Network Load Games*

Ioannis Caragiannis[1], Clemente Galdi[1,2], and Christos Kaklamanis[1]

[1] Research Academic Computer Technology Institute
Department of Computer Engineering and Informatics
University of Patras, 26500, Rio Greece
[2] Dipartimento di Informatica ed Applicazioni "R.M. Capocelli"
Universitá di Salerno, 84081, Baronissi (SA), Italy

Abstract. We study *network load games*, a class of routing games in networks which generalize selfish routing games on networks consisting of parallel links. In these games, each user aims to route some traffic from a source to a destination so that the maximum load she experiences in the links of the network she occupies is minimum given the routing decisions of other users. We present results related to the existence, complexity, and price of anarchy of Pure Nash Equilibria for several network load games. As corollaries, we present interesting new statements related to the complexity of computing equilibria for selfish routing games in networks of restricted parallel links.

1 Introduction

We study algorithmic questions related to a particular class of games in networks. A *game* with $n \geq 2$ *players* or *users* is a finite set of *actions* or *strategies* $\mathbf{S}_i$ for each user and a *payo® function* u_i (or, alternatively, a cost function c_i) defined over the users and the set $\mathbf{S}_1 \times \mathbf{S}_2 \times ... \times \mathbf{S}_n$. The elements of $\mathbf{S}_1 \times \mathbf{S}_2 \times ... \times \mathbf{S}_n$ are called *states*. A *Pure Nash Equilibrium* is a state $\mathbf{s} = \langle P_1, ..., P_n \rangle$ such that $u_i(P_1, ..., P_i, ..., P_n) \geq u_i(P_1, ...P_i', ..., P_n)$ for any $P_i' \in \mathbf{S}_i$. In general, a game does not have a Pure Nash Equilibrium.

In a *load game* the input is a set of n users, a set E of m resources, and the action sets are $\mathbf{S}_i \subseteq 2^E$. There are also given a load function L mapping $E \times \{1, ..., n\}$ to positive real numbers denoting how much a user increases the load of a resource it uses and a cost function ℓ mapping $E \times R$ to positive real numbers ($\ell_e(x)$ is non-decreasing in x). The payoffs/costs are computed as follows. Let $\mathbf{s} = (P_1, ..., P_n)$ be a state. Then

$$c_i(\mathbf{s}) = -u_i(\mathbf{s}) = \max_{e \in P_i} \left\{ \ell_e \left(\sum_{j: e \in P_j} L_e(j) \right) \right\}.$$

Intuitively, each user chooses a set of resources and the cost incurred by user i is the maximum load of the resources used by i.

* This work was partially supported by the European Union under IST FET Integrated Project 015964 AEOLUS.

In ***network load games***, the resources correspond to links of a directed network and the action sets correspond to paths in the network. Links are related in the sense that they have (possibly different) bandwidths and the users have (possibly different) traffic weights. Then the load function for edge e and user i is defined as $L_e(i) = w_i/b_e$ where w_i is the weight of user i and b_e is the bandwidth of edge e. Hence, there is given a directed network $G = (V, E)$, a source-destination pair s_i, t_i for each user i. The subsets of E available as actions to the user i is the set of all simple directed paths from s_i to t_i. In a state $\mathbf{s} = (P_1, ..., P_n)$, the payoff/cost for user i is defined as

$$c_i(\mathbf{s}) = -u_i(\mathbf{s}) = \max_{e \in P_i} \left\{ \frac{1}{b_e} \sum_{j:e \in P_j} w_j \right\}.$$

The ***social cost*** of a state $\mathbf{s} = (P_1, ..., P_n)$ is defined as the maximum load over all edges of the network, i.e., $SC(\mathbf{s}) = \max_{e \in E} \left\{ \frac{1}{b_e} \sum_{j:e \in P_j} w_j \right\}$.

The ***price of anarchy*** (or ***coordination ratio*** [11]) for a network load game is defined as the ratio of the worst social cost over all Pure Nash Equilibria over the optimal social cost. Intuitively, it gives a measure for the degradation of performance due to the selfish behavior of the users.

