CSE 5319 Notes 1: Game Theory Concepts

(Last updated 1/20/24 1:11 PM)

Karlin & Peres (KP) 1/2/3/4/5/7 (6 is worth skimming)

Nisan (N) 1/2

1.A. Selfish Behavior - Classic Games and Strategic Situations

Subtraction Game (Pile of chips, remove at least one and no more than four. Last move wins.)

N = moves where next/first player wins   P = moves where previous/second player wins

x chips is in P ⇔ x is divisible by 5

Impartial vs. Partisan Game; Progressively Bounded

Chomp (All starting positions are in N . . . "backward induction" on small cases leads to strategy-stealing)

Nim and Bouton's Exclusive-Or Solution (KP p. 17, Theorem 1.1.12)

Multiple piles

Remove any number of chips from one of the piles

Winner picks up the last chip(s)

1.1.2. Bouton's solution of Nim. We next describe a simple way of determining if a state is in **P** or **N**: We explicitly describe a set Z of configurations (containing the terminal position) such that, from every position in Z, all moves lead to Z^c, and from every position in Z^c, there is a move to Z. It will then follow by induction that $Z = \mathbf{P}$.

Such a set Z can be defined using the notion of *Nim-sum*. Given integers $x_1, x_2, \ldots, x_k$, the Nim-sum $x_1 \oplus x_2 \oplus \cdots \oplus x_k$ is obtained by writing each x_i in binary and then adding the digits in each column mod 2. For example:

	decimal	binary
x_1	3	0 0 1 1
x_2	9	1 0 0 1
x_3	13	1 1 0 1
$x_1 \oplus x_2 \oplus x_3$	7	0 1 1 1

Definition 1.1.11. The **Nim-sum** $x_1 \oplus x_2 \oplus \cdots \oplus x_k$ of a configuration $(x_1, x_2, \ldots, x_k)$ is defined as follows: Write each pile size x_i in binary; i.e., $x_i = \sum_{j\geq 0} x_{ij} 2^j$ where $x_{ij} \in \{0,1\}$. Then

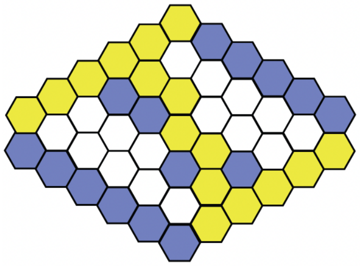
$$x_1 \oplus x_2 \oplus \cdots \oplus x_k = \sum_{j\geq 0} (x_{1j} \oplus \cdots \oplus x_{kj}) 2^j$$

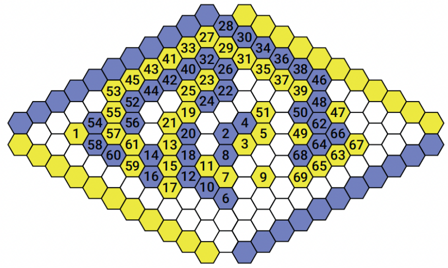
where for bits,

$$x_{1j} \oplus x_{2j} \oplus \cdots \oplus x_{kj} = \left(\sum_{i=1}^{k} x_{ij} \right) \bmod 2.$$

Theorem 1.1.12 (**Bouton's Theorem**). *A Nim position $x = (x_1, x_2, \ldots, x_k)$ is in* **P** *if and only if the Nim-sum of its components is* 0.

Hex - partisan, first player (standard board) has a win. Draw not possible (K p. 23)





(KP p. 2, Blue to move)

Shannon Switching Game (on a graph) - two players (short/reinforce, cut) compete to obtain a path between designated vertices or to block it. (First player has a win)

Tic-Tac-Toe

Optimal play leads to a draw

Isomorphism to Pick15 (`https://en.wikipedia.org/wiki/Number_Scrabble`)

3D Tic-Tac-Toe and `https://en.wikipedia.org/wiki/Zermelo%27s_theorem_(game_theory)` (`https://ranger.uta.edu/~weems/NOTES6319/PAPERSONE/patashnik.pdf`)

Optimal play leads to first player winning

Recursive Majority

Complete ternary tree

Players alternate marking (+/-) of the leaves

Mark parent based on majority marking of its children

Mark of parent indicates the winner

Like 3D Tic-Tac-Toe, contradiction for second-player to have forced win (p. 203 of Patashnik)

+ player has a forced win (Theorem 1.2.16)

Aside: `https://en.wikipedia.org/wiki/Sprouts_(game)`

Prisoner's Dilemma

KP p. 75

		prisoner II	
		silent	confess
prisoner I	silent	$(-1,-1)$	$(-10,0)$
	confess	$(0,-10)$	$(-8,-8)$

KP p. 129

		player II	
		cooperate (C)	defect (D)
player I	cooperate (C)	$(6,6)$	$(0,8)$
	defect (D)	$(8,0)$	$(2,2)$

Stag Hunt

KP p.75

Hunter I		Hunter II: Stag (S)	Hunter II: Hare (H)
	Stag (S)	(4, 4)	(0, 2)
	Hare (H)	(2, 0)	(1, 1)

War and Peace

KP p.75

Firm I		Firm II: diplomacy	Firm II: attack
	diplomacy	(2, 2)	(-2, 0)
	attack	(0, -2)	(-1, -1)

Driver and Parking Inspector

KP p. 76

Driver		Inspector: Don't Inspect	Inspector: Inspect
	Legal	$(0, 0)$	$(0, -1)$
	Illegal	$(10, -10)$	$(-90, -6)$

Cheetahs and Antelopes

KP p. 78

cheetah I		cheetah II: L	cheetah II: S
	L	$(\ell/2, \ell/2)$	(ℓ, s)
	S	(s, ℓ)	$(s/2, s/2)$

Chicken

KP p. 79

player I		player II: Swerve (S)	player II: Drive (D)
	Swerve (S)	(1, 1)	$(-1, 2)$
	Drive (D)	$(2, -1)$	$(-M, -M)$

Example 4.3.2 (**Pollution game**). Three firms will either pollute a lake in the following year or purify it. They pay 1 unit to purify, but it is free to pollute. If two or more pollute, then the water in the lake is useless, and each firm must pay 3 units to obtain the water that they need from elsewhere. If at most one firm pollutes, then the water is usable, and the firms incur no further costs.

If firm III purifies, the cost matrix (cost = − payoff) is

		firm II: purify	firm II: pollute
firm I	purify	(1,1,1)	(1,0,1)
	pollute	(0,1,1)	(3,3,4)

If firm III pollutes, then it is

		firm II: purify	firm II: pollute
firm I	purify	(1,1,0)	(4,3,3)
	pollute	(3,4,3)	(3,3,3)

1.B. Selfish Behavior - Two-Person/Zero-Sum

Matrix value is for the row player . . . Negate for the column player

Pick a Hand - Hider puts coin(s) in their two hands (left, right): (1, 0) or (0, 2).

