Algorithms for the Price of Optimum in Stackelberg Games (2006; Kaporis, Spirakis)

Alexis C. Kaporis, Department of Computer Engineering and Informatics, University of Patras
http://students.ceid.upatras.gr/~kaporis
Paul G. Spirakis, Department of Computer Engineering and Informatics, University of Patras & RA Computer Technology Institute
http://www.cti.gr/Paul_Spirakis

entry editor: Paul G. Spirakis

INDEX TERMS: Algorithms, Games, Stackelberg, Price of Optimum.

1 PROBLEM DEFINITION

Stackelberg games [15] may model the interplay amongst an authority and rational individuals that selfishly demand resources on a large scale network. In such a game, the authority *(Leader)* of the network is modeled by a distinguished player. The selfish users *(Followers)* are modeled by the remaining players.

It is well known that selfish behavior may yield a *Nash Equilibrium* with cost arbitrarily higher than the optimum one, yielding unbounded *Coordination Ratio* or *Price of Anarchy (PoA)* [7, 13]. Leader plays his strategy first assigning a portion of the total demand to some resources of the network. Followers observe and react selfishly assigning their demand to the most appealing resources. Leader aims to drive the system to an *a posteriori* Nash equilibrium with cost close to the overall optimum one [4, 6, 8, 10]. Leader may also eager for his own rather than system's performance [2, 3].

A Stackelberg game can be seen as a special, and easy [10] to implement, case of *Mechanism Design*. It avoids the complexities of either computing taxes or assigning prices, or even designing the network at hand [9]. However, a central authority capable to control the overall demand on the resources of a network may be unrealistic in networks which evolute and operate under the effect of many and diversing economic entities. A realistic way [6] to act centrally even in large nets could be via *Virtual Private Networks (VPNs)* [1]. Another flexible way is to combine such strategies with *Tolls* [5, 14].

A dictator controlling the entire demand optimally on the resources surely yields $PoA = 1$. On the other hand, rational users do prefer a liberal world to live. Thus, it is important to compute the optimal Leader-strategy which controls the *minimum* of the resources *(Price of Optimum)* and yields $PoA = 1$. What is the complexity of computing the Price of Optimum? This is not trivial to answer, since the Price of Optimum depends crucially on computing an optimal Leader strategy. In particular, [10] proved that computing the optimal Leader strategy is hard.

The central result of this lemma is Theorem 5. It says that on nonatomic flows and arbitrary *s*-*t* networks & latencies, computing the minimum portion of flow and Leader's optimal strategy sufficient to induce $PoA = 1$ is easy [4].

Problem 1 $(G(V, E), s, t \in V, r)$**.**
INPUT*: Graph* G*,* $\forall e \in E$ *latency* ℓ_e*, flow* r*, a source-destination pair* (s, t) *of vertices in* V*.*

Output*: (i) The minimum portion α_G of the total flow r sufficient for an optimal Stackelberg strategy to induce the optimum on G. (ii) The optimal Stackelberg strategy.*

1.1 Models & Notations

We are given a graph $G(V, E)$ with parallel edges allowed. A number of rational and selfish users wish to route from a given source s to a destination node t an amount of flow r. We may also consider a partition of users in k commodities, where user(s) in commodity i wish to route flow r_i through a source-destination pair (s_i, t_i), for each $i = 1, \ldots, k$. Each edge $e \in E$ is associated to a latency function $\ell_e()$, positive, differentiable and nondecreasing on the flow traversing it.

Nonatomic flows. There are infinitely many users, each routing his infinitesimally small amount of the total flow r_i from a given source s_i to a destination vertex t_i in graph $G(V, E)$. A flow f is an assignment of jobs f_e on each edge $e \in E$. The cost of the injected flow f_e (satisfying the standard constraints of the corresponding network-flow problem) that traverses edge $e \in E$ equals $c_e(f_e) = f_e \times \ell_e(f_e)$. It is assumed that on each edge e the cost is convex with respect the injected flow f_e. The overall system's cost is the sum $\sum_{e \in E} f_e \times \ell_e(f_e)$ of all edge-costs in G. Let $f_{\mathcal{P}}$ the amount of flow traversing the s_i-t_i path $\mathcal{P}$. The latency $\ell_{\mathcal{P}}(f)$ of s_i-t_i path $\mathcal{P}$ is the sum $\sum_{e \in \mathcal{P}} \ell_e(f_e)$ of latencies per edge $e \in \mathcal{P}$. The cost $C_{\mathcal{P}}(f)$ of s_i-t_i path $\mathcal{P}$ equals the flow $f_{\mathcal{P}}$ traversing it multiplied by path-latency $\ell_{\mathcal{P}}(f)$. That is, $C_{\mathcal{P}}(f) = f_{\mathcal{P}} \times \sum_{e \in \mathcal{P}} \ell_e(f_e)$.