In their full generality, network load games are defined on multicommodity networks (i.e., the source-destination pairs may be different) while we are also interested in single-commodity networks or multicommodity networks with either one source or one destination node. Given a single-commodity network, we call ***stretch*** the ratio of the length of the longest simple path over the length of the shortest path from s to t. An important class of networks is that of layered networks; in these networks, all s-t paths have the same length (i.e., the stretch is 1). The simplest single-commodity network is a network of m parallel links connecting the source to the destination. In the generalization of restricted parallel links studied in [7] each user has a permissible set of links corresponding to her set of strategies. This game can be thought of as a network load game on a multicommodity network with one source and many destinations as it has been observed in [10]. Another combinatorial problem to which this game is related is the task scheduling on unrelated machines.

In the following we survey the results of the literature which are related to load games; these include results on the well-studied congestion games. We consider only on the existence, complexity, and price of anarchy of Pure Nash Equilibria. The extensive literature on mixed equilibria (their existence is guaranteed by the famous theorem of Nash [13]) is not discussed here.

The network load game on m parallel links (also known as selfish routing on parallel links and also studied in the context of task scheduling on related parallel machines) is the mostly studied game in the literature starting with the work of Papadimitriou and Koutsoupias in [11]. For this game, polynomial-time algorithms for computing Pure Nash Equilibria are known in the most general case of links with different bandwidths and users with different traffics while the problems of computing Pure Nash Equilibria of best of worst social cost are NP-hard [5]. The nashification technique presented in [4] shows that, starting from any assignment of users to links, a Pure Nash Equilibrium of not larger social cost can be computed in polynomial time. Using a polynomial-time approximation scheme (PTAS) for task scheduling on related parallel machines [8], a PTAS for computing a Pure Nash Equilibrium of best social cost (i.e., a Pure Nash Equilibrium of cost at most $1+\epsilon$ times the best social cost, for any constant $\epsilon > 0$) is obtained. Concerning the price of anarchy of Pure Nash Equilibria in such games, there is a tight bound of $2 - 1/m$ [11].

In the restricted links model, there is a polynomial-time 2-approximation algorithm for computing a Pure Nash Equilibrium of best social cost in the case of identical links (and users with different traffic) due to [7]. In the same paper, tight bounds on the price of anarchy of Pure Nash Equilibria for these games are presented. The problem of computing an assignment of best social cost is a special case of the single-source unsplittable flow problem for which constant approximations are presented in [9, 10]. Unfortunately, the solutions computed by these algorithms are not Pure Nash Equilibria in general.

Network load games generalize the games on (restricted) parallel links. Another generalization is the classes of *congestion games* and *network congestion games* which have received significant attention in the literature. The selfish routing games on (restricted) parallel links are special cases of congestion games as well. The definition of congestion games is similar to that of load games; the main difference being that the payoff of each user is defined as the sum of the loads on the links (or resources) used by the user (as opposed to the maximum which is assumed in load games). As we will see, this subtle difference implies that load games are essentially different than congestion games with respect to the existence, complexity, and price of anarchy of Pure Nash Equilibria. Formally, in congestion games there is a set E of resources, a set of n users with action sets $\mathbf{S}_i \subseteq 2^E$, and a *delay function* mapping $E \times \{1, ..., n\}$ to the integers ($d_e(j)$ is non-decreasing on j). Given a state $\mathbf{s} = (P_1, ..., P_n)$, let $f_{\mathbf{s}}(e) = |\{i : e \in P_i\}|$. Then, the payoff/cost of user i is defined as $c_i(\mathbf{s}) = -u_i(\mathbf{s}) = \sum_{e \in P_i} d_e(f_{\mathbf{s}}(e))$. In network congestion games, the set E corresponds to the links of a directed network $G = (V, E)$, each user has a source-destination pair $s_i, t_i \in V$ and her strategies are all simple directed paths from s_i to t_i in G. *Linear network congestion games* have linear delay functions which is equivalent to assuming that links have bandwidths b_e and the payoffs are computed as $c_i(\mathbf{s}) = -u_i(\mathbf{s}) = \sum_{e \in P_i} \frac{f_{\mathbf{s}}(e)}{b_e}$. In *weighted linear network congestion games* each user i may have a positive weight w_i and her payoff/cost is defined as $c_i(\mathbf{s}) = -u_i(\mathbf{s}) = \sum_{e \in P_i} \frac{\sum_{i:e \in P_i} w_i}{b_e}$.