KP p. 35

		Hider: $L1$	Hider: $R2$
Chooser	L	1	0
	R	0	2

Rock/Scissors/Paper (. . . Lizard/Spock)

KP p. 139

		player II: Rock	player II: Paper	player II: Scissors
player I	Rock	0	−1	1
	Paper	1	0	−1
	Scissors	−1	1	0

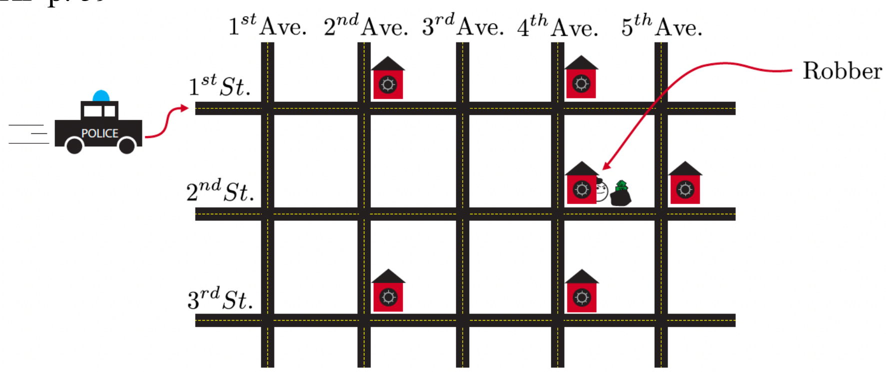
Santa Claus 🎅 and Easter Bunny 🐰?

Graphs

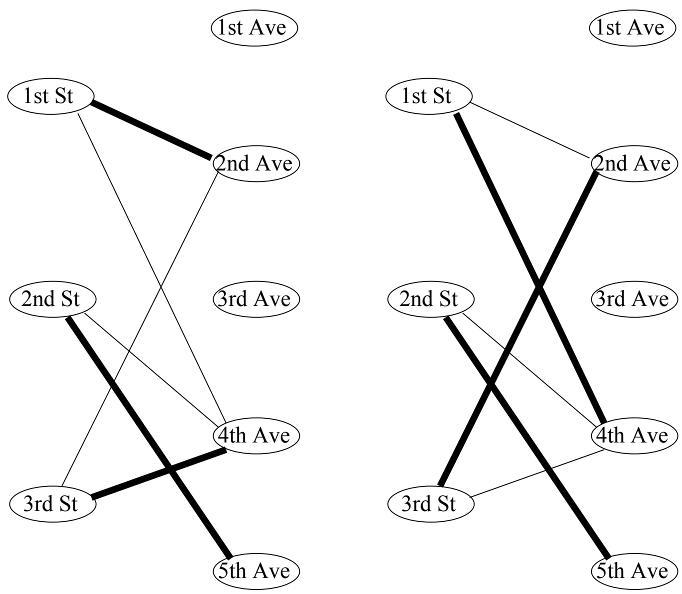
Series-sum, Parallel-sum, Troll and Traveler - SKIP

Hide and Seek

KP p. 59



COROLLARY 3.2.6. *For the Hide and Seek game, an optimal strategy for the cop is to choose uniformly at random a line in a minimum line-cover. An optimal strategy for the robber is to hide at a uniformly random safehouse in a maximum matching.*



Maximum (Bipartite) Matching

Maximum subset of edges such that no vertex is incident to two of the chosen edges

Minimum Covers

({1st St, 2nd St, 3rd St} {2nd Ave, 4th Ave, 5th Ave} {2nd St, 2nd Ave, 4th Ave})

0/1 matrix - Minimum set of rows and columns such that every 1 is in one of the chosen rows or columns

Graph - Minimum set of vertices such that every edge is incident to one of the chosen vertices

Hall's Theorem - Suppose the number of right column vertices is at least the number of left column vertices. If the neighborhood (e.g. corresponding right column vertices, Γ) for each subset of the left column contains as many vertices as in the subset, then there is a matching that includes all left column vertices. (KP p. 61, Theorem 3.2.4)

Konig's Lemma: Size of maximum matching = size of minimum line-cover. (KP p. 62, Lemma 3.2.5)

`hideAndSeek.1.kp59.gbt` - Finds one strategy using COROLLARY 3.2.6

`hideAndSeek.2.kp59.gbt` - Finds all strategies (same payoffs) using fundamental Nash equilibrium search

Linear Programming - KP Appendix A (Sedgewick's Java code for 2-person, 0-sum on webpage)

EQUILIBRIUM CONCEPTS FOR ZERO-SUM GAMES

Pure Strategies

Saddle Point - A row *i* and column *j* such that the smallest entry in the row is the same as the largest entry in the column. (The value is guaranteed to appear as entry a_{ij}). (Exercise 2.6)

Dominance - Recognizing and removing rows and columns that are not useful options

Simple - another row/column is always better

KP p. 41 - a convex combination dominates (2.9)

Mixed Strategy - Probability vector for playing available actions (KP p. 36-38)

Safety Strategy and Value for Player I (maximize expected gain)

Safety Strategy and Value for Player II (minimize expected loss)

THEOREM 2.3.1 (**Von Neumann's Minimax Theorem**). *For any two-person zero-sum game with $m \times n$ payoff matrix A, there is a number V, called the* value *of the game, satisfying*

$$\max_{\mathbf{x}\in\Delta_m} \min_{\mathbf{y}\in\Delta_n} \mathbf{x}^T A\mathbf{y} = V = \min_{\mathbf{y}\in\Delta_n} \max_{\mathbf{x}\in\Delta_m} \mathbf{x}^T A\mathbf{y}. \tag{2.3}$$

We will prove the Minimax Theorem in §2.6.

DEFINITION 2.5.1. A pair of strategies $(\mathbf{x}^*, \mathbf{y}^*)$ is a **Nash equilibrium** in a zero-sum game with payoff matrix A if

$$\min_{\mathbf{y}\in\Delta_n} (\mathbf{x}^*)^T A\mathbf{y} = (\mathbf{x}^*)^T A\mathbf{y}^* = \max_{\mathbf{x}\in\Delta_m} \mathbf{x}^T A\mathbf{y}^*. \tag{2.10}$$

Thus, $\mathbf{x}^*$ is a **best response** to $\mathbf{y}^*$ and vice versa.