In an Nash equilibrium, all s_i-t_i paths traversed by nonatomic users in part i have a common latency, which is at most the latency of any untraversed s_i-t_i path. More formally, for any part i and any pair $\mathcal{P}_1, \mathcal{P}_2$ of s_i-t_i paths, if $f_{\mathcal{P}_1} > 0$ then $\ell_{\mathcal{P}_1}(f) \leq \ell_{\mathcal{P}_2}(f)$. By the convexity of edge-costs the Nash equilibrium is unique and computable in polynomial time given a floating-point precision. Also computable is the unique *Optimum* assignment O of flow, assigning flow o_e on each $e \in E$ and minimizing the overall cost $\sum_{e \in E} o_e \ell_e(o_e)$. However, not all optimally traversed s_i-t_i paths experience the same latency. In particular, users traversing paths with high latency have incentive to reroute towards more speedy paths. Therefore the optimal assignment is unstable on selfish behavior.

A Leader dictates a *weak* Stackelberg strategy if on each commodity $i = 1, \ldots, k$ controls a fixed α portion of flow r_i, $\alpha \in [0, 1]$. A *strong* Stackelberg strategy is more flexible, since Leader may control $\alpha_i r_i$ flow in commodity i such that $\sum_{i=1}^{k} \alpha_i = \alpha$. Let a Leader dictating flow s_e on edge $e \in E$. The a posteriori latency $\widetilde{\ell}_e(n_e)$ of edge e, with respect to the induced flow n_e by the selfish users, equals $\widetilde{\ell}_e(n_e) = \ell_e(n_e + s_e)$. In the a posteriori Nash equilibrium, all s_i-t_i paths traversed by the free selfish users in commodity i have a common latency, which is at most the latency of any selfishly untraversed path, and its cost is $\sum_{e \in E}(n_e + s_e) \times \widetilde{\ell}_e(n_e)$.

Atomic splittable flows. There is a finite number of atomic users $1, \ldots, k$. Each user i is responsible for routing a non-negligible flow-amount r_i from a given source s_i to a destination vertex t_i in graph G. In turn, each flow-amount r_i consists of infinitesimally small jobs.

Let flow f assigning jobs f_e on each edge $e \in E$. Each edge-flow f_e is the sum of partial flows $f_e^1, \ldots, f_e^k$ injected by the corresponding users $1, \ldots, k$. That is, $f_e = f_e^1 + \ldots + f_e^k$. As in the model above, the latency on a given s_i-t_i path $\mathcal{P}$ is the sum $\sum_{e \in \mathcal{P}} \ell_e(f_e)$ of latencies per edge $e \in \mathcal{P}$. Let $f_{\mathcal{P}}^i$ be the flow that user i ships through an s_i-t_i path $\mathcal{P}$. The cost of user i on a given s_i-t_i path $\mathcal{P}$ is analogous to her path-flow $f_{\mathcal{P}}^i$ routed via $\mathcal{P}$ times the total path-latency $\sum_{e \in \mathcal{P}} \ell_e(f_e)$. That is, the path-cost equals $f_{\mathcal{P}}^i \times \sum_{e \in \mathcal{P}} \ell_e(f_e)$. The overall cost $C_i(f)$ of user i is the sum of the corresponding path-costs of all s_i-t_i paths.

In a Nash equilibrium no user i can improve his cost $C_i(f)$ by rerouting, given that any user $j \neq i$ keeps his routing fixed. Since each atomic user minimizes its cost, if the game consists of only one user then the cost of the Nash equilibrium coincides to the optimal one.

In a Stackelberg game, a distinguished atomic Leader-player controls flow r_0 and plays first assigning flow s_e on edge $e \in E$. The a posteriori latency $\widetilde{\ell}_e(x)$ of edge e on induced flow x equals $\widetilde{\ell}_e(x) = \ell_e(x + s_e)$. Intuitively, after Leader's move, the induced selfish play of the k atomic users is equivalent to atomic splittable flows on a graph where each initial edge-latency ℓ_e has been mapped to $\widetilde{\ell}_e$. In game-parlance, each atomic user $i \in \{1, \ldots, k\}$, having *fixed* Leader's strategy, computes his *best reply* against all others atomic users $\{1, \ldots, k\} \setminus \{i\}$. If n_e is the induced Nash flow on edge e this yields total cost $\sum_{e \in E}(n_e + s_e) \times \widetilde{\ell}_e(n_e)$.

