allocation is called *social optimal* if it minimizes the objective function $\sum_i C[i]$ over all allocations. We denote the value achieved by the social optimum as OPT. A allocation is is the fair, if its cost allocation vector is the lexicographically smallest one. That is its first coordinate, which is also the largest cost, is as small as possible; given that, the second largest cost is as small as possible, and so on. We call the largest cost faced by a player in a fair allocation the *minmax cost* and denote it as OPT'.

2. SYMMETRIC BINS

When all the bins are alike, then the input can be represented as an $n \times n$ matrix, $Q[l,i]$ giving the cost faced by the i^{th} player when he faces congestion l. We also call this matrix the *congestion matrix*. We first show an efficient algorithm to compute the fair allocation, when all the entries of Q are distinct. To do this we shall first show how to find the minmax cost, which is the first coordinate of the cost allocation vector of the fair allocation. Then we show how to iterate the same procedure to get the fair allocation. In the next subsection we shall describe an algorithm to get the social optimum under certain restrictions.

2.1 Fair Allocations

We make a few observations about the congestion matrix Q. Firstly note that all its columns are nondecreasing (since the congestion function is nondecreasing). Secondly, since all entries of Q are distinct, the fair allocation is unique. Thus to get the minmax cost, we look at the entries of the congestion matrix in increasing order, and check if there exists a *feasible* allocation with minmax cost as that entry, and we stop once we get a feasible allocation. We now describe the algorithm and the feasibility subroutine in a little detail.

Sort the entries of Q in ascending order: $M_1 \to M_{n^2}$. For each M in this range, let c_i^M be the maximum congestion the player i can face with its cost being less than M. That is $Q[c_i^M, i] \leq M$ but $Q[c_i^M + 1, i] > M$. Let $feasibility(c_1, \cdots, c_n)$ be a function which returns an allocation of n players in the minimum number of bins with the constraint that i faces congestion atmost c_i. Thus
$OPT' = \min\{M | feasibility(c_1^M, \cdots, c_n^M)$ returns an allocation in less than k bins. $\}$
What remains is the description of $feasibility$.

We show that a simple greedy strategy works for $feasibility$: allocate players to bins of as high congestion as possible. Firstly note that we may assume $c_1 \geq c_2 \geq \cdots \geq c_n$, by renumbering players. The algorithm first places player 1 in bin 1. It then continues placing each player i according to the rule: If c_i is greater than the number of players in the current bin, add i to the current bin; otherwise open a new bin to contain player i, and this becomes the current bin. Thus the allocation satisfies the property that each i faces congestion atmost c_i. Let r be the number of bins used by this algorithm, and ALG be the allocation obtained. Let the bins be numbered in the order in which they were opened. Let N_j be the set of players in bin j, and $n_j = |N_j|$ its congestion. We show that r is the minimum number of bins in which all the players can be accommodated, which proves the correctness of $feasibility$. The following claims follow from the algorithm.
Claim 1: If player $l \in N_j$, then $n_{j-1} \geq c_l \geq n_j$
Claim 2: $n_1 \geq n_2 \cdots \geq n_r$

Lemma 2.1. *The minimum number of bins required to allocate the players given their maximum congestions is r.*

Proof: Suppose the allocation with the minimum number of bins, OPT has $r' < r$ bins. Let the bins in OPT be numbered in decreasing order of congestion. Let the set of players in the bins be $M_1, \cdots, M_{r'}$, and $m_j = |M_j|$. Since the number of players in both OPT and ALG is the same, and the number of bins in OPT is less than in ALG, there must be some i such that $\sum_{j=1}^{i} m_j > \sum_{j=1}^{i} n_j$. That is, number of players in the first i bins in ALG is less than the number of players in the first i bins in OPT. Choose the first such i; note that $m_i > n_i$. Now consider any player $l \in M_{i'}$ where $i' \leq i$. Since in OPT, l faces congestion $m_{i'}$, $c_l \geq m_{i'} \geq m_i > n_i$. Thus from Claims 1 & 2 we have that l must be in one of the first i bins of ALG. Thus all the players in the first i bins in OPT must be in the first i bins of ALG, which contradicts the choice of i. ■

