Markets with Production and constant number of goods

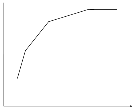
Jugal Garg and Ravi Kannan [Nikhil Devanur]

- First, assume no production.

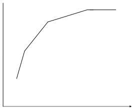
- First, assume no production.
- n agents. m goods.

- First, assume no production.
- n agents. m goods.
- Agent i has a utility function $U_i(x_i)$, where

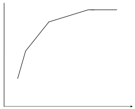
- First, assume no production.
- n agents. m goods.
- Agent i has a utility function $U_i(x_i)$, where
  - x_i is his "consumption bundle" (amount of each good consumed) given as a m vector of non-negative reals. (eg. (number of loaves of bread, liters of milk)).
    U_i is concave (Diminishing returns) and here assume piece-wise linear (we call it **PLC**) utilities.



- First, assume no production.
- n agents. m goods.
- Agent i has a utility function $U_i(x_i)$, where
  - x_i is his "consumption bundle" (amount of each good consumed) given as a m vector of non-negative reals. (eg. (number of loaves of bread, liters of milk)).
    U_i is concave (Diminishing returns) and here assume piece-wise linear (we call it **PLC**) utilities.

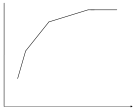


- First, assume no production.
- n agents. m goods.
- Agent i has a utility function $U_i(x_i)$, where
  - x_i is his "consumption bundle" (amount of each good consumed) given as a m vector of non-negative reals. (eg. (number of loaves of bread, liters of milk)).
    U_i is concave (Diminishing returns) and here assume piece-wise linear (we call it **PLC**) utilities.



- i always agent. j always good.

- First, assume no production.
- n agents. m goods.
- Agent i has a utility function $U_i(x_i)$, where
  - x_i is his "consumption bundle" (amount of each good consumed) given as a m vector of non-negative reals. (eg. (number of loaves of bread, liters of milk)).
    U_i is concave (Diminishing returns) and here assume piece-wise linear (we call it **PLC**) utilities.



- i always agent. j always good.
- 2 models : (i) Each agent comes with a fixed amount of money or (ii) a fixed bundle of goods.

- Price : a m vector giving unit price for each good. [Cost Linear]

- Price : a m vector giving unit price for each good. [Cost Linear]
- Equilibrium Price : A price at which

- Price : a m vector giving unit price for each good. [Cost Linear]
- Equilibrium Price : A price at which
  - if P_i is the set of bundles achieving maximum utility for agent i subject to budget (each agent maximizes separately ignoring others),

- Price : a m vector giving unit price for each good. [Cost Linear]
- Equilibrium Price : A price at which
  - if P_i is the set of bundles achieving maximum utility for agent i subject to budget (each agent maximizes separately ignoring others),
  - there exists a set of bundles $x_1 \in P_1, x_2 \in P_2, \ldots$ so that the market clears exacly.

- Price : a m vector giving unit price for each good. [Cost Linear]
- Equilibrium Price : A price at which
  - if P_i is the set of bundles achieving maximum utility for agent i subject to budget (each agent maximizes separately ignoring others),
  - there exists a set of bundles $x_1 \in P_1, x_2 \in P_2, \ldots$ so that the market clears exacly.
- Nash : An equilibrium price always exists.

- Price : a m vector giving unit price for each good. [Cost Linear]
- Equilibrium Price : A price at which
  - if P_i is the set of bundles achieving maximum utility for agent i subject to budget (each agent maximizes separately ignoring others),
  - there exists a set of bundles $x_1 \in P_1, x_2 \in P_2, \ldots$ so that the market clears exacly.
- Nash : An equilibrium price always exists.
- Central Problem in Computational Economics : Algorithm to find an equilibrium price.

- Price : a m vector giving unit price for each good. [Cost Linear]
- Equilibrium Price : A price at which
  - if P_i is the set of bundles achieving maximum utility for agent i subject to budget (each agent maximizes separately ignoring others),
  - there exists a set of bundles $x_1 \in P_1, x_2 \in P_2, \ldots$ so that the market clears exacly.
- Nash : An equilibrium price always exists.
- Central Problem in Computational Economics : Algorithm to find an equilibrium price.
- Our interest : Can we do it in poly time ? Yes, for constant number of goods.