Atomic unsplittable flows. The users are finite $1, \ldots, k$ and user i is allowed to sent his non-negligible job r_i only on a *single* path. Despite this restriction, all definitions given in atomic splittable model remain the same.

2 KEY RESULTS

Let us see first the case of atomic splittable flows, on parallel M/M/1 links with different speeds connecting a given source-destination pair of vertices.

Theorem 1 (Korilis, Lazar, Orda [6]). *The Leader can enforce in polynomial time the network optimum if she controls flow r_0 exceeding a critical value $\underline{r}^0$.*

In the sequel, we focus on nonatomic flows on *s-t* graphs with parallel links. In [6] primarily were studied cases that Leader's flow cannot induce network's optimum and was shown that an optimal Stackelberg strategy is easy to compute. In this vain, if *s-t* parallel-links instances are restricted to ones with linear latencies of equal slope then an optimal strategy is easy [4].

Theorem 2 (Kaporis, Spirakis [4]). *We can efficiently compute the optimal Leader strategy on any instance (G, r, α) where G is an s-t graph with parallel-links and linear latencies of equal slope.*

We can approximate within $(1 + \epsilon)$ the optimal strategy in polynomial time if link-latencies are polynomials with non-negative coefficients.

Theorem 3 (Kumar, Marathe [8]). *There is a* fully polynomial approximate *Stackelberg scheme that runs in poly$(m, \frac{1}{\epsilon})$ time and outputs a strategy with cost $(1 + \epsilon)$ within the optimum strategy.*

We can do even better for parallel link *s-t* graphs with arbitrary latencies. Intuitively, in polynomial time we can compute a "threshold" value α_G sufficient for the Leader's portion to induce the optimum. The complexity of computing optimal strategies changes in a dramatic way around the critical value α_G from "hard" to "easy" (G, r, α) Stackelberg scheduling instances. We call α_G as the *Price of Optimum* for graph G.

Theorem 4 (Kaporis, Spirakis [4]). *On input an s-t parallel link graph G with arbitrary latencies we can efficiently compute the minimum portion α_G sufficient for a Leader to induce the optimum, as well as her optimal strategy.*

We conclude that the Price of Optimum α_G essentially captures the hardness of instances (G, r, α). Since, for Stackelberg scheduling instances $(G, r, \alpha \geq \alpha_G)$ the optimal Leader strategy yields $PoA = 1$ and it is computed as hard as in P, while for $(G, r, \alpha < \alpha_G)$ the optimal strategy yields $PoA < 1$ and it is as easy as NP [10].

The results above are limited to parallel-links connecting a given *s-t* pair of vertices. Is it possible to efficiently compute the Price of Optimum for nonatomic flows on arbitrary graphs? This is not trivial to settle. Not only because it relies on computing an optimal Stackelberg strategy, which is hard to tackle [10], but also because Proposition B.3.1 in [11] ruled out previously known performance guarantees for Stackelberg strategies on general nets.

The central result of this lemma is presented below and completely resolves this question (extending Theorem 4).

Theorem 5 (Kaporis, Spirakis [4]). *On arbitrary s-t graphs G with arbitrary latencies we can efficiently compute the minimum portion α_G sufficient for a Leader to induce the optimum, as well as her optimal strategy.*

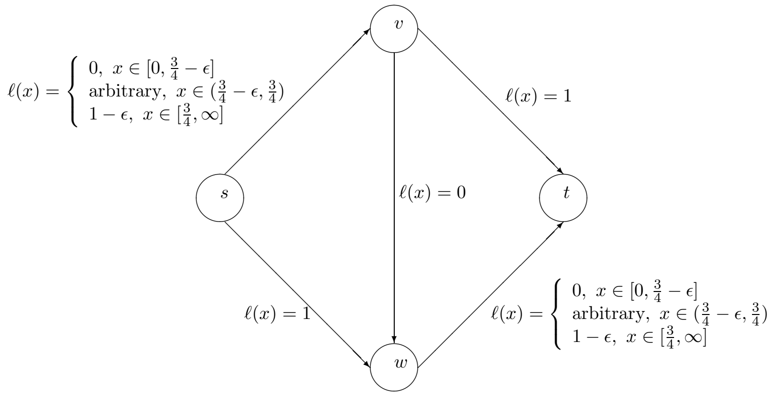

Figure 1: A bad example for Stackelberg routing.