Our algorithm to compute the fair allocation runs in n rounds. In the first round, we compute the smallest entry M_1 in Q such that there is an allocation with each player facing cost at most M_1 (note $M_1 = OPT'$), by running the above algorithm. Since all entries in Q are distinct, this cost is faced by a *unique* player p at a unique congestion c. We *freeze* p at congestion c. In the next round, we find the smallest entry $M_2 \leq M_1$ in Q such that there is an allocation with p facing M_1, and all other players (non frozen) facing cost at most M_2. We know that such an allocation exists because the allocation obtained in round 1 is one such. Computing the allocation in round 2 is similar to computing the minmax allocation (round 1); the difference being that now in all calls to $feasibility$, c_p is always fixed at c. Again there is a unique player (and his congestion) who faces cost M_2, and we freeze this player at his congestion. Now we proceed to the next round. This way at the end of n rounds we would have frozen all the players. It is not hard to see that the allocation returned by our algorithm minimizes the maximum cost, conditioned on that minimizes the second maximum, and so on. By a careful implementation of this, we get

Theorem 2.2. *In the case of Symmetric Bins when all the costs are distinct, the fair allocation can be found in time $\tilde{O}(n^2)$*

We point out that this algorithm does not generalize to the case when the costs are non distinct. This is because with non distinct costs, the algorithm is unable to decide which entry corresponds to the minmax cost.

However, we note that this problem can be reduced to finding the social optimal allocation of a different congestion game. Firstly note that, to get the fair allocation we don't need the exact matrix entries but just the relative order between them. Thus these can be assumed to be from $\{1, 2, \cdots, n^2\}$. We create a new congestion matrix Q' such that $Q'[i,j] = (n+1)^{Q[i,j]}$. Note that the entries in Q' are polynomial in n. Its easy to check that the social optimal allocation for Q' is a fair allocation for Q. As we shall see in the next subsection, the social optimal allocation can be found in special cases. Thus even with non distinct entries, the fair allocation can be found in some special cases (eg: when the costs are linear with congestion).

2.2 Social Optimum in Symmetric Bins

In this section we shall look at the problem of finding the social optimum in the case of symmetric bins. We leave open the question of hardness in this setting. Here we present a polynomial time algorithm for a special case of this problem. An $n \times n$ matrix Q is *anti-Monge* if:

$$\forall r_1 < r_2, \forall c_1 < c_2 : Q[r_1, c_2] + Q[r_2, c_1] \leq Q[r_1, c_1] + Q[r_2, c_2]$$

We present a dynamic programming based algorithm to compute the social optimum in polynomial time if the congestion matrix Q is anti-Monge. Since there are efficient algorithms to rearrange the columns of a matrix to make it anti-Monge (if possible) [3], this algorithm is applicable to a fairly large class of matrices. For example, if the cost faced by each player is a linear function of his congestion, its easy to see that Q is anti-Monge and thus social optimum is computable in polynomial time. A limited class of polynomial functions can also be represented by anti-Monge matrices.

Monge and anti-Monge matrices have very rich structures, and many hard problems (e.g. TSP) have efficient algorithms if the underlying input matrix is Monge or anti-Monge. [3] is an excellent survey, and indeed the following lemma can be easily derived from the result about the Northwest corner rule for Monge matrices. (Theorem 3.1, [3]).

