Great Ideas in Computing

University of Toronto CSC196
Fall 2023

Week 11: November 27-Dec 1 (2023)

Week 11 slides

- I have posted the last question for Assignment 4. I modified the last question by also assuming $\cup A_i = S$; that is, the entire resource is allocated. This question will become clear today.
- By popular demand, I am postponing the due date of Assignment 4 to Wednesday, December 6, 9AM.
- Any comments about the guest presentation by Kyros Kutalakos on Monday, November 27.
- The results for the second quiz were very good! The mean was 88% and the median was 94%. I am very pleased about the performance in the quiz.

Agenda

- Today: Computational social choice
- Next week: Social networks

New topic: Computational social science

Traditionally, computer science (CS) and in particular theoretical CS (TCS) concerns itself with different computational settings and then studies what can and what can't be computed efficiently. Traditional applications come from commerce, operations research (e.g., scheduling and optimization), and the physical and mathematical sciences.

In recent years, in CS and TCS, we are becoming increasingly concerned with settings where algorithms take data coming from self-interested agents (i.e. people). That is, each agent has their own preferences or values for different outomes.

This brings us into the domain of the social sciences (e.g., sociology, psychology, political science, and economics). From economics, results in game theory and mechanism design (e.g., auctions) are now having a significant influence on CS. And conversely, CS is now having a significant influence in economics. As we have previoulsy discussed, online auctions in online advertising is the main source of revenue for search engines. .

Computational social choice

Traditional fields of study in the social sciences are now dealing with unprecedented volumes of data and hence computational issues are impacting these well studied disciplines.
Two main (and related) aspects in social choice theory are

1. Forming consensus (e.g, voting)
2. Fairness, and in particular, fair allocation of resources and chores.

Like almost anything in life there are many competing objectives and tradeoffs between what might be good for what we call the common "social welfare" vs what is fair to individuals or groups of individuals.

We will taken a brief look at this relatively new aspect of CS and TCS, called computational social choice.

Fairness

We shall avoid the more controversial aspects of "fair decisions" say as in decisions for who gets loans, paroles, admission to Universities, etc which are increasingly being made to some extent by machine learning algorithms. In these contexts, it can be very controversial as to how to define "fair".

Instead, we will focus on some precise well-studied applications and meanings of "fairness".

We assume that agents (i.e., people or groups of people) either have values for whatever they are allocated or *preferences* (i.e., a total or partial) order with respect to specific outcomes. We will assume no payments. Payments are used in some applications (e.g., auctions) to elicit truthfulness but using payments in say voting is not allowed.

Note that given a value for say each outcome, we have an induced (total or partial) order amongst the outcomes; that is, if agent i has values $v_i(A)$ and $v_i(B)$ for outcomes A and B then agent i weakly (resp. strictctly) prefers A to B iff $v_i(A) \geq v_i(B)$ (resp. $v_i(A) > v_i(B)$).

Some of the many dimensions of fair allocation of resources

The following are some of the dimensions of fair division when agents have values. Some definitions have natural analogues when agents only have preferences and not values.

- What are the fairness criteria?
- What properties do the valuation functions obey?
- Are decisions being made by a centralized mechanism or by decentralized self interested agents?
- Does the allocation take place simultaneously or in stages.
- Are agents truthful? How do we incentivize agents to particpate? How much information do agents have to reveal and how do we protect any information that is provided.
- Are the items divisible or indivisible?
- Is there a single (divisible) item that is being shared or multiple (divisible or indivisible) items to be shared.

And beyond all these issues, as I indicated, we now deal with enormous amounts of data when making decisions, thus necessitating algorithms that are computational efficiency in terms of time and memory.

Fairness criteria

There are a number of precisely defined criteria for fairness in the literature of fair division. Let S be the entire set of items or an entire divisible resource (e.g., "cake cutting") to be allocated and let n be the number of agents. Let $v_i(A_i)$ be the value that agent i obtains when receiving the allocation A_i. The following are well studied fairness measures: Let $(A_1, A_2, \ldots, A_n)$ be an allocation with $S = \cup_{i=1}^{n} A_i$ and the $\{A_i\}$ disjoint.