REMARK 2.5.2. If $\mathbf{x}^* = \mathbf{e}_{i^*}$ and $\mathbf{y}^* = \mathbf{e}_{j^*}$, then by (2.2), this definition coincides with Definition 2.4.1.

Proposition 2.5.3. *Let $\mathbf{x} \in \Delta_m$ and $\mathbf{y} \in \Delta_n$ be a pair of mixed strategies. The following are equivalent:*

(i) *The vectors $\mathbf{x}$ and $\mathbf{y}$ are in Nash equilibrium.*

(ii) *There are V_1, V_2 such that*

$$\sum_i x_i a_{ij} \begin{cases} = V_1 & \text{for every } j \text{ such that } y_j > 0, \\ \geq V_1 & \text{for every } j \text{ such that } y_j = 0. \end{cases} \tag{2.11}$$

and

$$\sum_j a_{ij} y_j \begin{cases} = V_2 & \text{for every } i \text{ such that } x_i > 0, \\ \leq V_2 & \text{for every } i \text{ such that } x_i = 0. \end{cases} \tag{2.12}$$

(iii) *The vectors $\mathbf{x}$ and $\mathbf{y}$ are optimal.*

Safety values for the two players are *equal*

Safety strategies that yield the safety value are called ***optimal strategies***

Equalization Principle (KP p. 39)

Easily applied when each of the two players has two actions

Proposition 2.5.3 (KP p. 43) suggests finding optimal strategies by using linear programming (e.g. simplex method) to determine mixed strategies with identical safety values (V_1=V_2)

Removing Restriction to Zero-Sum

Definition 4.2.1 (**Nash equilibrium**). A pair of mixed strategy vectors $(\mathbf{x}^*, \mathbf{y}^*)$ with $\mathbf{x}^* \in \Delta_m$ (where $\Delta_m = \{\mathbf{x} \in \mathbb{R}^m : x_i \geq 0, \sum_{i=1}^m x_i = 1\}$) and $\mathbf{y}^* \in \Delta_n$ is a **Nash equilibrium** if no player gains by unilaterally deviating from it. That is,

$$(\mathbf{x}^*)^T A \mathbf{y}^* \geq \mathbf{x}^T A \mathbf{y}^*$$

for all $\mathbf{x} \in \Delta_m$ and

$$(\mathbf{x}^*)^T B \mathbf{y}^* \geq (\mathbf{x}^*)^T B \mathbf{y}$$

for all $\mathbf{y} \in \Delta_n$.

Nash's Theorem (KP p. 84)

Theorem 4.3.6 (**Nash's Theorem**). *Every finite general-sum game has a Nash equilibrium.*

For determining Nash equilibria in (small) games, the following lemma (which we have already applied several times) is useful.

Lemma 4.3.7. *Consider a k-player game where $\mathbf{x}_i$ is the mixed strategy of player i. For each i, let $T_i = \{s \in S_i \mid \mathbf{x}_i(s) > 0\}$. Then $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is a Nash equilibrium if and only if for each i, there is a constant c_i such that*[5]

$$\forall s_i \in T_i \quad u_i(s_i, \mathbf{x}_{-i}) = c_i \quad \text{and} \quad \forall s_i \notin T_i \quad u_i(s_i, \mathbf{x}_{-i}) \leq c_i.$$

Symmetric Games (KP p. 85)

1.C. POTENTIAL GAMES

Congestion Game

Cost for a driver i: $\text{cost}_i(\mathbf{P}) = \sum_{r \in P_i} c_r(n_r(\mathbf{P}))$

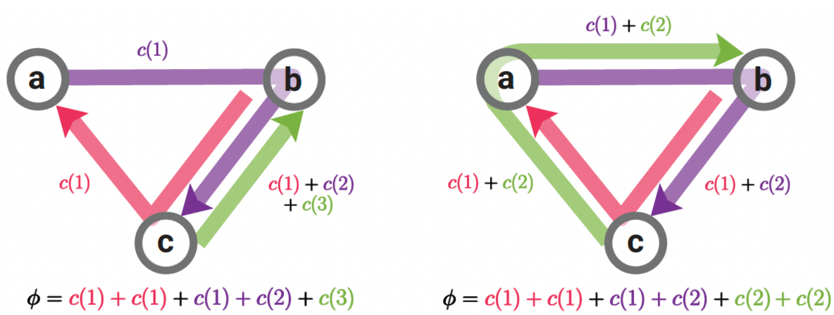
Potential Function (KP p. 86, (4.5)): $\phi(\mathbf{P}) := \sum_{r=1}^{R} \sum_{\ell=1}^{n_r(\mathbf{P})} c_r(\ell)$

($C_r(\bullet)$ is monotone increasing, but is not necessarily linear)

Minimizing a potential function gives a local optimum/pure Nash equilibrium (PNE).

COROLLARY 4.4.2. *Let ϕ be defined by* (4.5). *Fix a strategy profile* $\mathbf{P} = (P_1, \ldots, P_k)$. *If player i switches from path P_i to an alternative path P_i', then the change in the value of ϕ equals the change in the cost he incurs:*

$$\phi(P_i', \mathbf{P}_{-i}) - \phi(\mathbf{P}) = \text{cost}_i(P_i', \mathbf{P}_{-i}) - \text{cost}_i(\mathbf{P}). \tag{4.7}$$



(Easily translated to minimum-cost, maximum-flow problem. Solvable by linear programming.)

Nash Equilibrium in Pure Strategies (KP p. 87)

CLAIM 4.4.3. *Every potential game has a Nash equilibrium in pure strategies.*

Repeated Play Dynamics

"Last player" concept in potential function is straightforward motivation.

Best-response dynamics:

In turn, try each player:

Remove player's current path from G to give G′

Find shortest path in G′ (e.g. Dijkstra with $C_r(\bullet)$ for edge costs)

If path cost is an improvement, replace G with G′

Consensus

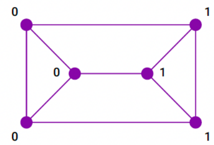


FIGURE 4.11. This is a Nash equilibrium in the Consensus game.

$$D_i(\mathbf{b}) = \sum_{j \in N(i)} |b_i - b_j|$$

$$\phi(\mathbf{b}) = \frac{1}{2} \sum_i D_i(\mathbf{b})$$

(comment regarding "*simultaneously*" and converging to cycle)

Graph Coloring (KP p.89) - seek to maximize potential function

EXAMPLE 4.4.9 (**Graph Coloring**). Consider an arbitrary undirected graph $G = (V, E)$ on n vertices. In this game, each vertex $v_i \in V$ is a player, and its possible actions consist of choosing a color s_i from the set $[n] := \{1, \ldots, n\}$. For any color c, define

$n_c(\mathbf{s})$ = number of vertices with color c when players color according to $\mathbf{s}$.