Lemma 2.3. *If the congestion matrix Q is anti-Monge, and the number of players in each bin is $n_1 \geq n_2 \geq \cdots \geq n_k$ s.t. $\sum_{j=1}^{k} n_j = n$, the cheapest allocation assigns players as follows - players 1 to n_1 to bin 1, players $n_1 + 1$ to $n_1 + n_2$ to bin 2, and so on.*

Proof: We associate any allocation with an n dimensional vector, where the ith entry is the congestion faced by player i under this allocation. Let α be the allocation stated in the lemma : players 1 to n_1 at congestion n_1, players $n_1 + 1$ to $n_1 + n_2$ at congestion n_2, and so on. Let β be the cheapest allocation under the specified bin congestions. Note that by fixing the congestion of each bin, the number of players at each congestion level is also fixed. So allocations α and β have the same number of players at each congestion level.

Suppose $\alpha \neq \beta$. Let i_1 be the last player that has different congestions in α and β. *i.e.* $\alpha[i] = \beta[i]$ for all $i > i_1$. Choose β among all the cheapest allocations such that i_1 is as small as possible. If there is no such player, $\alpha = \beta$ and the lemma is true. Otherwise, let $c_1 = \alpha[i_1]$ ($c_2 = \beta[i_1]$) be the congestion level of player i_1 in allocation α (β). Note that $c_2 > c_1$: if $c_2 < c_1$, the number of players at congestion c_2 in allocation β is greater than in α. Among players $i_1 \cdots n$, allocation β has one less player at congestion c_1 than allocation α. So there is a player $i_2 < i_1$ such that $\beta[i_2] = c_1$. Now consider the allocation β' obtained from β by interchanging the places of players i_1 and i_2.

$$\beta'[i] = \begin{cases} \beta[i_2] & i = i_1 \\ \beta[i_1] & i = i_2 \\ \beta[i] & otherwise \end{cases}$$

The difference in the total cost of β' and β is $cost(\beta') - cost(\beta) = Q[c_2, i_2] + Q[c_1, i_1] - (Q[c_1, i_2] + Q[c_2, i_1]) \leq 0$, from the anti-monge property. But the last point of difference between allocations β' and α is a player smaller than i_1, contradicting the choice of β. ■

We will now see how this lemma implies a dynamic programming algorithm to compute OPT. Let us define the table $T[i, n_{max}, r]$ as the cheapest allocation of players i through n, into r bins such that the congestion in any bin is at most n_{max}. Here $i \in [1, n+1]$, $n_{max} \in [1, n]$, and $r \in [0, k]$; so this table has size $O(n^2 k)$. We want to obtain $OPT = T[1, n, k]$.

From the lemma above, if we wish to optimally assign players i through n to r bins such that the maximum congestion is exactly c, players i to $i+c-1$ occupy one bin, and the remaining players ($i+c$ to n) are optimally assigned to $r-1$ bins with congestion at most c in each bin. Thus we have :

$$T[i, n_{max}, r] = \min_c \{ \sum_{i \le p \le i+c-1} Q[c, p] + T[i+c, c, r-1] \}$$

where the minimum is over all $c : 1 \le c \le \min(n_{max}, n-i+1)$. Note

$$T[i, n_{max}, 0] = \begin{cases} 0 & i = n+1 \\ \infty & otherwise \end{cases}$$

An implementation of these recurrences would take $O(n^2)$ time at each step, and $O(n^4 k)$ time overall, to compute OPT. Thus we have

Theorem 2.4. *If the congestion matrix is anti-Monge, then the social optimum can be computed in $O(n^4 k)$ time.*

3. SYMMETRIC PLAYERS AND NETWORK CONGESTION GAMES

In the bin-player model, when all players are symmetric, then both the problems of finding social optimum and fair allocations become simple and can be solved by dynamic programming. Firstly note that now the input is represented by a $n \times k$ matrix P, where $P[l, j]$ is the cost for using the j^{th} bin with (any, since now they are all similar) l people. Let the (distinct) bins be $\{a_1, \cdots, a_k\}$ Let S be an $n \times k$ matrix such that $S[l, r]$ be the optimal allocation for l players in bins a_r to a_k. Note that $S[l, k] = P[l, k]$ for all l and

$$S[m, j] = \min_l \{ S[m-l, j+1] + P[l, j] \}$$

Noting that the social optimum is $S[n, 1]$, we see that the social optimum for symmetric people can be solved in time $O(n^2 k)$ by dynamic programming. The fair allocation is similar, where we are interested in the lexicographic order rather than the sum.