Example. Consider the optimum assignment O of flow r that wishes to travel from source vertex s to sink t. O assigns flow o_e incurring latency $\ell_e(o_e)$ per edge $e \in G$. Let $\mathcal{P}_{s\to t}$ the set of all s-t paths. We can compute in polynomial time the *shortest paths* in $\mathcal{P}_{s\to t}$ with respect to costs $\ell_e(o_e)$ per edge $e \in G$. That is, the paths that given flow assignment O attain latency: $\min_{P\in\mathcal{P}_{s\to t}} \left(\sum_{e\in P} \ell_e(o_e)\right)$ i.e., minimize their latency. It is crucial to observe that, if we want the *induced* Nash assignment by the Stackelberg strategy to attain the optimum cost, then these shortest paths are *the only choice* for selfish users that eager to travel from s to t. Furthermore, the uniqueness of the optimum assignment O determines the minimum part of flow which can be selfishly scheduled on these shortest paths. Observe that any flow assigned by O on a non-shortest s-t path has incentive to opt for a shortest one. Then a Stackelberg strategy *must* frozen the flow on all non-shortest s-t paths.

In particular, the idea sketched above achieves coordination ratio 1 on the graph in Fig. 1. On this graph Roughgarden proved that $\frac{1}{\alpha}\times$ (optimum cost) guarantee is *not* possible for general (s,t)-networks, Appendix B.3 in [11]. The optimal edge-flows are ($r = 1$):

$$o_{s\to v} = \frac{3}{4} - \epsilon, o_{s\to w} = \frac{1}{4} + \epsilon, o_{v\to w} = \frac{1}{2} - 2\epsilon, o_{v\to t} = \frac{1}{4} + \epsilon, o_{w\to t} = \frac{3}{4} - \epsilon$$

The shortest path $P_0 \in \mathcal{P}$ with respect to the optimum O is $P_0 = s \to v \to w \to t$ (see [11] pp. 143, 5th-3th lines before the end) and its flow is $f_{P_0} = \frac{1}{2} - 2\epsilon$. The non shortest paths are: $P_1 = s \to v \to t$ and $P_2 = s \to w \to t$ with corresponding optimal flows: $f_{P_1} = \frac{1}{4} + \epsilon$ and $f_{P_2} = \frac{1}{4} + \epsilon$. Thus the Price of Optimum is

$$f_{P_1} + f_{P_2} = \frac{1}{2} + 2\epsilon = r - f_{P_0}$$

3 APPLICATIONS

Stackelberg strategies are widely applicable in networking [6], see also Section 6.7 in [12].

4 OPEN PROBLEMS

It is important to extend the above results on atomic unsplittable flows.

5 RECOMMENDED READING

[1] K. Birman. *Building Secure and Reliable Network Applications*. Manning, 1996.

[2] C. Douligeris and R. Mazumdar. Multilevel flow control of queues. In *Johns Hopkins Conference on Information Sciences*, 2006.

[3] A. Economides and J. Silvester. Priority load sharing: an approach using stackelberg games. In *28th Annual Alletron Conference on Communications, Control and Computing*, 1990.

[4] A. Kaporis and P. G. Spirakis. Stackelberg games on arbitrary networks and latency functions. In *18th ACM Symposium on Parallelism in Algorithms and Architectures*, 2006.

[5] G. Karakostas and G. Kolliopoulos. Stackelberg strategies for selfish routing in general multicommodity networks. Technical report, Advanced Optimization Laboratory, McMaster Univercity, 2006. AdvOL2006/08, 2006-06-27.

[6] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using stackelberg routing strategies. *IEEE/ACM Trans. Netw.*, 5(1):161–173, 1997.

[7] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *16th Symposium on Theoretical Aspects in Computer Science*, volume 1563, pages 404–413. LNCS, 1999.

[8] V. S. A. Kumar and M. V. Marathe. Improved results for stackelberg scheduling strategies. In *29th International Colloquium, Automata, Languages and Programming*, pages 776–787. LNCS, 2002.

[9] T. Roughgarden. Designing networks for selfish users is hard. In *42nd IEEE Annual Symposium of Foundations of Computer Science*, pages 472–481, 2001.

[10] T. Roughgarden. Stackelberg scheduling strategies. In *33rd ACM Annual Symposium on Theory of Computing*, pages 104–113, 2001.

[11] T. Roughgarden. *Selfish Routing*. Ph.D dissertation, Cornell University, USA, May 2002. http://theory.stanford.edu/ tim/.

[12] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. The MIT Press, 2005.

[13] T. Roughgarden and E. Tardos. How bad is selfish routing? In *41st IEEE Annual Symposium of Foundations of Computer Science*, pages 93–102, 2000.

[14] C. Swamy. The effectiveness of stackelberg strategies and tolls for network congestion games. In *ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[15] H. von Stackelberg. *Marktform und Gleichgewicht*. Springer-Verlag, 1934.