The payoff of a vertex v_j (with color s_j) is equal to the number of other vertices with the same color if v_j's color is different from that of its neighbors, and it is 0 otherwise; i.e.,

$$u_j(\mathbf{s}) = \begin{cases} n_{s_j}(\mathbf{s}) & \text{if no neighbor of } v_j \text{ has the same color as } v_j, \\ 0 & \text{otherwise.} \end{cases}$$

LEMMA 4.4.10. *Graph Coloring has a pure Nash equilibrium.*

COROLLARY 4.4.11. *Let $\chi(G)$ be the* ***chromatic number*** *of the graph G, that is, the minimum number of colors in any proper coloring of G. Then the graph coloring game has a pure Nash equilibrium with $\chi(G)$ colors.*

INFINITE STRATEGY SPACES

Tragedy of the Commons - network channel version (KP p. 90; N p. 6)

Note that the player's utility at equilibrium ($1/(k+1)^2$) is not maximized ($1/4k$).

Nightclub Pricing (KP p. 91)

Aside: When to fire in a duel . . . (`https://en.wikipedia.org/wiki/Hamilton_(musical)`)

Fixed-Points, Sperner's Lemma, and Existence of Nash Equilibria (skim)

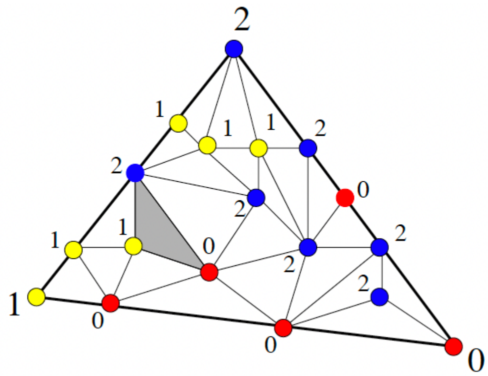


Figure 5.2. Sperner's lemma when $d = 2$.

Intractability (N p.33)

> **Theorem 2.3 (Gilboa and Zemel, 1989)** *The following are NP-complete problems, even for symmetric games: Given a two-player game in strategic form, does it have*
>
> - *at least two Nash equilibria?*
> - *a Nash equilibrium in which player 1 has utility at least a given amount?*
> - *a Nash equilibrium in which the two players have total utility at least a given amount?*
> - *a Nash equilibrium with support of size greater than a given number?*
> - *a Nash equilibrium whose support contains strategy s?*
> - *a Nash equilibrium whose support does not contain strategy s?*
> - *etc., etc.*

Succinct Representation (N p. 39)

Extensive (tree) vs. Strategic (matrix) Form

Complexity Theorists work with "natural" or "compact" representations

Approximate Equilibria (N p. 45)

(R p. 261 has summary of tractability results)

(R p. 240) $\in$-Approximate Coarse Correlated Equilibrium

(R p. 248) $\in$-Approximate Correlated Equilibrium

(R p. 296) $\in$-Approximate Mixed Nash Equilibrium

(R p. 197, 219) $\in$-Pure Nash Equilibrium

Example Bimatrix Game:

(5, 0)	(-25, 0)	(-25, 1)
(-25, 0)	(1, 0)	(-25, -1)

Nash Equilibria Distributions (payoff) (`optimal.cce.p3.gbt`)

Row	Column
1, 0 (-25)	0, 0, 1 (1)
1/2, 1/2 (-155/14)	13/28, 15/28, 0 (0)
1/2, 1/2 (-25)	0, 0, 1 (0)
0, 1 (-155/14)	13/28, 15/28, 0 (0)
0, 1 (1)	0, 1, 0 (0)

Correlated Equilibria Distribution (payoffs: 1, 0) (`corrEq.mod optimal.cce.p3.dat`)

0	0	0
0	1	0

Coarse Correlated Equilibria Distribution (payoffs: 3, 0) (`coarseCorrEq.mod`)

1/2	0	0
0	1/2	0

1.D. UTILITY FUNCTIONS, RISK AVERSION, ST. PETERSBURG PARADOX

The term *utility* is often used as a substitute for concepts like "money", "value", "revenue", "cost", "payoff", or "profit".

A *utility function* is a mapping from choices to an amount. (KP p. 81)

A narrower concept (vNM utility function) was developed in `https://en.wikipedia.org/wiki/Theory_of_Games_and_Economic_Behavior` to address preference and risk.

`https://en.wikipedia.org/wiki/St._Petersburg_paradox`

A casino offers a game of chance for a single player in which a fair coin is tossed at each stage. The initial stake begins at 2 dollars and is doubled every time heads appears. The first time tails appears, the game ends and the player wins whatever is in the pot. Thus the player wins 2 dollars if tails appears on the first toss, 4 dollars if heads appears on the first toss and tails on the second, 8 dollars if heads appears on the first two tosses and tails on the third, and so on. Mathematically, the player wins 2^{k+1} dollars, where k is the number of consecutive head tosses. What would be a fair price to pay the casino for entering the game?

To answer this, one needs to consider what would be the expected payout at each stage: with probability $\frac{1}{2}$, the player wins 2 dollars; with probability $\frac{1}{4}$ the player wins 4 dollars; with probability $\frac{1}{8}$ the player wins 8 dollars, and so on. Assuming the game can continue as long as the coin toss results in heads and, in particular, that the casino has unlimited resources, the expected value is thus

$$\begin{aligned} E &= \frac{1}{2}\cdot 2 + \frac{1}{4}\cdot 4 + \frac{1}{8}\cdot 8 + \frac{1}{16}\cdot 16 + \cdots \\ &= 1 + 1 + 1 + 1 + \cdots \\ &= \infty\,. \end{aligned}$$

Lottery: Set of mutually exclusive numeric outcomes with a probability for each outcome. (probabilities sum to 1.0)

St. Petersburg Lottery: (probability, outcome) pairs ($1/2^i$, 2^i)

(vNM) Utility function (to be applied to outcomes) such as:

Logarithm to some base

Square root

Example: Using *ln* as utility function for each lottery

Lottery 1: (0.5, \$100), (0.5, \$1)	$0.5 \bullet ln(100) + 0.5 \bullet ln(1) = 0.5 \bullet 4.6 = 2.3$
Lottery 2: (1.0, \$50)	$1.0 \bullet ln(50) = 3.9$

How to determine some utility function for an agent?