As noted in the introduction, the symmetric network congestion game is a generalization of the symmetric player congestion game in the bin-player model. In the network model, players wish to travel from the source to the sink on paths (which correspond to strategies), and each edge has a cost function associated with it, which is increasing with congestion. Note that this is a much more succinct way of representing the strategies, and this makes the problem more interesting. We describe the model in brief.

In the symmetric case, there is a network G with a single source and destination, and n *identical* players that need to be routed from the source to destination. Every arc e in G has a cost function $c_e(l)$, in terms of the congestion l on arc e. We restrict ourselves to the case when the edge costs $c_e(l) = a_e l$ are linear in l.

A modification of the [4] [1] algorithm gives us the social optimum in polynomial time for single commodity network congestion games when the congestion functions are linear (actually convex). As in [4], we replace each edge e in the network by n parallel edges, where n is the number of players. The costs on these edges are $c_e(1), 2c_e(2) - c_e(1), \cdots, nc_e(n) - (n-1)c_e(n-1)$ and the capacity is 1. Note that since c_e is convex, the above sequence is increasing. Thus if k players use this ensemble of edges, the minimum total cost paid will be $kc_e(k)$ which is the cost paid if k players use the edge e in the original network. Thus a min-cost flow of this new network would give us the social optimum.

In contrast, as we show below, the problem of finding the minmax cost, and thus the fair allocation, is $\mathcal{NP}$-hard, even when the cost functions are linear functions of the congestion.

Theorem 3.1. *Calculating the minmax cost in single commodity symmetric network congestion games with linear costs is $\mathcal{NP}$-hard*

Proof Sketch: We reduce 3-partition to computing the minmax cost. In 3-partition, we are given a set of positive integers $A = \{a_1, \cdots, a_{3m}\}$ and an integer B. We want to determine if we can partition A into m parts of 3 elements each, such that the total of each part is exactly B. We may assume that the a_i's satisfy $\sum_{i=1}^{3m} a_i = mB$ and $\frac{B}{4} < a_i < \frac{B}{2}$.

The instance G that we construct has a path from the source s to destination t with one arc for each $a_i \in A$ - the arc e_i corresponding to a_i has a cost function $c_{e_i}(l) = a_i \cdot l$. In addition each arc e_i has another arc parallel to it with cost function $c(l) = a \cdot l$, where a is chosen appropriately (call this the zero arc). There are m players to be routed from s to t. Let $\lambda = B + 3(m-1)^2 a$. We show that 3-partition has a solution *iff* there is a routing in G where each player faces cost at most λ.

If 3-partition has a solution, player $p[1 \cdots m]$ uses arc e_i iff the partition p contains element a_i; and the zero arc otherwise. This is clearly a routing where each player faces cost λ.

Suppose there is a routing with each player facing cost at most λ. We first show that the social optimal routing sends exactly one unit of flow on each e_i and $m-1$ units on each zero arc. Note that we can account for the cost of each link i, that is, arc e_i and its corresponding zero arc, separately. If $\frac{a_i}{2m-1} < a < \frac{3a_i}{2m-3}$, the cost of this link is minimized precisely when one unit of flow is sent on e_i and $m-1$ units are sent on the zero arc. Since $\frac{B}{4} < a_i < \frac{B}{2}$ there is a non empty interval of possible values of a.
Now the social optimum value OPT=$m\lambda$. Any routing with each player facing cost $\le \lambda$ has total cost at most $m\lambda$=OPT. By the preceding observation such a routing must sent one unit of flow on each e_i and $m-1$ units on each zero arc. Each player in this routing must face cost exactly λ - the cost of any player is the sum of some subset of A and some zero arcs (each of cost $a(m-1)$). If we choose a to be a rational with sufficiently large reduced form, each player is restricted to take exactly 3 e_is and $3m-3$ zero arcs. This in turn implies a solution to 3-partition. ■

Claim 1. *In a network congestion game with linear costs, maximum cost faced by a player in the social optimum is at most 3 times the minmax cost.* [2]

[1] [4] show that the Nash Equilibria for these games can be calculated efficiently

[2] A similar result for the nash routing in the nonatomic setting is implicit in [11].