Rosenthal [14] has shown that any congestion game has a Pure Nash Equilibrium. His proof is based on the definition of a potential function associated to the states of the game whose local minima correspond to Pure Nash Equilibria. Since that paper, the use of potential function arguments is the main tool for proving the existence of Pure Nash Equilibria. An interesting characterization of games with respect to the potential functions they admit is presented in [12]. Interestingly, generalizations of congestion games on single-commodity networks may not admit Pure Nash Equilibria as it is shown in [6]. Pure Nash Equilibria of best social cost can be computed efficiently in weighted linear network congestion games on single-commodity layered networks [6].

Fabrikant et al. in [3] study the complexity of computing any Pure Nash Equilibrium in congestion games. They show that, in general, the problem of computing a Pure Nash Equilibrium is PLS-complete (i.e., as hard as computing any object whose existence is guaranteed by a polynomially-computable potential function). For symmetric network congestion games (i.e., congestion games with users having the same set of strategies), a Pure Nash Equilibrium can be computed by a polynomial-time algorithm which minimizes Rosenthal's potential function through a reduction to min-cost flow [3].

In this paper, we present new results for the existence, complexity, and price of anarchy of Pure Nash Equilibria in network load games. In particular:

- In Section 2, we show that any load game has a Pure Nash Equilibrium. This is very interesting since the class of load games is especially broad. Also, this result stands in contrast with a result in [6] for generalized versions of network congestion games which do not always have Pure Nash Equilibria. We also study the relation of network load games to congestion games with respect to the potential functions they admit.
- We study the complexity of the problem of computing the best Pure Nash Equilibrium in network load games. The NP-completeness of the problem in single-commodity networks follows by the NP-completeness of the problem in the network of m parallel links. We use the Nashification technique of [4] in the case of single-commodity networks with identical links and users with different traffics, and we obtain a polynomial time approximation scheme for computing the best social cost. In the case of users with identical traffic, we show that Pure Nash Equilibria of best social cost can be computed in polynomial time in networks with either a single source or a single destination using a reduction to min-cost flow (the proof can be thought of as the minimization of a potential function). These networks include networks of restricted parallel links [7] as a special case. Concerning Pure Nash Equilibria of worst social cost, we show that the problem of computing them is inapproximable in single-commodity networks with stretch strictly larger than 1 and NP-hard in single-commodity layered networks and networks of restricted parallel links. Here, we exploit the intuitive relations of Pure Nash Equilibria in network load games to maximal matchings and longest paths in graphs. All these results are presented in Section 3.
- We also present tight bounds of $\Theta(\sqrt{\alpha m})$ on the price of anarchy in single-commodity networks with m identical links and stretch $\alpha \geq 1$. Due to our related inapproximability results, our upper bound also yield almost optimal approximations to the worst social cost of Pure Nash Equilibria in single-commodity networks with stretch strictly larger than 1. We also show a higher lower bound on the price of anarchy for single-commodity networks with arbitrary links. These results are presented in Section 4.

We conclude with open problems in Section 5. Due to lack of space, all the proofs but one have been omitted.

2 Existence of Pure Nash Equilibria

Although the definition of load games is quite general, we show that any such game has a Pure Nash Equilibrium. This comes in contrast with generalized versions of congestion games which may not have a Pure Nash Equilibrium [6]. The proof uses an appropriately defined potential function over the states of the game.

Theorem 1. *Any load game has a Pure Nash Equilibrium.*

In the following, we focus on network load games. We first attempt a characterization of these games with respect to the potential functions they admit. A potential function Φ defined over the states $\mathbf{s}_1$ and $\mathbf{s}_2$ of a game is called an exact potential if for any two states differing in the strategy of user i, it is $\Phi(\mathbf{s}_1) - \Phi(\mathbf{s}_2) = c_i(\mathbf{s}_1) - c_i(\mathbf{s}_2) = u_i(\mathbf{s}_2) - u_i(\mathbf{s}_1)$. By extending this definition, a potential function Φ defined over the states $\mathbf{s}_1$ and $\mathbf{s}_2$ of a game is called a ξ-potential for a positive vector ξ defined over the users if for any two states differing in the strategy of user i, it is $\Phi(\mathbf{s}_1) - \Phi(\mathbf{s}_2) = \xi_i(u_i(\mathbf{s}_1) - u_i(\mathbf{s}_2))$. Monterer and Shapley have proved in [12] that every finite potential game (i.e., a game admitting an exact potential function) is isomorphic to a congestion game. Weighted network congestion games in single-commodity layered networks are known to admit a ξ-potential [6]. Clearly, network load games with identical users on networks of parallel links are congestion games. Furthermore, it can be easily seen that load games with two identical users on networks with identical links admit an exact potential while load games with two users with different weights on networks with identical links admit a ξ-potential. The following lemma states that these are the only similarities load games share with congestion games with respect to the potential functions they admit.