Presumably, an agent's preference over lotteries is consistent with four axioms:

`https://en.wikipedia.org/wiki/Von_Neumann–Morgenstern_utility_theorem#The_axioms` allowing a construction to be applied:

https://en.wikipedia.org/wiki/Von_Neumann–Morgenstern_utility_theorem#The_theorem

Where this leads:

https://en.wikipedia.org/wiki/Risk_aversion#Measures_of_risk_aversion_under_expected_utility_theory

BAYESIAN GAMES AND BAYES-NASH EQUILIBRIUM

KP p. 126 introduces Bayesian games using extensive-form notions, but switches to strategic-form (like N p. 20 and p. 233)

First-Price Auction (KP p. 233, N 9.6.2, R 2)

Single item

Two bidders with *private values* ($a < b$) from the same uniform distribution (0, 1]

What happens in ascending auction? (Aside: https://www.youtube.com/watch?v=akwSGr-9Ldc)

What should happen in sealed-bid auction? (What about n bidders?)

Vickrey Auction (KP p. 237, N 9.3.1, R 2)

How should two bidders bid in sealed-bid second-price auction?

What about n bidders?

EVOLUTIONARILY STABLE STRATEGIES - 6319 presentation possibility?

Hawk and Doves (KP p. 137)

player I \ player II	H	D
H	$(\frac{v}{2}-c, \frac{v}{2}-c)$	$(v, 0)$
D	$(0, v)$	$(\frac{v}{2}, \frac{v}{2})$

CORRELATED EQUILIBRIA

If there is intelligent life on other planets, in a majority of them, they would have discovered correlated equilibrium before Nash equilibrium. – Roger Myerson

https://ranger.uta.edu/~weems/NOTES6319/PAPERSONE/papadimitriou.pdf - 6319 presentation possibility?

Concepts (KP p. 143):

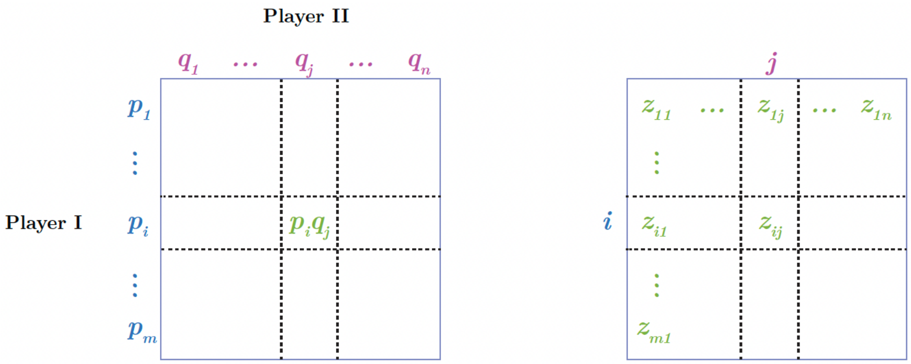


FIGURE 7.4. This figure illustrates the difference between a Nash equilibrium and a correlated equilibrium. In a Nash equilibrium, the probability that player I plays i and player II plays j is the product of the two corresponding probabilities (in this case $p_i q_j$), whereas a correlated equilibrium puts a probability, say z_{ij}, on each pair (i, j) of strategies.

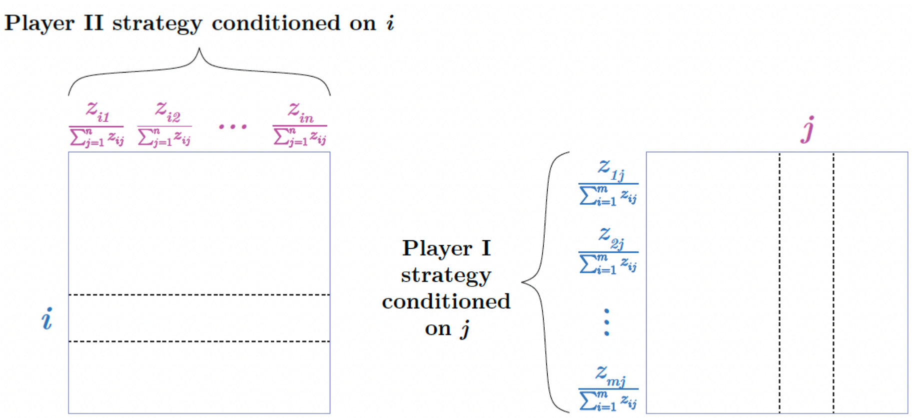


FIGURE 7.5. The left figure shows the distribution player I faces (the labels on the columns) when the correlated equilibrium indicates that she should play i. Given this distribution over columns, Definition 7.2.3 says that she has no incentive to switch to a different row strategy. The right figure shows the distribution player II faces when told to play j.

Battle of the Sexes (KP p. 142; N p. 7)

`https://www.milb.com/frisco` `https://www.santafeopera.org`

		husband	
		opera	baseball
wife	opera	(4,1)	(0,0)
	baseball	(0,0)	(1,4)

Traffic Light (N p. 14; R p. 177, correct values shown here)

1 \ 2	Cross	Stop
Cross	-100, -100	1, 0
Stop	0, 1	0, 0

	Stop	Go
Stop	-1,-1	-1,0
Go	0,-1	-5,-5

Chicken (KP p. 79, 144; N p. 45)

player I \ player II	Swerve (S)	Drive (D)
Swerve (S)	(1, 1)	$(-1, 2)$
Drive (D)	$(2, -1)$	$(-M, -M)$

player I \ player II	Swerve (S)	Drive (D)
Swerve (S)	(6, 6)	(2, 7)
Drive (D)	(7, 2)	(0, 0)

Consider the symmetric game (often called *chicken*) with payoffs

$$\begin{pmatrix} 4 & 1 \\ 5 & 0 \end{pmatrix}$$

	Stop	Go
Stop	4,4	1,5
Go	5,1	0,0

Optimal Baskets of Goods via Network Flow (N 1.8.1)

Trivial Example:

3 buyers ("*B*") { a/\$300, b/\$500, c/\$100 } with indicated budgets

2 interchangeable divisible goods ("*A*") { 0/10, 1/20 } with indicated amounts

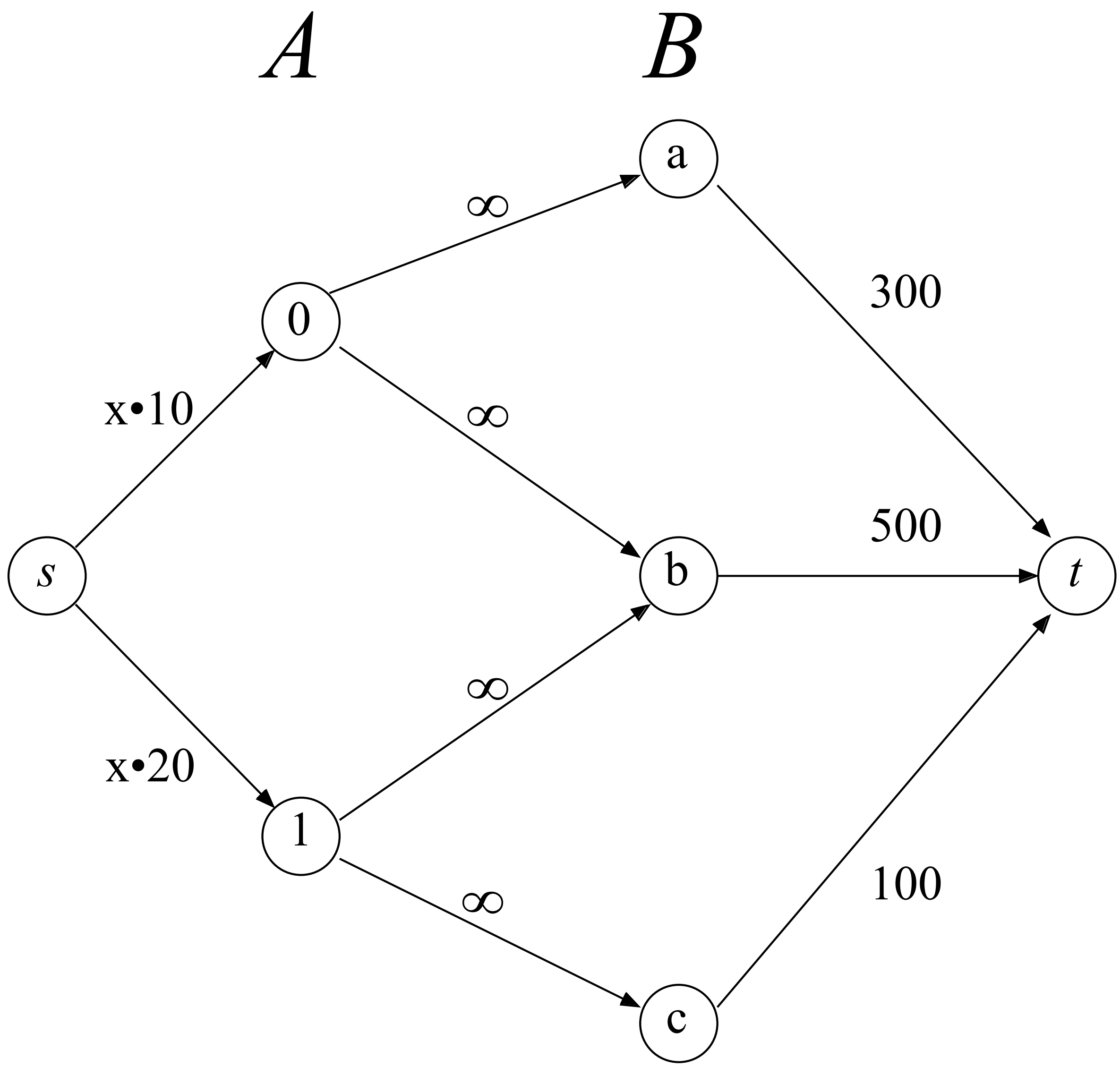
Buyers have limited access to goods (∞ edges between *A* and *B* below)

Buyers prefer lower prices

Determine "market clearing" (AKA equilibrium) prices

Binary search (between 0 and (300+500+100)/(10+20)=900/30=30) gives x^* = \$30 as market clearing price (per unit) for both goods.

x^* is the largest value at which (s, $A \cup B \cup t$) remains a min-cut in N. (N p. 25)



a will get 10 units of 0   b will get 16.667 units of 1   c will get 3.333 units of 1

