

# Achieving Allocatively-Efficient and Strongly Budget-Balanced Mechanisms in the Network Flow Domain for Bounded-Rational Agents

Yoram Bachrach and Jeffrey S. Rosenschein

{yori, jeff}@cs.huji.ac.il

School of Engineering and Computer Science

Hebrew University

Jerusalem, Israel

## Abstract

Vickrey-Clarke-Groves (VCG) mechanisms are a framework for finding a solution to a distributed optimization problem in systems of self-interested agents. VCG mechanisms have received wide attention in the AI community because they are efficient and strategy-proof; a special case of the Groves family of mechanisms, VCG mechanisms are the *only* direct-revelation mechanisms that are allocatively efficient and strategy-proof. Unfortunately, they are only weakly budget-balanced.

We consider self-interested agents in a network flow domain, and show that in this domain, it is possible to design a mechanism that is both allocatively-efficient and almost completely budget-balanced. This is done by choosing a mechanism that is not *strategy-proof* but rather *strategy-resistant*. Instead of using the VCG mechanism, we propose a mechanism in which finding a beneficial manipulation is an NP-complete problem, and the payments from the agents to the mechanism may be minimized as much as desired.

## 1 Mechanisms for Network Flow

Mechanisms face the problem of finding a system-wide solution to an optimization problem based on private information given by self-interested agents. A well-known solution to such problems in the case of quasi-linear preferences is that of Groves mechanisms. A special case of the Groves family of mechanisms are VCG mechanisms, which are budget-balanced, allocatively-efficient, and strategy-proof.

A significant disadvantage of VCG mechanisms is that they are only *weakly* budget-balanced. We would in principle prefer a *strongly* budget-balanced mechanism, where the total sum of payments to the mechanism is zero:  $\sum_i t_i(\theta) = 0$ . Impossibility results ([Green and Laffont, 1977] and [Hurwicz, 1975]) show that in a quasi-linear environment, it is impossible to achieve a mechanism that is strategy-proof, allocatively-efficient, and strongly budget-balanced.

We address this by slightly relaxing the strategy-proof requirement, replacing it with *strategy-resistance*. A mechanism is strategy-proof if the dominant strategy of each agent is to reveal its true type to the mechanism. Strategy-resistance

only requires that even if an agent is given the reported types of the other agents, it has a computationally intractable problem to solve if it wishes to find a beneficial manipulation (i.e., report a false type to the mechanism to gain higher utility).

In this work, we consider the domain of network flows. In that domain, the edges of a network flow belong to several self-interested agents. Each agent reports its edges to the mechanism. The mechanism is then required to choose a flow from the source vertex to the target vertex. Agents gain utility from flow units on their edges. Agents can lie, and declare a subset of the edges they really own. This manipulation may change the flow that the mechanism chooses, and agents may try to take advantage of this. However, we show that it is possible to use the fact that agents have computational limitations and are not unboundedly rational, so as to construct mechanisms with beneficial properties.

### 1.1 Self-Interested Network Flows

We explore the problem of designing a mechanism for bounded-rational agents in a distributed flow problem. We demonstrate that for this domain, we can find a mechanism that is allocatively-efficient,  $\epsilon$ -budget-balanced, and strategy-resistant. This means that if we assume the agents are bounded-rational and would not try to manipulate the mechanism if such manipulation is an NP-complete problem, they would all truthfully report their preferences. Once the mechanism gets their true preferences, it chooses the outcome that maximizes total utility of the agents. To achieve this truthfulness, the mechanism requires payments; however, the total sum of these payments can be minimized as much as required.

We now present the self-interested layered-graph network flow problem. Consider a flow network on a layered graph. We have a graph  $G = \langle V, E \rangle$ , with source vertex  $s$  and target vertex  $t$ . The vertices of the graph are partitioned into  $n + 1$  layers,  $L_0 = \{s\}, L_1, \dots, L_n = \{t\}$ . The edges only run between consecutive layers. We have a capacity function  $c : E \rightarrow \mathbb{R}$  which is the maximal flow allowed on the edges. We also have a set  $I$  of agents. Each agent controls a subset  $E_i \subset E$  of the graph's edges. No two agents control the same edge:  $\forall_{i \neq j} E_i \cap E_j = \emptyset$ .

The mechanism chooses a valid flow from  $s$  to  $t$ . A valid flow is a function  $f : E \rightarrow \mathbb{R}$  such that  $\forall_{(u,v) \in E} f(u,v) \leq c(u,v)$ ,  $\forall_{(u,v) \in E} f(u,v) = -f(v,u)$ , and  $\forall_{u \in V - \{s,t\}} \sum_{v \in V} f(u,v) = 0$ .

Each agent values the flow according to the total flow going through its edges. Let  $f$  be the valid flow chosen by the mechanism, and  $E_f$  the set of edges in  $f$  through which there is a positive flow:  $E_f = \{e \in E \mid f(e) > 0\}$ . We denote the set of  $A_i$ 's edges used in the flow  $f$  by:  $E_{f,i} = E_f \cap E_i$ . The agent's valuation of the flow is  $v_i(f) = \sum_{e \in E_{f,i}} f(e)$ .

When the mechanism is given the agents' true types,  $\theta = E_1, E_2, \dots, E_I$ , we want it to choose the flow that maximizes the total utility of the agents. The mechanism would be allocatively-efficient if it chooses  $f^*(\theta) = \arg \max_f \sum_i \sum_{e \in E_{f,i}} f(e)$ . In a *layered* graph the flow that maximizes the sum of agents' utilities is the maximal flow (proof of this is given in the full paper). Thus, if each agent truthfully declares its subset of edges, the mechanism can easily compute  $f^*(\theta)$  by running a maximal flow algorithm, such as the Edmonds-Karp algorithm.