- Proportionality: If agent i has value $v_i(S)$ when allocated the entire resource S, then the allocation A_i it receives has value $v_i(A_i) \geq \frac{v_i(S)}{n}$.
- Envy-freeness: Agent i envies agent j if $v_i(A_j) > v_i(A_i)$. An allocation is envy-free if no agent envies another agent. One might say that "fairness is in the eye of the beholder".
- Equitability: An allocation $\mathbf{A}$ is equitable if there exists a value v such that $v_i(A_i) = v$ for all agents i. Note equitability does not that preclude $v_i(A_j) > v_i(A_i)$.
- Max-min fairness: The objective is to maximize the minimum allocation to any agent.

Pareto optimnality and social welfare

In addition to wanting some degree of fairness, we usually want some degree of "social benefit" as there is a social cost when agents try to deviate from a solution.
In particular, it is desireable to have outcomes that are *stable*.

Pareto optimality: An allocation $\mathbf{A} = (A_1, \ldots, A_n)$ is Pareto optimal if there does not exist another allocation $\mathbf{A}' = (A'_1, \ldots, A'_n)$ such that for some i, $v_i(A'_i) > v_i(A_i)$ and for all $j \neq i$, $v_j(A'_i) \geq v_j(A_j)$.

Pareto optimality is a stability condition but it also can be viewed as another fairness criteria. Pareto optimality seems "fair" in the sense that the given allocation can not be altered by anyone without harming someone else,

The *social welfare* of an allocation $(A_1, A_2, \ldots, A_n)$ amonsgt n agents is defined as $\sum_{i=1}^{n} v_i(A_i)$.

What kinds of questions about fairness do we ask?

For any fairness criteria, we want to know in whats setting can we achieve that criteria. And when we can't achieve the criteria is there an "approximate" version that can be achieve.

When we can achieve a fairness criteria, what is the computational cost and what is the social cost?

What is the relation (if any) between two different criteria? When can different criteria be simutaneoulsy be met?

In general we can envision allocating both divisible resoruces and indivisable items at the same time. For example, a will may specify that certain individual possessions (e,g, jewelery) and money be shared "equally" or "fairly" amongst the beneficiaries.

We will just takle a quick look at fair allocation of a single divisble resource and then at fair allocation of a set of indivisible items.

A quick look at a divisible resource

The common terminology for fair allocation of a divisible resource is called *cake cutting* where a cake can represent an arbitrary divisible resource. Note that some individuals may like certain parts of a cake and not other parts, or have different values for different parts of the cake.

Suppose we want to cut a cake amongst n people. Abstractly, we can think of the cake as a line. An allocation A_i to an agent i can be for a contiguous piece (i.e., a single interval) of the line or a union of intervals. We usually assume that the entire cake (the line) is consumed and that everyones valuation function is monotone.

Different settings induce different assumptions

In cake cutting, it may seem obvious that if every agent should have a monotone valuation function and that the resource is always entirely consumed. That is, if $B \subseteq C$, then $v_i(B) \leq v_i(C)$ and that $\cup_i A_i = S$.

But if you eat too much, you may regret it, and the result is that the eaters don't want to consume the entire cake. We will ignore that aspect and at least assume that the entire cake is eaten. In economic terms,

monotoncity of an agent's valuation function is refered to a free disposl; i.e. again extra amount of allocation can always be discarded at no cost.

If the divisible resource is money, is it always reasonable to assume that $v_(B) = v_j(B)$ for all i, j?

Different settings induce different assumptions

In cake cutting, it may seem obvious that if every agent should have a monotone valuation function and that the resource is always entirely consumed. That is, if $B \subseteq C$, then $v_i(B) \leq v_i(C)$ and that $\cup_i A_i = S$.