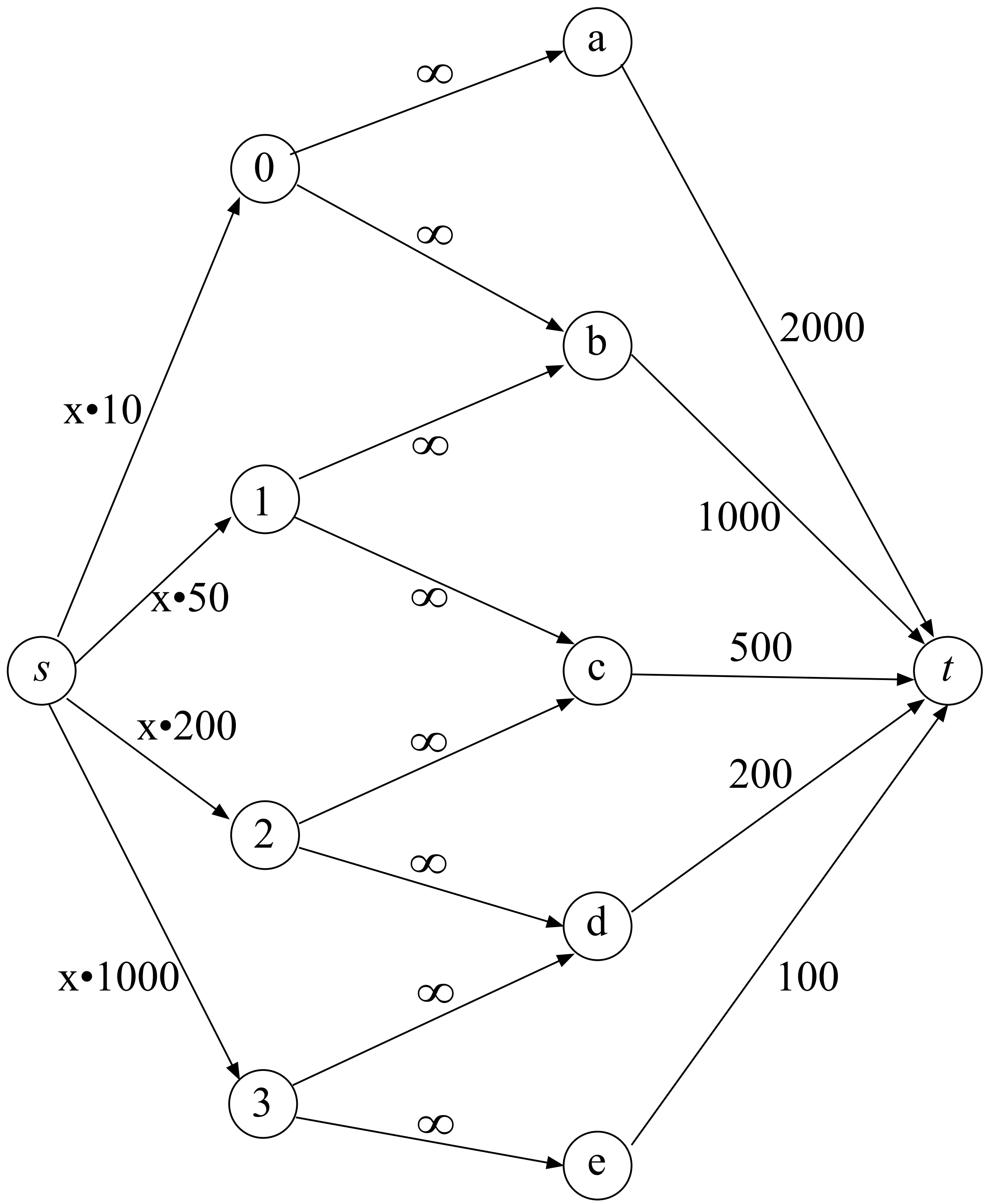
Second Example

5 buyers ("B") { a/\$2000, b/\$1000, c/\$500, d/\$200, e/\$100 } with indicated budgets

4 interchangeable divisible goods ("A") { 0/10, 1/50, 2/200, 3/1000 } with indicated amounts

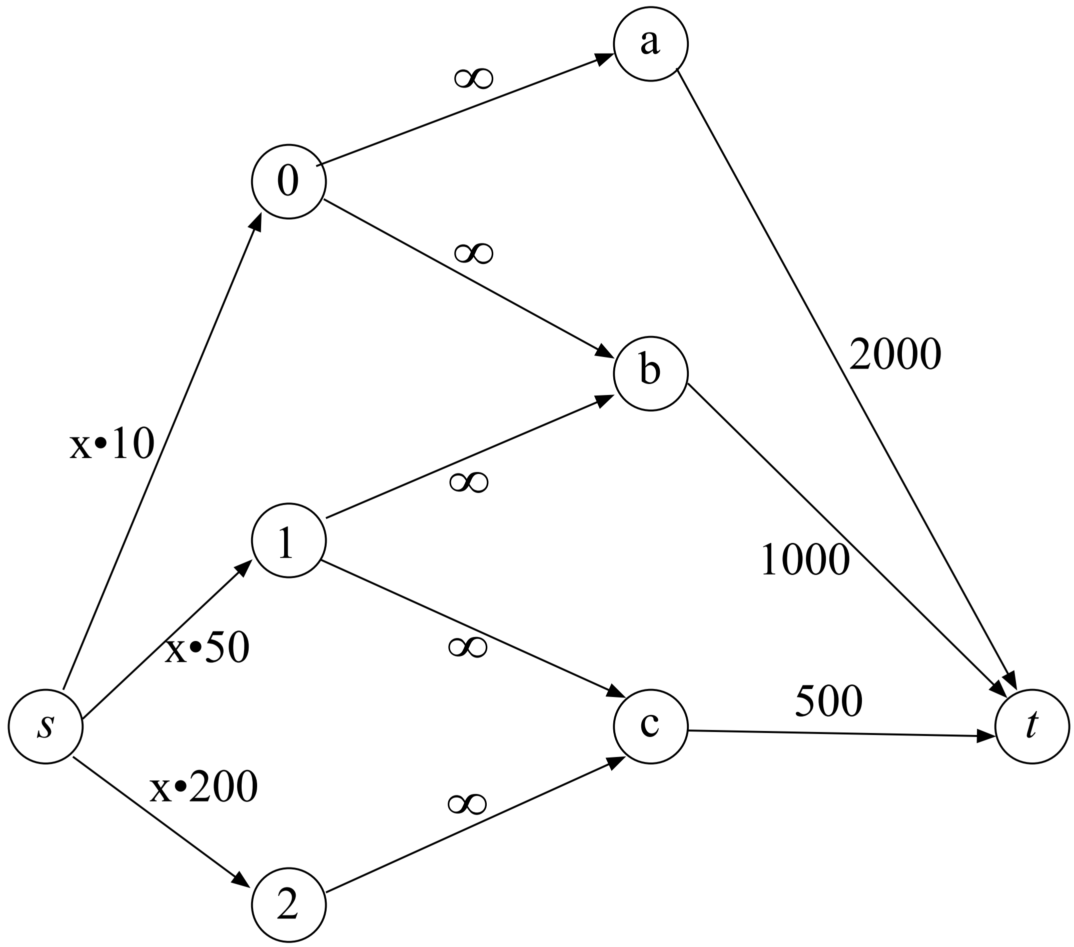
Buyers have limited access to goods (∞ edges between A and B below)

Binary search (between 0 and (2000+1000+500+200+100)/(10+50+200+1000)=3800/1260=3.0159) gives $x^* = \$0.30$ (largest x value that saturates all edges leaving s)



$W = \{ 0, 1, 2, a, b, c \}$ (vertices in residual network with path to t) $S^* = A - W = \{ 3 \}$ $\Gamma(S^*) = \{ d, e \}$

Binary search (between 0.3 and (2000+1000+500)/(10+50+200)=3500/260=13.462) gives x^* = \$2.50 (largest x value that saturates all edges leaving s)

