Lecture notes ?: An illustrative example: the core and network flow

Vincent Conitzer

1 Introduction

We will now consider an example from *cooperative game theory* (also known as *coalitional game theory*), the less-known sibling of noncooperative game theory (under which, for example, the zero-sum games studied earlier fall). Consider a setting with a set A of agents. Any subset $S \subseteq A$, in this context also known as a *coalition* of agents, can work together as a team; when they do, they generate a total value (say, profit) of $v(S)$, which they can distribute among themselves as they see fit. (There are also variants where there are limitations on how the value can be distributed, but we will not consider such variants here.) The function v is known as the *characteristic function*. We would like the *grand coalition* of all agents A to work together. If they do so, they will generate a value of $v(A)$, which they need to distribute among themselves. Let us say that agent i receives $\pi(i)$. We require that $\sum_{i \in A} \pi(i) \leq v(A)$: the agents cannot distribute more value than they generate. We now ask whether this payoff vector is *stable*. That is, it may be the case that there is a coalition $S \subset A$ such that $\sum_{i \in S} \pi(s) < v(S)$, so that the agents in S would be better off breaking off from the grand coalition and forming a coalition on their own. (It is not hard to see that in this case, the agents in S can distribute the $v(S)$ in such a way among themselves that each $i \in S$ receives more than the $\pi(i)$ she received before.) Such a coalition is called a *blocking coalition*. The existence of a blocking coalition implies that the payoff vector is not stable, in a sense. A payoff vector that is stable, that is, one for which no blocking coalition exists, is said to be in the *core*.

As an example, let us consider the following characteristic function for a game in which $A = \{a, b, c\}$:

- $v(\{a, b, c\} = 20$
- $v(\{a, b\} = 10$
- $v(\{a, c\} = 17$
- $v(\{b, c\} = 12$
- $v(\{a\}) = 1$
- $v(\{b\}) = 3$
- $v(\{c\}) = 10$

(We assume that $v(\emptyset)$ is always 0.) The only solution in the core is to set $\pi(a) = 7, \pi(b) = 3$, and $\pi(c) = 10$. If we change $v(\{b\})$ to 2, then there are multiple solutions in the core: $\pi(a) = 7, \pi(b) = 3$, and $\pi(c) = 10$ is still in the core, but so is $\pi(a) = 8, \pi(b) = 2$, and $\pi(c) = 10$ (and so is everything between these two vectors). If we change $v(\{b\})$ to 4, then the core is empty: that is, it is impossible to distribute the payoffs in such a way that no coalition wants to break off.

We can formulate the problem of finding a payoff vector that is in the core as a linear program (in fact, a linear feasibility program) as follows. (The $\pi(i)$ are the variables of this program.)

subject to
$(\forall S \subseteq A)\ \sum_{i \in S} \pi(i) \geq v(S)$
$-\sum_{i \in A} \pi(i) \geq -v(A)$
$(\forall i \in A)\ \pi(i) \geq 0$

One issue is that the number of constraints is exponential in the number of agents. We should note, however, that to specify a general characteristic function, we need to specify an exponential number of values, too. Hence, if we represent the characteristic function by listing the value for every coalition, then the size of the linear program is still polynomial in the size of the input. Of course, this is generally not a practical way of representing the characteristic function. Usually, the characteristic function is represented in a more concise way, and we will see examples of this shortly. We temporarily ignore this issue. Let us consider the dual of the linear program. Taking the dual of a linear feasibility program corresponds to the *Farkas Lemma*, but it is not necessary to remember this lemma explicitly, We can take the dual by first adding a trivial objective to the primal:

minimize 0
subject to
$(\forall S \subseteq A)\ \sum_{i \in S} \pi(i) \geq v(S)$
$-\sum_{i \in A} \pi(A) \geq -v(A)$
$(\forall i \in A)\ \pi(i) \geq 0$

The dual then becomes:

maximize $-v(A)u + \sum_{S \subseteq A} v(S)w(S)$
subject to
$(\forall i \in A)\ \ -u + \sum_{S \subseteq A : i \in S} w(S) \leq 0$
$u \geq 0$
$(\forall S \subseteq A) w(S) \geq 0$

Any feasible solution to the dual gives a lower bound on the objective that can be obtained in the primal. Therefore, if the dual has a feasible solution with positive objective value, then there cannot be any feasible solution to the primal (because any feasible solution to the primal has objective value 0). Conversely, by strong duality, if the dual has no feasible solution with positive value, then there must exist a feasible solution to the primal (with objective value 0). Hence, we know that the core is nonempty if and only if for any nonnegative u and $w(S)$ for which $\sum_{S \subseteq A : i \in S} w(S) \leq u$, it must be the case that $\sum_{S \subseteq A} v(S)w(S) \leq v(A)u$. Equivalently, dividing by u, we obtain that the core is nonempty if and only if for any nonnegative $w(S)$ for which $\sum_{S \subseteq A : i \in S} w(S) \leq 1$, it must be the case that $\sum_{S \subseteq A} v(S)w(S) \leq v(A)$. Finally, we can without loss of generality restrict attention to cases where each of the constraints is binding (since for constraint i we can always increase $w(\{i\})$), thereby obtaining that he core is nonempty if and only if for any nonnegative $w(S)$ for which $\sum_{S \subseteq A : i \in S} w(S) = 1$, it must be the case that $\sum_{S \subseteq A} v(S)w(S) \leq v(A)$. This result is known as the *Bondareva-Shapley Theorem*.