But if you eat too much, you may regret it, and the result is that the eaters don't want to consume the entire cake. We will ignore that aspect and at least assume that the entire cake is eaten. In economic terms,

monotoncity of an agent's valuation function is refered to a free disposl; i.e. again extra amount of allocation can always be discarded at no cost.

If the divisible resource is money, is it always reasonable to assume that $v_(B) = v_j(B)$ for all i, j?

People can value (or have different utilities) for the same amount of money. The St. Petersburg Paradox.

Some results about cake cutting

I am relying on slides from Professor Nisarg Shah.

The problem was first studied by Steinhaus [1948] and there are still unresolved computational questions.

There is a condition on the agent valuation functions (i.e., integrable density functions) but your intution about how you would cut a cake will suffice.

For $n = 2$, there is a folklore allocation method that is envy-free (and hence proportional), namely the cut and choose algorithm. Without loss of generality, we can assume that $v_1([0,1]) = 1$.
Agent 1 cuts the cake (i.e., the line [0,1]) and chooses some point x such that $v_1([0,x] = v_1([x,1]) = \frac{1}{2}$.
Agent 2 cuts chooses the piece that she prefers.

Cake cutting for $n > 2$ agents

The situation for more than 2 agents gets complicated. How do we study the complexity of cake cutting? The Robertson-Webb Model counts the number of queries of the following forms:

1. $Eval_i(x, y)$ returns $v_i([x, y])$
2. $Cut_i(x, \alpha)$ returns y: $v_i([x, y]) = \alpha$.

For proportional fairness, there is a moving knife algorithm: A referee starts a knife at 0 and moves the knife to the right.
When the piece to the left of the knife is worth $v_i(S)/n$, agent i requests that the knife stop moving and he gets that piece and exits.
This continues until there is only one agent left and she gets the remainder.

Claim: This can be implemented in the model and the query complexity is $\Theta(n^2)$; meaning that there the algorithm uses that many queries.

There is anlothert algorithm that uses $O(n \log n)$ queriess and $\Omega(n \log n)$ are necessary for any algorithm in the model.

Envy freeness for $n > 2$

And now the existence and complexity of envy freeness is really complicated.

- $n = 2$: cut and choose optimall uses 2 queries
- $n = 3$: There is a method useing 14 queries.
- $n = 4$: There is a method using 171 queries.
- Arbitrary n: There is a method due to Aziz and MacKenzie [2016] that uses $O(n^{n^{n^{n^{n^{n}}}}})$ queries.
- No method can use less than $\Omega(n^2)$ queries

The very different situation for indivisible

While for a specific set of agent valuations, there may be an envy-free allocation, in general proportionality and envy-freeness is not achievable. Just consider one item and two agents. Clearly the person who doesn't get the item is envious and doesn't get a proportional share of the total worth.

There is a weakening of the envy-free requirement that can always be achieved. Namely EF1 (envy-freeness minus one item) is the following property:
For every agent i, and every A_j for $j \neq i$, there exists at most element $x_j \in A_j$: $v_i(A_i) \geq v_i(A_j \setminus \{x_j\})$.

For monotone valuations, there is always an EF1 allocation.
Perhaps the simplest algorithm is the Round Robin algorithm.

The Round Robin algorithm

Algorithm 1: Round-Robin.

Input: A fair allocation instance $I = (N, M, v)$ with n agents and m goods.
Output: Allocation $A = (A_1, \ldots, A_n)$.
for each agent $i \in N$ **do**
    $A_i \leftarrow \varnothing$;
end
for $\ell = 1, \ldots, m$ **do**
    Let $i \leftarrow \ell \mod n$;
    Let $g^* \in \arg\max_{g \in M} v_i(g)$;
    $A_i \leftarrow A_i \cup \{g^*\}$;
    $M \leftarrow M \setminus \{g^*\}$;
end

Figure: The Round Robin algorithm