- Price : a m vector giving unit price for each good. [Cost Linear]
- Equilibrium Price : A price at which
  - if P_i is the set of bundles achieving maximum utility for agent i subject to budget (each agent maximizes separately ignoring others),
  - there exists a set of bundles $x_1 \in P_1, x_2 \in P_2, \ldots$ so that the market clears exacly.
- Nash : An equilibrium price always exists.
- Central Problem in Computational Economics : Algorithm to find an equilibrium price.
- Our interest : Can we do it in poly time ? Yes, for constant number of goods.
- Note : If prices are given, each agent's problem is an LP – maximize min of several linear function. If prices are unknowns, problem is non-linear – (price)(amount of good bought).

- Deng Devanur Papdimitriou, Saberi Vazirani; Jain gave a poly-time algorithm for linear utilities.

- Deng Devanur Papdimitriou, Saberi Vazirani; Jain gave a poly-time algorithm for linear utilities.
- In many cases equilibrium prices and/or allocations are irrational numbers and hence one has to settle for approximations to the equilibrium. For instance, Codenotti et al. [CPV05] give an algorithm based on the ellipsoid method to compute an approximate equilibrium in markets with utilities that satisfy the so called weak gross substitutes (WGS) property. [Note : Approximate equilibrium very different beast than exact one.]

- Deng Devanur Papdimitriou, Saberi Vazirani; Jain gave a poly-time algorithm for linear utilities.
- In many cases equilibrium prices and/or allocations are irrational numbers and hence one has to settle for approximations to the equilibrium. For instance, Codenotti et al. [CPV05] give an algorithm based on the ellipsoid method to compute an approximate equilibrium in markets with utilities that satisfy the so called weak gross substitutes (WGS) property. [Note : Approximate equilibrium very different beast than exact one.]
- For non-constant number of goods, with PLC concave utilities, (or even Leontief utilities) PPAD hard. Codenotti, Saberi, Varadarajan and Ye

- Deng Devanur Papdimitriou, Saberi Vazirani; Jain gave a poly-time algorithm for linear utilities.
- In many cases equilibrium prices and/or allocations are irrational numbers and hence one has to settle for approximations to the equilibrium. For instance, Codenotti et al. [CPV05] give an algorithm based on the ellipsoid method to compute an approximate equilibrium in markets with utilities that satisfy the so called weak gross substitutes (WGS) property. [Note : Approximate equilibrium very different beast than exact one.]
- For non-constant number of goods, with PLC concave utilities, (or even Leontief utilities) PPAD hard. Codenotti, Saberi, Varadarajan and Ye
  - Leontief utilities (special case of PLC)

$$U_i(x_i) = min_j x_{ij}/\phi_{ij}.$$

Our Results

- Devanur, K. Polynomial time algorithm for finding an exact equilibrium when the number of goods in constant. We also do the same for the case when the number of agents is constant provided the utilities are separable –

$$U_i(x_i) = \sum_j U_{ij} x_{ij}.$$

  An important open question : Poly time alg for the case when number of agents is constant, but utilities are not separable.

Our Results

- Devanur, K. Polynomial time algorithm for finding an exact equilibrium when the number of goods in constant. We also do the same for the case when the number of agents is constant provided the utilities are separable –

  $$U_i(x_i) = \sum_j U_{ij} x_{ij}.$$

  An important open question : Poly time alg for the case when number of agents is constant, but utilities are not separable.
- Jugal Garg, K.
  - Constant number of goods now with Production - poly time
  - Reduction from market with Production, PLC utilities to one with no production and PLC utilities with same set of equilibria (1-1 correspondance)

Overview of method

- Step 1: Cell decomposition: Divide price space $\mathbf{R}_+^m$ into small cells, either by hyperplanes or polynomial surfaces so that (intuitively), the "order" of the pieces in the PLC utilities is the same for all price vectors in the cell.

Overview of method

- Step 1: Cell decomposition: Divide price space $\mathbf{R}_+^m$ into small cells, either by hyperplanes or polynomial surfaces so that (intuitively), the "order" of the pieces in the PLC utilities is the same for all price vectors in the cell.
- Step II : In each cell, either find a price that has a market clearing allocation, or certify that no such price exists. For this, one solves a system of linear/polynomial equations involving a constant number of variables.