2 Constraint generation for the core

Now we return to the issue that the number of constraints in the primal is exponential. A general solution to this is to find an oracle that, given a particular solution $\pi(i)$, identifies a coalition S such that $\sum_{i \in S} \pi(s) < v(S)$, corresponding to a violated constraint. In general, a natural approach is to

look for the most violated constraint, that is, the coalition that maximizes $v(S) - \sum_{i \in S} \pi(s)$, How difficult this is depends on the way in which the characteristic function is represented. We will see an example in the next section.

3 A network flow game

Suppose that we have a directed graph $G = (V, E)$ with distinguished vertices s and t. A client needs to route a total flow of F (a positive integer) from s to t. Each of the edges in G has capacity 1 and corresponds to an agent that owns (only) that edge, that is, $A = E$. There is no inherent cost for routing flow along these edges, but the agents expect to be compensated for how valuable they are. Namely, if some of the flow is not routed along G, then the client needs to use an outside option for routing the flow. This outside option is represented by an additional edge from s to t. This edge has infinite capacity, a per-unit cost of 1, is outside of G and does not correspond to an agent. The value of a coalition S of agents is the total cost savings that would be obtained if only they were present, which is $v(S) = \min\{F, M(S)\}$, where $M(S)$ is the maximum flow over S. (Each unit that is routed via the agents represents a cost savings of 1, so the maximum flow corresponds to the highest cost savings that can be obtained, unless the total required flow F is less than this maximum flow, in which case the cost savings is simply F.) The goal now is to distribute the total savings $v(A) = \min\{F, M(A)\}$ among the agents in a way that is in the core. We note that the characteristic function, which is defined over exponentially many subsets of agents, is implicitly defined by the network.

We note that in this setup, the client, who has to compensate the agents, is not reaping any benefits from using the agents: the client still has to pay a total of F. Thus, the core constraint can be interpreted as follows: if there is a blocking coalition S (one for which $\sum_{i \in S} \pi(s) < v(S)$), these agents can get together with the client and propose that the client only uses their services (together with the outside option), compensating them more than $\sum_{i \in S} \pi(s)$ but less than $v(S)$. Then, the agents in S are happier, and also the client, who now pays less than F in total, is happier. Hence the original payoff vector is not stable.

As an example, consider the graph in Figure 1. The black, solid edge corresponds to the outside option with infinite capacity but a cost of 1. The blue, dashed edges correspond to the agents, each with a capacity of 1. The maximum flow over the agents is 2 (so 1 will have to be routed through the outside option). A solution in the core is to give 1 to each of the two leftmost edges.

As another example, consider the graph in Figure 2. Here, the maximum flow over the agents is 3, but because there is only demand for a total flow of 2, the value of the grand coalition is 2. In fact, here, the core is empty: any two agents have a value of 2, but it is impossible to ensure that each pair of agents receives 2.

Because there are exponentially many coalitions of agents, we cannot simply use the standard linear program for finding a payoff vector in the core. The first approach that we take is to find a constraint generation oracle that finds the maximally violated coalition. We can do so as follows. Given the current payoff vector of $\pi(i)$, we modify the network by placing a cost of $\pi(i)$ on each edge i (the edge's capacity remains 1). We also consider the outside-option edge part of the network. We then solve the minimum cost network flow problem on this capacity-constraint network. Because we can assume that the solution is integral by the Integrality Theorem, each agent is either used (with a flow of 1) or not used at all. Let S be the set of agents that are used: this will be the most violated coalition.

The idea behind this approach is the following. Suppose we take the perspective of the client, who is unhappy with the current arrangement in which she pays a total of F, and wants to find a cheaper solution by getting a coalition to deviate. In order to get a particular agent i to deviate, she needs to pay that agent at least $\pi(i)$, so that is effectively the cost of using that agent. From this