Lemma 1. *There exists a load game with* 3 *identical users on a layered network with identical links and a load game with* 2 *users on a layered network with di®erent links that do not admit a* ξ*-potential.*

3 Complexity of Computing Pure Nash Equilibria

Computing Pure Nash Equilibria of best social cost. We first consider single-commodity networks with identical links and show how to extend the nashification technique for selfish routing on parallel links [4] in this case. In this way, we obtain a polynomial time approximation scheme (PTAS) for computing a Pure Nash Equilibrium with best social cost.

Theorem 2. *There is a PTAS for computing a Pure Nash Equilibrium of best social cost in single-commodity networks with identical links.*

The NP-hardness of the problem of computing a Pure Nash Equilibrium of best social cost in single-commodity networks with arbitrary users follows by the NP-hardness of the problem on the network of m parallel links. The proof assumes identical links. We can prove the following theorem which essentially shows that what makes the parallel links model hard is the existence of arbitrary users. The result is more general and also applies to selfish routing games on restricted parallel links [7].

Theorem 3. *A Pure Nash Equilibrium with the best social cost in networks with either one source or one destination and with identical users can be computed in polynomial time.*

Proof. Consider a network load game with n users with identical traffic on a network $G = (V, E)$ with a single source s and k destination nodes $t_1, ..., t_k$ and m links (the proof for networks with one destination is very similar). Denote by n_i the number of users wishing to route their traffic from s to t_i.

Consider all pairs (i, e) of the cartesian product $\{1, 2, ..., n\} \times E$ and sort them in non-decreasing order with respect to the value of the quantity i/b_e. Let $L \leq nm$ be the number of different values of i/b_e, and let $r(i, e)$ be such that i/b_e is the $r(i, e)$-th smallest among the L different values.

The proof uses a reduction to min-cost flow. We construct a network N having the same set of nodes as G by replacing each link of the original network by n parallel directed edges. Each edge has a unit capacity. The cost of the i-th edge corresponding to the link e is equal to $(m + 1)^{r(i,e)-1}$. The min-cost flow problem is defined by $f_s = n$, $f_{t_i} = -n_i$, for each destination node t_i, and $f_u = 0$ for any other node. Intuitively, this flow problem asks for pushing a total amount n units of flows from node s so that amounts of n_i units of flow reach the sink nodes t_i satisfying the capacity constraints so that the total cost of the edges carrying flow is minimized. We use a min-cost flow algorithm to to compute an optimal solution f for this problem. Note that, although the cost function on the edges is exponential, min-cost flow algorithms work in polynomial time and perform a polynomial number of operations (i.e., comparisons, additions, multiplications, etc.) which do not depend on the cost function. In our case, the costs are representable with a polynomial number of bits and each of the operations mentioned above can be implemented with a polynomial number of bit operations. Overall, both the space required and the running time are polynomial (see [1] for an extensive overview of min-cost flow algorithms).

Using the optimal solution to the above min-cost flow we construct an assignment for the original network load game as follows. We decompose the flow in N into n disjoint paths. Each of these paths corresponds to the strategy of the user in the obvious way. If a path uses some of the parallel edges corresponding to link e, the corresponding user 's path in G uses e.

Observe that if i flow paths use some of the parallel edges between two nodes u and v in the flow solution f, then these paths should traverse the first i edges, otherwise the solution would not be optimal.

We first show that the assignment produced in this way is a Pure Nash Equilibrium. Assume that this is not the case and that the maximum load in the path assigned to some user j by the flow algorithm is ℓ while user j has an incentive to change her strategy p_j and use another path p'_j of maximum load $\ell' < \ell$. Denote by i_e the number of users using link e in the assignment produced by the flow algorithm. Then ℓ corresponds to the load of some link e' which is used by i' users and it is $i'/b_{e'} = \ell$ while ℓ' corresponds to some link e'' which is used by $i_{e''}$ users and it is $\frac{i''+1}{b_{e''}} = \ell'$. This also implies that $r(i_{e''}, e'') < r(i_{e'}, e')$. Without loss of generality, we may assume that for each link $e \in p_j$, the flow path corresponding to path p_j traverses the i_e-th parallel edge corresponding to the link e of G. If this is not the case and some other flow path uses the i_e-th parallel edge, we can trivially exchange the edges used by the two flows without affecting the cost of f. Now, consider the flow f' which is obtained by f by changing the route of the flow path corresponding to user j and route it through the i_e-th parallel edge corresponding to link e for each $e \in p'_j \cap p_j$ and through the $i_e + 1$-th parallel edge corresponding to link e for each $e \in p'_j \backslash p_j$ (while no flow is routed through the i_e-th parallel edge corresponding to the link e for each $e \in p_j \backslash p'_j$). The cost of the new flow f' is