MUPUC : Marginal utility per unit cost

- MUPUC = Increase in optimal utility when per unit increase in budget. (Simply slope divided by price of piece.)

MUPUC : Marginal utility per unit cost

- MUPUC = Increase in optimal utility when per unit increase in budget. (Simply slope divided by price of piece.)
- **Warm-up** : Constant number of goods. ($m \in O(1)$.) Separable utilities.

MUPUC : Marginal utility per unit cost

- MUPUC = Increase in optimal utility when per unit increase in budget. (Simply slope divided by price of piece.)
- **Warm-up** : Constant number of goods. ($m \in O(1)$.) Separable utilities.
- **Fact** There is a "critical MUPUC" - call it α_i for each agent i such that the set of optimal solutions for the agent consist precisely of those in which

MUPUC : Marginal utility per unit cost

- MUPUC = Increase in optimal utility when per unit increase in budget. (Simply slope divided by price of piece.)
- **Warm-up** : Constant number of goods. ($m \in O(1)$.) Separable utilities.
- **Fact** There is a "critical MUPUC" - call it α_i for each agent i such that the set of optimal solutions for the agent consist precisely of those in which
  - Agent buys fully all pieces with MUPUC $> \alpha_i$.

MUPUC : Marginal utility per unit cost

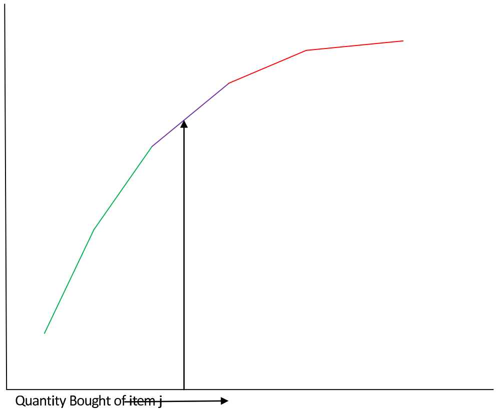
- MUPUC = Increase in optimal utility when per unit increase in budget. (Simply slope divided by price of piece.)
- **Warm-up** : Constant number of goods. ($m \in O(1)$.) Separable utilities.
- **Fact** There is a "critical MUPUC" - call it α_i for each agent i such that the set of optimal solutions for the agent consist precisely of those in which
  - Agent buys fully all pieces with MUPUC $> \alpha_i$.
  - Agent does not buy any of the pieces with MUPUC $< \alpha_i$.

MUPUC : Marginal utility per unit cost

- MUPUC $=$ Increase in optimal utility when per unit increase in budget. (Simply slope divided by price of piece.)
- **Warm-up** : Constant number of goods. ($m \in O(1)$.) Separable utilities.
- **Fact** There is a "critical MUPUC" - call it α_i for each agent i such that the set of optimal solutions for the agent consist precisely of those in which
  - Agent buys fully all pieces with MUPUC $> \alpha_i$.
  - Agent does not buy any of the pieces with MUPUC $< \alpha_i$.
  - Buys partially the pieces with exact equality.

MUPUC : Marginal utility per unit cost

- MUPUC $=$ Increase in optimal utility when per unit increase in budget. (Simply slope divided by price of piece.)
- **Warm-up** : Constant number of goods. ($m \in O(1)$.) Separable utilities.
- **Fact** There is a "critical MUPUC" - call it α_i for each agent i such that the set of optimal solutions for the agent consist precisely of those in which
  - Agent buys fully all pieces with MUPUC $> \alpha_i$.
  - Agent does not buy any of the pieces with MUPUC $< \alpha_i$.
  - Buys partially the pieces with exact equality.
- Quick Proof: If the agent was buying something with lower MUPUC than something the agent was not buying, then transfer some money from the former to the latter to gain. [Separable.]


Quantity Bought of item j

- Want to divide price space $\mathbf{R}_+^m$ into cells so that in each cell, the order of MUPUC of pieces is same for all prices in the cell.