As shown in the full version of this paper, a naive mechanism for the self-interested flow problem, with no payments to the mechanism, is not strategy-proof. An agent may declare a subset of the edges it controls, to change the flow that the mechanism chooses to a flow that the agent values more.

## 1.2 Self-Interested Network Flow Mechanism

We assume quasi-linear utility. Each agent pays the mechanism a payment  $p_i$ , and its utility is  $u_i(f) = v_i(f) - p_i$ . We now show that by using a straightforward payment rule, we make finding a beneficial manipulation NP-hard. The payment rule we use is simple: each agent pays the mechanism a constant of  $c$  for each edge it declares it owns. Let  $E'_i \subset E_i$  be the subset of edges an agent declares it owns. Then  $p_i(E'_i) = c|E'_i|$ . We also give a minor correction to make sure the mechanism is individual-rational. The payment  $p_i$  gives the agent a *utility* of 0 when the agent's valuation of the chosen flow is less than the payment to the mechanism. Assume that  $A_i$  knows  $E'_j$  for all  $j \neq i$ . It can easily calculate the utility it would get by truthfully declaring all its edges. How hard is it for  $i$  to find a subset of edges it could declare to the mechanism so as to gain a higher utility? We define the problem of *finding* a useful manipulation in the self-interested network flow domain as follows.

**FLOW-EDGE-SUBSET:** We are given a layered graph flow network, with the capacity function  $c : E \rightarrow \mathbb{R}$ ,  $E_{-i}$  the declared edges of the other agents, and  $E_i$ , the set of our edges. We are also given the constant  $c$  of the payment, and we know that if we declare that we have  $k$  edges, our payment to the mechanism would be  $p_i = ck$ . We assume the mechanism prefers a maximal flow that maximizes *our* utility of all the possible maximal flows. We are also given a constant  $k$ , the target utility for  $A_i$ . We are asked if there is a subset of  $A_i$ 's edges  $E'_i \subset E_i$ , such that the maximal flow chosen by the mechanism,  $f^*(E_1, \dots, E_{i-1}, E'_i, E_{i+1}, \dots, E_I)$  gives  $A_i$  a utility of at least  $k$ :  $u_i(f^*, p_i) = v_i(f^*) - p_i = \sum_{e \in E_{f^*,i}} f(e) - c|E'_i| \geq k$ .

A reduction, given in the full version of this paper, indicates that FLOW-EDGE-SUBSET is an NP-complete problem. The reduction given is from a general VERTEX-COVER instance to a FLOW-EDGE-SUBSET problem.

Although we do not include the full details of the reduc-

tion here, we do give some intuition about how it was constructed. Given the original VERTEX-COVER instance, we build a layer graph, where one of the agents,  $A_1$ , controls all the edges between two consecutive layers. Each edge  $A_1$  owns corresponds to a vertex in the original VERTEX-COVER graph. The network flow graph is constructed so that in order for  $A_1$  to maximize its valuation, it must declare a subset of edges that is a *vertex cover* of  $G$ . However, in order to minimize its payment to the mechanism,  $A_1$  must choose as few edges as possible. If the payment constant  $c$  is small enough,  $A_1$ 's top priority is to make its chosen vertices a vertex-cover, and only then choosing as few edges as possible. This makes its preferred choice the minimal vertex cover of the original graph.

## 2 Conclusions and Future Directions

We have suggested a mechanism for the distributed network flow problem with self-interested agents. With a proper choice of the payment constant  $c$ , finding a beneficial manipulation is an NP-complete problem. Since finding a beneficial manipulation is intractable, we encourage agents to truthfully report their preferences. In this case, the mechanism would choose the result maximizing the sum of agents' utilities, and we have an allocatively-efficient mechanism.

Given some  $\epsilon > 0$ , we can make the mechanism  $\epsilon$ -budget-balanced, by choosing a small enough payment constant  $c < \frac{\epsilon}{n(|E|+|V|)}$ , where  $n$  is the number of agents. This way, all of the agents together pay less than  $\epsilon$ . The mechanism we have described is also individual-rational. Therefore, in the domain of network flows, this achieves an individual-rational, allocatively-efficient,  $\epsilon$ -budget-balanced, and strategy-resistant mechanism. The standard VCG solution in this domain would be only weakly budget-balanced, but strategy-proof.

Impossibility results ([Green and Laffont, 1977] and [Hurwicz, 1975]) indicate that no direct-revelation mechanism can achieve strong budget-balance without sacrificing either allocative-efficiency or strategy-proofness. We believe that in many cases, trading strategy-proofness for strategy-resistance is a fair price to pay for achieving strong budget-balance.

Our strategy-resistance rested on the worst case assumptions of NP-hardness, which only indicates that a certain problem has *some* hard instances. A stronger notion of strategy-resistance could also require the manipulation problem to have no approximation methods, that it be in some harder complexity class, or that it be hard to find *any* (not just the optimal) beneficial manipulation. This work constitutes a first step in establishing the notion of strategy-resistance.

## References

- [Green and Laffont, 1977] Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45(2):427–38, March 1977.
- [Hurwicz, 1975] Leonid Hurwicz. On the existence of allocation systems whose manipulative Nash equilibria are pareto-optimal. In *3rd World Congress of the Econometric Society (Unpublished)*, 1975.