$$\begin{aligned} COST(f') &= COST(f) - \sum_{e \in p_j \backslash p'_j} (m+1)^{r(i_e,e)-1} + \sum_{e \in p'_j \backslash p_j} (m+1)^{r(i_e+1,e)-1} \\ &\leq COST(f) - (m+1)^{r(i_{e'},e')-1} + (m-1)(m+1)^{r(i_{e''},e'')-1} \\ &< COST(f) \end{aligned}$$

which contradicts the fact that f is a flow of minimum cost.

We now show that the Pure Nash Equilibrium defined by the optimal solution f of the flow problem has optimal social cost. Again, denote by ℓ the social cost of this assignment which corresponds to some link e' used by i' users so that $i'/b_{e'} = \ell$. Assume that there was another assignment with maximum load $\ell' < \ell$ corresponding to a pair (i'', e'') meaning that link

In the following, we show that computing the worst social cost in single-commodity networks is inherently more difficult than in the case of parallel links. The proof uses an approximation-preserving reduction from Longest Directed Path in Hamiltonian Graphs [2].

Theorem 4. *For any constants $\delta, \epsilon > 0$, there is no polynomial-time algorithm which approximates the worst social cost within $O((am)^{1/2-\epsilon})$ in single-commodity networks with m identical links and stretch $\alpha > 1+\delta$, unless P=NP. Also, there is no polynomial-time algorithm which approximates the worst social cost within $o(\sqrt{am}/\log^2 m)$, unless the Exponential Time Hypothesis fails.*

In the case of networks with arbitrary links we can show a stronger result by slightly modifying the construction in the proof of Theorem 4.

Corollary 1. *For any constants $\delta, \epsilon > 0$, there is no polynomial-time algorithm which approximates the worst social cost within $O(m^{1-\epsilon})$ in single-commodity networks with m arbitrary links and stretch $\alpha > 1+\delta$, unless P=NP. Also, there is no polynomial-time algorithm which approximates the worst social cost within $o(m/\log^2 m)$, unless the Exponential Time Hypothesis fails.*

The proofs of the above two statements make use of the fact that the stretch is strictly larger than 1. In the case of single-commodity layered networks, we can still show a negative result. The proof is based on a reduction from Minimum Maximal Bipartite Matching [16].

Theorem 5. *Computing a Pure Nash Equilibrium of worst social cost in single-commodity layered networks with identical links and users with identical traffic is NP-hard.*

The case of restricted parallel links. Next, we consider the case of restricted parallel links studied in [7], a special case of network load games on multicommodity networks. We consider the case of users with identical traffic. Recall that Theorem 3 yields a polynomial-time algorithm for computing a Pure Nash Equilibrium of the best social cost. The following theorem states that computing Pure Nash Equilibria of worst social cost becomes difficult in restricted parallel links with identical users (as opposed to parallel links). The proof uses a reduction from Minimum Maximal Bipartite Matching [16].

Theorem 6. *Computing the worst Pure Nash Equilibrium for users of identical traffic in restricted parallel identical links is NP-hard.*

4 Bounds on the Price of Anarchy

The results presented in the following establish a tight bound of $\Theta(\sqrt{\alpha m})$ on the price of anarchy of Pure Nash Equilibria on network load games on single-commodity networks with identical links. The upper bound is stated in Theorem 7 while the lower bound is stated in Theorem 8. In particular, Theorem 7 also implies that any Pure Nash Equilibrium (including the one of the best social cost) gives an almost optimal approximation to the worst social cost. Assuming that the Exponential Time Hypothesis (that Satisfiability has no subexponential-time algorithms) holds, the corresponding approximation ratio is optimal within polylogarithmic factors. Our constructions in the proof of Theorem 8 are generalizations of the construction yielding the Braess Paradox in other selfish routing games (see e.g., [15]).