- Want to divide price space $\mathbf{R}_+^m$ into cells so that in each cell, the order of MUPUC of pieces is same for all prices in the cell.
- Order of two pieces :

$$\frac{\text{Slope 1}}{\text{price 1}} \geq \frac{\text{slope 2}}{\text{price 2}} \equiv \text{ Slope 1price 2} \geq \text{ Slope 2price 1.}$$

- Want to divide price space $\mathbf{R}_+^m$ into cells so that in each cell, the order of MUPUC of pieces is same for all prices in the cell.
- Order of two pieces :

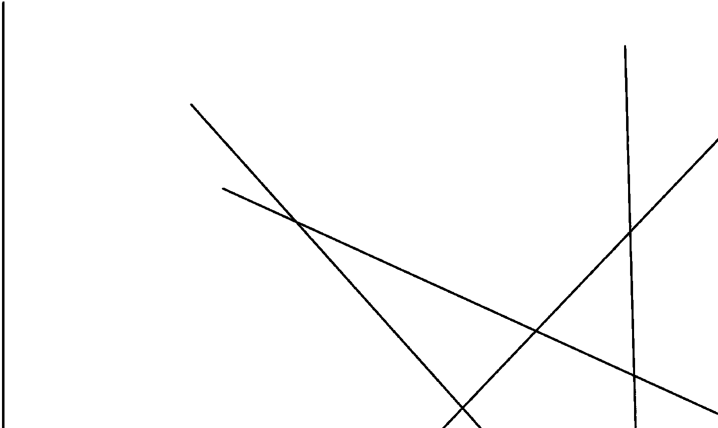
$$\frac{\text{Slope 1}}{\text{price 1}} \geq \frac{\text{slope 2}}{\text{price 2}} \equiv \text{ Slope 1price 2} \geq \text{ Slope 2price 1.}$$

- Put down all such linear constriants. Subdivides space into cells. Can we have exp(poly) cells ??

- Want to divide price space $\mathbf{R}_+^m$ into cells so that in each cell, the order of MUPUC of pieces is same for all prices in the cell.
- Order of two pieces :

$$\frac{\text{Slope 1}}{\text{price 1}} \geq \frac{\text{slope 2}}{\text{price 2}} \equiv \text{ Slope 1price 2} \geq \text{ Slope 2price 1.}$$

- Put down all such linear constriants. Subdivides space into cells. Can we have exp(poly) cells ??
- N hyperplanes in m space divide space into at most $\binom{N}{m}$ non-empty cells !! They can be found.



Problem in a cell

- In one cell, the pieces are linearly orderable by MUPUC's.

Problem in a cell

- In one cell, the pieces are linearly orderable by MUPUC's.
- Agent buys all of the first s pieces iff Total cost is at most budget - linear constraint in p, the price vector.

Problem in a cell

- In one cell, the pieces are linearly orderable by MUPUC's.
- Agent buys all of the first s pieces iff Total cost is at most budget - linear constraint in p, the price vector.
- Sub-divide cell with all such linear constraints so that in each sub-cell, know what whole pieces each agent buys.

Problem in a cell

- In one cell, the pieces are linearly orderable by MUPUC's.
- Agent buys all of the first s pieces iff Total cost is at most budget - linear constraint in p, the price vector.
- Sub-divide cell with all such linear constraints so that in each sub-cell, know what whole pieces each agent buys.
- Subtract full pieces to get residual.

Problem in a cell

- In one cell, the pieces are linearly orderable by MUPUC's.
- Agent buys all of the first s pieces iff Total cost is at most budget - linear constraint in p, the price vector.
- Sub-divide cell with all such linear constraints so that in each sub-cell, know what whole pieces each agent buys.
- Subtract full pieces to get residual.
- Variables : p, $f_{ij} =$ amount of money spent by i on good j only for partial pieces.

Problem in a cell

- In one cell, the pieces are linearly orderable by MUPUC's.
- Agent buys all of the first s pieces iff Total cost is at most budget - linear constraint in p, the price vector.
- Sub-divide cell with all such linear constraints so that in each sub-cell, know what whole pieces each agent buys.
- Subtract full pieces to get residual.
- Variables : p, $f_{ij} =$ amount of money spent by i on good j only for partial pieces.
-

$$\text{Clearing Good } j: \sum_i f_{ij} = p_j(\text{residual amount of good } j)$$

$$\text{Agent } i: \sum_j f_{ij} = \text{ residual budget of } i$$

$$0 \leq f_{ij}$$

Linear Program !!