Proof: Let $O = \{O_1 \geq \cdots \geq O_n\}$ denote the cost allocation vector of the social optimum. Let P denote the path of the player with maximum cost O_1, and P' the path of the player with minimum cost O_n. Also let l_e be the congestion on edge e in the social optimal routing. We show that $O_n \geq \frac{O_1}{3}$. If not, consider a different routing by shifting one unit of flow (the player facing maximum cost) from the path P to the path P'. The resulting increase in total cost is

$$\sum_{e\in P'\setminus P} a_e[(l_e+1)^2 - l_e^2] + \sum_{e\in P\setminus P'} a_e[(l_e-1)^2 - l_e^2]$$

$$\begin{aligned}
&\leq \sum_{e\in P'} a_e[(l_e+1)^2 - l_e^2] + \sum_{e\in P} a_e[(l_e-1)^2 - l_e^2]\\
&= \sum_{e\in P'} a_e(2l_e+1) - \sum_{e\in P} a_e(2l_e-1)\\
&\leq \sum_{e\in P'} 3a_e l_e - \sum_{e\in P} a_e l_e\\
&= 3O_n - O_1\\
&< 0
\end{aligned}$$

where the third inequality follows since $l_e \geq 1$ for $e \in P \bigcup P'$.

Now if M denotes the minmax cost, the total cost of such a routing is at most nM. Thus we have $nM \geq \sum_{i=1}^n O_i \geq \sum_{i=1}^n \frac{O_1}{3} = n\frac{O_1}{3}$. Thus we have the claim. ■

We now show that the social optimum is also a good approximation to the fair solution. We use the notion of *prefix-sum approximation* used commonly in settings of approximate fairness [5, 7]. An nondecreasing vector X is an α-prefix-sum approximation to another nondecreasing vector Y, if each of prefix sums of X is within an α multiplicative factor of that of Y.

Theorem 3.2. *In a network congestion game with linear costs, the social optimum is a 3-prefix-sum approximation to the fair allocation.*

Proof: Let $F = \{F_1 \geq \cdots \geq F_n\}$ and $O = \{O_1 \geq \cdots \geq O_n\}$ denote the cost allocation vectors of the minmax fair allocation and the social optimum respectively. We have to show that for all $i = 1$ to n : $\sum_{j=1}^i O_j \leq 3\sum_{j=1}^i F_j$.

Let $1 \leq h \leq n$ be the first coordinate (if any) where $O_h > 3F_h$. If there is no such h, O is clearly a 3-approximation (in fact, coordinate-wise). We have the following cases:

- $1 \leq i < h$. Here $\sum_{j=1}^i O_j \leq 3\sum_{j=1}^i F_j$, coordinate-wise.
- $h \leq i \leq n$. Here $F_i \leq F_h < \frac{O_h}{3} \leq \frac{O_1}{3}$, so $\sum_{j=i+1}^n F_j < (n-i)\frac{O_1}{3}$. Also, $O_n \geq \frac{O_1}{3}$, hence $\sum_{j=i+1}^n O_j \geq (n-i)\frac{O_1}{3}$. So,

$$\begin{aligned}
\sum_{j=1}^i O_j &= \sum_{j=1}^n O_j - \sum_{j=i+1}^n O_j\\
&\leq \sum_{j=1}^n O_j - (n-i)\frac{O_1}{3}\\
&< \sum_{j=1}^n O_j - \sum_{j=i+1}^n F_j\\
&\leq \sum_{j=1}^n F_j - \sum_{j=i+1}^n F_j\\
&= \sum_{j=1}^i F_j
\end{aligned}$$

where the second last inequality follows from the fact that O corresponds to the social optimum.