Theorem 7. *The price of anarchy of network load games in single-commodity networks with m identical links and stretch α is at most $O(\sqrt{\alpha m})$.*

Theorem 8. *For any integer $m \geq 9$ and α such that $1 \leq \alpha \leq m-1$, there exists a network load game on a single-commodity network with at most m identical links and stretch at most α such that the price of anarchy is $\Omega(\sqrt{\alpha m})$.*

By slightly modifying the construction in the proof of Theorem 8, we obtain an even worse lower bound on the price of anarchy in network load games on single-commodity networks with arbitrary links (and identical users).

Theorem 9. *For any integer $m \geq 7$, there exists a network load game with identical users on a single-commodity layered network of at most m arbitrary links with price of anarchy $\Omega(m)$.*

5 Open Problems

Our work reveals some interesting open questions:

- We have not provided any polynomial-time algorithm for computing any Pure Nash Equilibrium in single-commodity networks with arbitrary links and users with arbitrary traffic. The nashfication technique of [4] for arbitrary users and arbitrary parallel links does not seem to apply in this case. Furthermore, the following question is very challenging. Is there a constant approximation algorithm or even a PTAS for computating of a Pure Nash Equilibrium of best social?
- What is approximability of computing a Pure Nash Equilibrium of worst social cost in single-commodity layered networks? Although we have proved that the problem is NP-hard, the only known upper bound on its approximability is the $O(\sqrt{m})$ implied by Theorem 7 for single-commodity layered networks with identical links.
- Is there a polynomial-time algorithm for computing the best social cost in networks with either a single source or a single destination with arbitrary links when the number of different traffic weights of the users is constant? Note that all possible values of the potential function are representable with a polynomial number of bits in this case as well but the reduction of Theorem 3 does not seem to extend in this case. Also, even in the simplest case of restricted parallel arbitrary links and users of arbitrary traffic, computing any Pure Nash Equilibrium in polynomial time is still open.

– Is there a load game or even a network load game which is PLS-complete? A characterization of these games like the one presented in [3] for congestion games would be very interesting.

References

1. R.K. Ahuja, T.L. Magnati, and J.B. Orlin. Network flows, Theory, Algorithms, and Applications. *Prentice Hall*, 1993.
2. A. Björklund, T. Husfeldt, and S. Khanna. Approximating longest directed paths and cycles. In *Proceedings of the 31st International Colloquium on Automata, Languages, and Programming (ICALP '04)*, LNCS 3142, Springer, pp. 222-233, 2004.
3. A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC '04)*, pp. 604-612, 2004.
4. R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode. Nashification and the coordination ratio for a selfish routing game. In *Proceedings of the 30th International Colloquium on Automata, Language, and Programming (ICALP '03)*, LNCS 2719, Springer, pp. 514-526, 2003.
5. D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *Proceedings of the 29th International Colloquium on Automata, Language, and Programming (ICALP '02)*, LNCS 2380, Springer, pp. 123-134, 2002.
6. D. Fotakis, S. Kontogiannis, and P. Spirakis. Selfish unsplittable flows. In *Proceedings of the 31st International Colloquium on Automata, Language, and Programming (ICALP '04)*, LNCS 3142, Springer, pp. 593-605, 2004.
7. M. Gairing, T. Lücking, M. Mavronicolas, and B. Monien. Computing Nash equilibria for restricted parallel links. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC '04)*, pp. 613-622, 2004.
8. D.S. Hochbaum and D. Shmoys. A polynomial approximation scheme for scheduling on uniform processors: using the dual approximation approach. *SIAM Journal on Computing*, 17(3), pp. 539-551, 1988.
9. J. Kleinberg. Single-source unsplittable flow. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS '97)*, pp. 68-77, 1996.
10. S. Kolliopoulos and C. Stein. Approximation algorithms for single-source unsplittable flow. *SIAM Journal on Computing*, 31, pp. 919-946, 2002.
11. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS '99)*, LNCS 1563, Springer, pp. 404-413, 1999.
12. D. Monterer and L. S. Shapley. Potential games. *Games and Economic Behavior*, Vol. 14, pp. 124-143, 1996.
13. J. F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2), pp. 286-295, 1951.
14. R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, Vol. 2, pp. 65-67, 1973.
15. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2), pp. 236-259, 2002.
16. M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, Vol. 38(3), pp. 364-372, 1980.