Constant Number of Goods, general PLC utilities

- Each agent's LP is "local" to the agent EXCEPT:

Constant Number of Goods, general PLC utilities

- Each agent's LP is "local" to the agent EXCEPT:
  - Agent's budget: $\sum_j p_j x_{ij} = m_i$ and
- Want to Divide into cells, so that for any price vector in a cell, solution for all agents is "uniformly" computable.

Constant Number of Goods, general PLC utilities

- Each agent's LP is "local" to the agent EXCEPT:
  - Agent's budget: $\sum_j p_j x_{ij} = m_i$ and
  - Market Clearing: $\sum_i x_i = \mathbf{1}$.
- Want to Divide into cells, so that for any price vector in a cell, solution for all agents is "uniformly" computable.
- Agent i 's Linear Program:
  $\max y_0 : Ay \leq b \; ; \; y \geq 0 \; ; \; \sum_j p_j y_j = m$, where, A, b do not depend on price.

Constant Number of Goods, general PLC utilities

- Each agent's LP is "local" to the agent EXCEPT:
  - Agent's budget: $\sum_j p_j x_{ij} = m_i$ and
  - Market Clearing: $\sum_i x_i = \mathbf{1}$.
- Want to Divide into cells, so that for any price vector in a cell, solution for all agents is "uniformly" computable.
- Agent i 's Linear Program:
  $\max y_0 : Ay \leq b$; $y \geq 0$; $\sum_j p_j y_j = m$, where, A, b do not depend on price.
- Set of optimal solutions (need whole set - Why ?) is some face F of $\{y : Ay \leq b; y \geq 0\}$ intersected with budget constraint.

Constant Number of Goods, general PLC utilities

- Each agent's LP is "local" to the agent EXCEPT:
  - Agent's budget: $\sum_j p_j x_{ij} = m_i$ and
  - Market Clearing: $\sum_i x_i = \mathbf{1}$.
- Want to Divide into cells, so that for any price vector in a cell, solution for all agents is "uniformly" computable.
- Agent i 's Linear Program:
  $\max y_0 : Ay \leq b \;;\; y \geq 0 \;;\; \sum_j p_j y_j = m$, where, A, b do not depend on price.
- Set of optimal solutions (need whole set - Why ?) is some face F of $\{y : Ay \leq b; y \geq 0\}$ intersected with budget constraint.
- We can find some M linear constraints on p so that for all p satisfying those constraints, there is a single face F so that the set of optimal solutions is $F \cap$ budget. (LP Duality).
  $M \leq$POLY for $m \in O(1)$.

The harder part- In Each Cell

- Need to check if there is an equilibrium in a given cell. Now this is a non-linear problem. [The trick used in separable case of "taking critical MUPUC" to the other side to get rid of non-linearity does not work anymore......]

The harder part- In Each Cell

- Need to check if there is an equilibrium in a given cell. Now this is a non-linear problem. [The trick used in separable case of "taking critical MUPUC" to the other side to get rid of non-linearity does not work anymore......]
- Let P_i be set of optimal solutions of agent i.

The harder part- In Each Cell

- Need to check if there is an equilibrium in a given cell. Now this is a non-linear problem. [The trick used in separable case of "taking critical MUPUC" to the other side to get rid of non-linearity does not work anymore......]
- Let P_i be set of optimal solutions of agent i.
  - Want $x_i \in P_i, i = 1, 2, \ldots, n$ so that market clears : $\sum_i x_i = \mathbf{1}$. (*)

The harder part- In Each Cell

- Need to check if there is an equilibrium in a given cell. Now this is a non-linear problem. [The trick used in separable case of "taking critical MUPUC" to the other side to get rid of non-linearity does not work anymore......]
- Let P_i be set of optimal solutions of agent i.
  - Want $x_i \in P_i, i = 1, 2, \ldots, n$ so that market clears : $\sum_i x_i = \mathbf{1}$. (*)
- Lemma (Again LP Duality): (*) iff for all $m-$vectors q, $\sum_j q_j \leq \sum_i \text{Max}(q \cdot x_i : x_i \in P_i)$.