Remark: In a similar fashion it can be shown that the Nash equilibrium, which can be found using [4], is a 4-prefix sum approximation to the fair allocation.

4. GENERAL CASE : COMPUTATIONAL COMPLEXITY

We shall show that even if the values of the matrix are restricted to $\{1, \infty\}$, the general problem is $\mathcal{NP}$-hard. The corresponding decision problems for the given optimization problems can be stated thus:

Input: A $n \times k \times n$ $\{1, \infty\}$-matrix A which corresponds to a congestion game , $t \in \mathbb{N}$.

$\mathcal{OPT}$: Is there an allocation of the n-players to the k-bins such that the total cost $OPT \leq t$?

$\mathcal{OPT}'$:Is there an allocation of the n-players to the k-bins such that the maximum cost faced by any player, $\leq t$? (Note that there are only two values possible)

Theorem 4.1. *$\mathcal{OPT}$ is $\mathcal{NP}$-hard*

Proof Sketch: Reduction from 3-SAT.

Given an instance of 3SAT consists of m clauses of 3 literals each, and a total of r variables, we construct the following instance of a congestion game. Let there be $k = 2r$ bins, corresponding to the variables and their negations. Call the bins $\{x_1, \cdots, x_r, \overline{x_1}, \cdots, \overline{x_r}\}$.

For each variable x_i, we have $2m$ dummy players. These players are all identical and have cost 1 (at every congestion) in the bins x_i and $\overline{x_i}$. In bins of other variables, these dummy players face *infinite* cost irrespective of congestion. (Thus in any allocation, the dummy players are restricted to their two bins)

We also have clause players corresponding to each clause in 3SAT. The clause player p_c for $c = \overline{x_i} \vee x_j \vee x_k$ has infinite cost in all bins except $\overline{x_i}$, x_j and x_k, irrespective of congestion. In these three bins, player p_c has a cost of 1 upto congestion m and ∞ for higher congestions. Thus the total number of players is $n = 2m \cdot r + m$. We then set $t = n$.

If the formula is satisfiable, for true variables x_i place its dummy players in the bin $\overline{x_i}$, and similarly for false variables put the dummies in bin x_i. Since the formula is satisfiable, each clause has a true variable, place the clause player in the true variable's bin. It is not hard to see that all players, dummies and clause, all pay cost 1, thus making the total cost n.

On the other hand, if the total cost is n (note it can't be less), then each player must pay cost 1. This implies clause players face congestion at most m. Choose all bins which contain clause players, and set the corresponding variables true. To see that a bin and its "complement" bin, both cannot contain clause players, note that of the $2m$ dummy players atleast one of them would contain atleat m. To see that the assignment is satisfying, note that each clause player is assigned a bin, implying that variable is true. ■

Corollary 1 $\mathcal{OPT}'$ is $\mathcal{NP}$-hard.

Corollary 2 $\mathcal{OPT}$, $\mathcal{OPT}'$ are hard to approximate to any finite factor, unless $\mathcal{P} = \mathcal{NP}$.

Proof: Note that in this reduction, we only generate congestion game instances where the players face costs of 1 or ∞. So any approximation algorithm will also serve as an exact algorithm - if the approximate solution has a finite cost then $OPT = n$, otherwise $OPT = \infty$. So this reduction also proves that we cannot have an approximation algorithm of any factor unless $\mathcal{P} = \mathcal{NP}$.

Remark: A Polytime Algorithm for the Constant k case

We would like to show that even the general problem becomes easy if the number of bins is constant. In particular, we show that there is an efficient algorithm if we are given the congestion vector. To remind, the congestion vector is a k-dimensional vector, where the i-th coordinate denotes the number of players in that bin. Note that if the number of bins is constant, then the number of possible congestion vectors is polynomial in n.