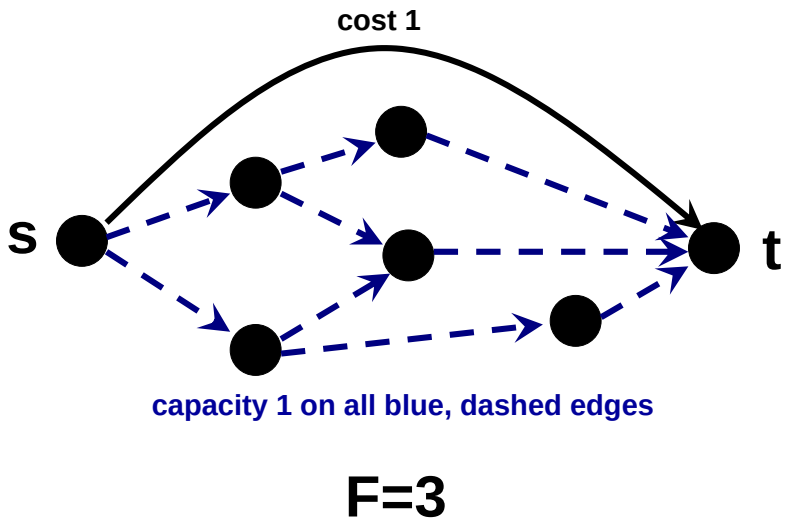

Figure 1: An instance of our problem with a nonempty core.

(and the Integrality Theorem) it is clear that the minimum cost network flow problem given above corresponds to the center's problem of minimizing her cost. If the client ends up using the agents in coalition S, then the client's cost savings (relative to paying the full cost of F) are $v(S) - \sum_{i \in S} \pi(S)$. Hence, the client's goal is to find the most violated coalition.

As it turns out, there is a simpler way to generate violated coalitions in this setting. We will prove the following: if there is a violated coalition, then there is also a violated coalition that corresponds to a path from s to t in G. To see this, first of all, we note that if there is a violated coalition, then there is also a violated coalition in which every agent is contributing to the flow. That is, it makes no sense to include in the coalition an isolated agent that does not contribute to the flow that can pass through that coalition, because if we remove this agent from the coalition, the resulting coalition will be at least as violated. Now, by the Integrality Theorem, this means that every agent in the violated coalition S has a flow of 1 passing through it. If a total flow of M passes through the coalition (we can assume $M \leq F$), then it must be the case that the coalition consists of M disjoint paths p from s to t. (To identify these paths, we can follow each unit of flow through the network.) Because the coalition is violated, we know that $M = v(S) > \sum_{i \in S} \pi(i) = \sum_p \sum_{i \in p} \pi(i)$. From this it follows that there must exist at least one path p such that $v(p) = 1 > \sum_{i \in p} \pi(i)$, which is what we set out to show.

From this, it follows that we only need to make sure that no coalition corresponding to a path is violated. There can still be exponentially many paths, though (can you create an example?). Can we efficiently generate the most violated path? This turns out to be quite easy: the most violated path is simply the shortest path from s to t, where the length of each edge i is $\pi(i)$. So we can simply use a shortest path algorithm for constraint generation.

It turns out that there is something *even* simpler that we can do, for which we do not need

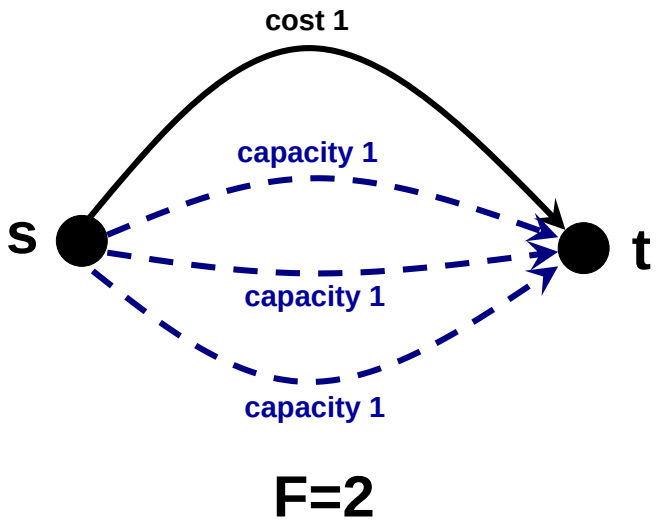

Figure 2: An instance of our problem with an empty core.

constraint generation at all. We make the following claim: there is a solution in the core if and only if the maximum flow $M(A)$ through the grand coalition is at most the total demand F. Here is how we show this. First, suppose that $M(A) \leq F$ (so that $v(A) = M(A)$). By the maximum flow/minimum cut theorem, we know that there then also consists a minimum cut S of $M(A)$ agents. We set $\pi(i) = 1$ for every one of these agents (which is feasible). Now, no path is violated, because every path from s to t includes one of the agents in the cut, who receives 1, which is the value of the path. So we have found a solution in the core. Conversely, suppose that $M(A) > F$. Then, there exist $M(A)$ disjoint paths from s to t. For each of these paths p, to have a solution in the core, we need $\sum_{i \in p} \pi(i) \geq 1$, but this is impossible because $v(A) = F < M(A)$. So there is no solution in the core.