The harder part- In Each Cell

- Need to check if there is an equilibrium in a given cell. Now this is a non-linear problem. [The trick used in separable case of "taking critical MUPUC" to the other side to get rid of non-linearity does not work anymore......]
- Let P_i be set of optimal solutions of agent i.
  - Want $x_i \in P_i, i = 1, 2, \ldots, n$ so that market clears : $\sum_i x_i = \mathbf{1}$. (*)
- Lemma (Again LP Duality): (*) iff for all $m-$vectors q, $\sum_j q_j \leq \sum_i \text{Max}(q \cdot x_i : x_i \in P_i)$.
- Can cut up (p, q) space by polynomial inequalities, so that in each cell, the set of n vertices of P_i at which attain $\text{Max}(q \cdot x_i : x_i \in P_i)$ is the same.

The harder part- In Each Cell

- Need to check if there is an equilibrium in a given cell. Now this is a non-linear problem. [The trick used in separable case of "taking critical MUPUC" to the other side to get rid of non-linearity does not work anymore......]
- Let P_i be set of optimal solutions of agent i.
  - Want $x_i \in P_i, i = 1, 2, \ldots, n$ so that market clears : $\sum_i x_i = \mathbf{1}$. (*)
- Lemma (Again LP Duality): (*) iff for all $m-$vectors q, $\sum_j q_j \leq \sum_i \text{Max}(q \cdot x_i : x_i \in P_i)$.
- Can cut up (p, q) space by polynomial inequalities, so that in each cell, the set of n vertices of P_i at which attain $\text{Max}(q \cdot x_i : x_i \in P_i)$ is the same.
- N degree d polynomial inequalities in $2m$ space produce at most N^{md} cells.

Production

- Now include a number of factories. Each factory has a *production set* of pairs (x, y), where x, y are each a bundles (of goods). With x as raw materials, factory can produce y. The production set is convex. Each factory maximizes its profit independent of others.

Production

- Now include a number of factories. Each factory has a *production set* of pairs (x, y), where x, y are each a bundles (of goods). With x as raw materials, factory can produce y. The production set is convex. Each factory maximizes its profit independent of others.
- Each agent owns a fixed share of each factory.

Production

- Now include a number of factories. Each factory has a *production set* of pairs (x, y), where x, y are each a bundles (of goods). With x as raw materials, factory can produce y. The production set is convex. Each factory maximizes its profit independent of others.
- Each agent owns a fixed share of each factory.
- Now, there is a consumption bundle x_i for each agent, a "raw" bundle y_l for factory l, produced bundle z_l for factory l, all connected by obvious constraints. Each agent maximizes utility plus share of all factories.

Production

- Now include a number of factories. Each factory has a *production set* of pairs (x, y), where x, y are each a bundles (of goods). With x as raw materials, factory can produce y. The production set is convex. Each factory maximizes its profit independent of others.
- Each agent owns a fixed share of each factory.
- Now, there is a consumption bundle x_i for each agent, a "raw" bundle y_l for factory l, produced bundle z_l for factory l, all connected by obvious constraints. Each agent maximizes utility plus share of all factories.
- Equilibrium: Prices at which individually optimized x_i, y_l, z_l clear the market.

Equilibrium with Production - Constant number of goods

- We can convert a market with production into one without preserving (essentially) the set of equalibria by having a new "good" and a new agent for each factory.

Equilibrium with Production - Constant number of goods

- We can convert a market with production into one without preserving (essentially) the set of equalibria by having a new "good" and a new agent for each factory.
- But number of factories is not a constant, so does not reduce to old case.

Equilibrium with Production - Constant number of goods

- We can convert a market with production into one without preserving (essentially) the set of equalibria by having a new "good" and a new agent for each factory.
- But number of factories is not a constant, so does not reduce to old case.
- More directly, going back to the cell decomposition : Each factory's production set is a polytope and by decomposing cells further, we can ensure that the face of optimal production bundles of every factory is fixed throughout the (subdivided) cell.

Equilibrium with Production - Constant number of goods

- We can convert a market with production into one without preserving (essentially) the set of equalibria by having a new "good" and a new agent for each factory.
- But number of factories is not a constant, so does not reduce to old case.
- More directly, going back to the cell decomposition : Each factory's production set is a polytope and by decomposing cells further, we can ensure that the face of optimal production bundles of every factory is fixed throughout the (subdivided) cell.
- Proceed as earlier.... (Some more technical issues...)