Once given the congestion vector, we reduce the problem of finding the OPT to finding the *minimum weight perfect b-matching* in a bipartite graph as follows. Consider the complete bipartite graph, $\mathcal{G}(U, V, E)$ where $U = \{1, \cdots, n\}$ denotes the n players while $V = \{1, \cdots, k\}$ denotes the k bins. The requirements (*b-values*) of $u \in U$ is 1. For a vector $j \in V$, $b(j) = n_j$, the jth coordinate of the congestion vector. The weights on each edge are $w(i, j) = A[i, j, n_j]$, where A is the input matrix. The optimal allocation with cost OPT is obtained from the minimum weight perfect b-matching, as each player is allocated one bin, and each bin gets the number of players as given by the congestion vector.

To get the minmax cost note that the number of possibilities for it are only polynomially many (n^2k). Thus as in the symmetric bins case we start from the smallest value M, and each time check the feasibility by checking the existence of a perfect b-matching in the above graph, with all values greater than M replaced by ∞.

5. CONCLUSIONS

In this paper we introduced the study of congestion games from a centralized viewpoint, where the players might not be free to make their own decisions. We have looked at two problems, one in which we try to find an allocation minimizing the total cost, and the other where we search for a fair allocation. We show that both these problems are very hard in the general case.

We consider natural restrictions of this problem, and give simple algortihms for these. In particular, both the problems are easy when all players are symmetric. We also study the same problems in the network congestion model, and show that social optimum can be found efficiently, while the minmax cost is hard to compute. We then show that the social optimum itself is a 3-prefix-sum approximation of the fair allocation. An interesting question is whether a better factor, or a coordinate wise approximation can be obtained by a totally different algorithm.

When the bins are symmetric, we give a simple algorithm to get the fair allocation. When the underlying congestion matrix is anti-Monge, we give an algorithm for finding the social optimum. We do not resolve the hardness in the case of general congestion matrices, although we believe that it is indeed hard, and approximation algorithms for the same might be an avenue for further research.

6. REFERENCES

[1] A. Awerbuch, A. Azar, E.F.Grove, P. Krishnan, M.Y.Kao, and J.S.Vitter. Load balancing in the l_p norm. In *Proc. of 36th FOCS 1995*, pages 383–391, 1995.

[2] R. Beier, A. Czumaj, P. Krysta, and B. Vocking. Computing equilibria for congestion games with (im)perfect information. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'04)*, pages 739–748, New Orleans, LA, Jan 2004.

[3] R. Burkard, B. Klinz, and R. Rudolf. Perspective of monge properties in optimization. *Discrete Applied Mathematics*, 70:95–161, 1996.

[4] A. Fabrikant, C. Papadinitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proc. of STOC 2004*, 2004.

[5] J. Kleinberg, Y. Rabani, and E. Tardos. Fairness in routing and load balancing. *J. Comput. Syst. Sci.*, 63(1):2–20, 2001.

[6] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria, 1999.

[7] A. Kumar and J. Kleinberg. Fairness measures for resource allocation. In *Proc. of FOCS 2000*, pages 75–85, 2000.

[8] I. Milchtaich. Congestion games with player specific pay-off functions. *Games and Economic Behavior*, 13:111–124, 1996.

[9] I. Milchtaich. Social optimality and cooperation in large congestion games. Manuscript, 2001.

[10] R. W. Rosenthal. A class of games possessing pure strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[11] T. Roughgarden. How unfair is optimal routing? In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 203–204. Society for Industrial and Applied Mathematics, 2002.

[12] T. Roughgarden and E. Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.

[13] S. Suri, C. Toth, and Y. Zhou. Uncoordinated load balancing and congestion games in p2p systems. In *3rd International Workshop on Peer-to-Peer Systems*, 2004.