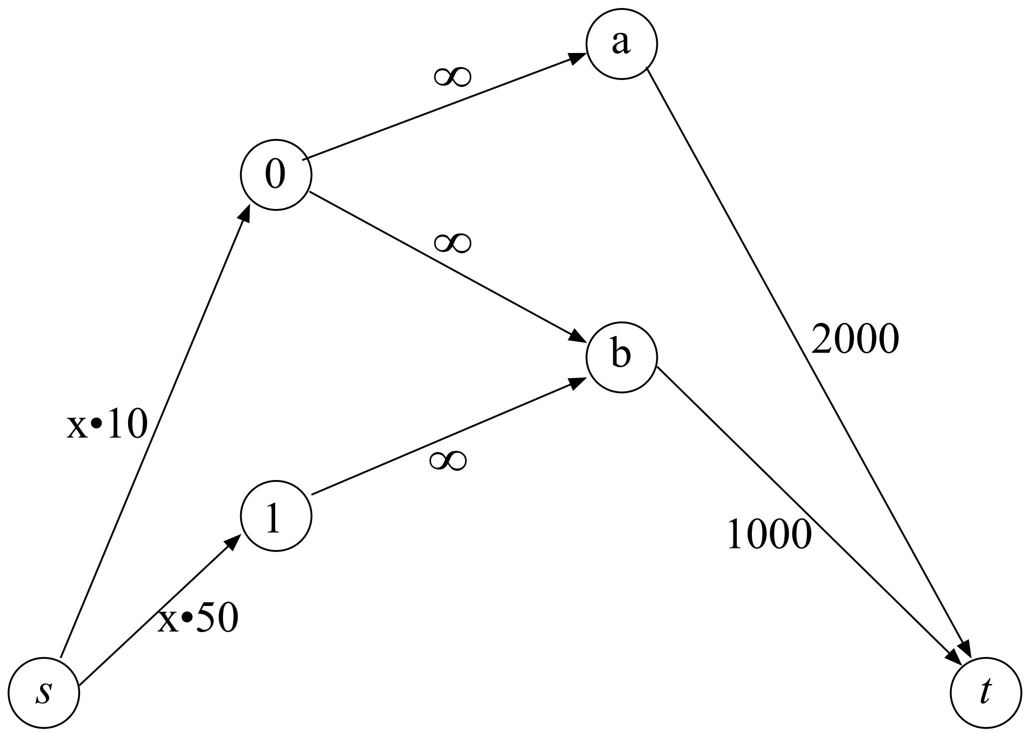


$W = \{ 0, 1, a, b \}$ (vertices in residual network with path to t)

$S^* = A - W = \{ 2 \}$ Remove S^*

$\Gamma(S^*) = \{ c \}$ Remove $\Gamma(S^*)$

Binary search (between 2.5 and (2000+1000)/(10+50)=3000/60=50) gives x^* = \$20 (largest x value that saturates all edges leaving s)

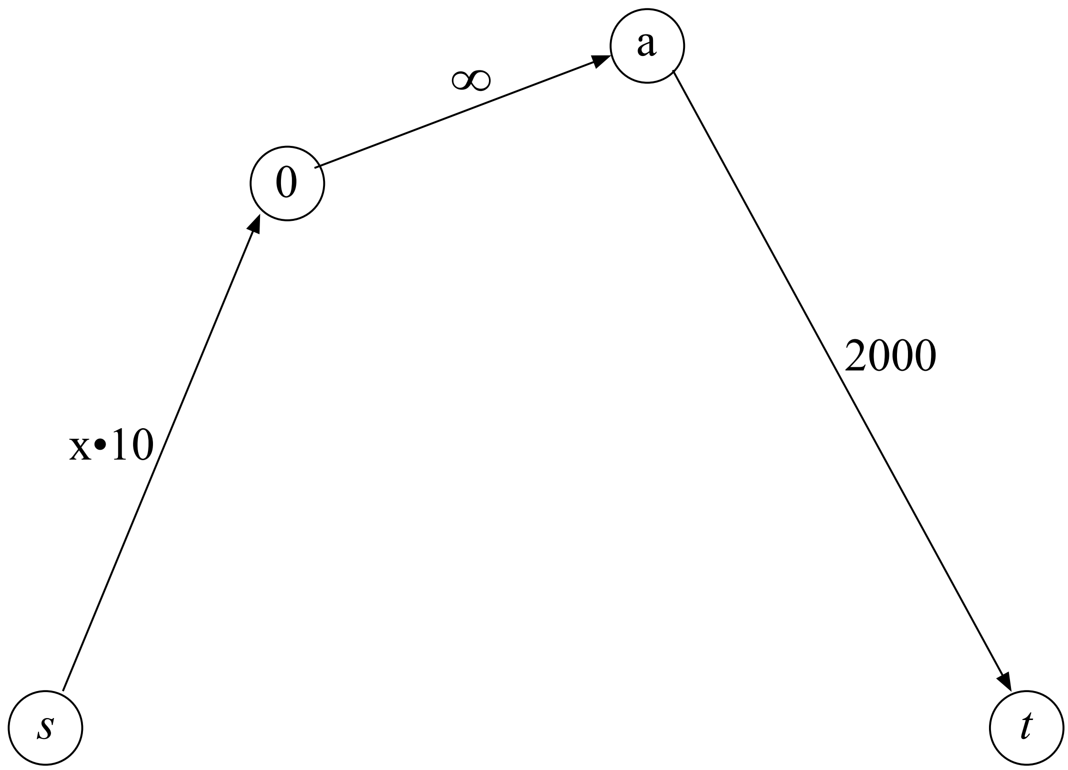


W = { 0, a }
(vertices in residual network with path to t)

$S^* = A - W$ = { 1 }
Remove S^*

$\Gamma(S^*)$ = { b }
Remove $\Gamma(S^*)$

Binary search (between 20 and 2000/10=200) gives x^* = \$200 (largest x value that saturates all edges leaving s)



$W = \varnothing$ $S^* = A - W = \{ 0 \}$ $\Gamma(S^*) = \{ a \}$

Remove S^* Remove $\Gamma(S^*)$

A	Price	*B*	Units
0	\$200	a	10 of 0
1	\$20	b	50 of 1
2	\$2.5	c	200 of 2
3	\$0.3	d	666.67 of 3
		e	333.33 of 3

1.E. LINEAR PROGRAMMING

Protein Problem (KP p. 334) - `protein.kp334.mod`

```
# Karlin/Peres, p. 334
# glpsol --model protein.kp334.mod
var steak; # x1
var peanutButter; # x2
s.t. cost: 4*steak + peanutButter <= 6;
s.t. fat: steak + 2*peanutButter <= 5;
maximize protein: 2*steak + peanutButter;
solve;
printf "steak %10g\n", steak;
printf "peanut butter %10g\n", peanutButter;
printf "cost %10g\n", cost;
printf "fat %10g\n", fat;
printf "protein %10g\n", protein;
end;
```

Simplex Method / Slack Variables / Duality / Farkas's Lemma

Matousek and Gartner is recommended for examples, algorithms, and theory

Two-Person Zero-Sum

`2pers0sum.max.mod 2pers0sum.min.mod`

`2pers0sum.mod`

```
# Computes both players distributions for 2-person, 0-sum Nash equilibrium
# p. 286 of G. B. Dantzig, Linear Programming and Extensions
# Combines 2pers0sum.max.mod and 2pers0sum.min.mod to eliminate the need for
# a min/maximize objective.
# glpsol --model 2pers0sum.mod --data rockSP.kp139.dat
param m, integer, > 0;
param n, integer, > 0;
set I := 1..m; # rows
set J := 1..n; # columns
param a{i in I, j in J}; # input matrix, a[i,j] is payoff for i, -a[i,j] is payoff to j
var x{i in I}, >= 0;
var y{i in J}, >= 0;
var V;
s.t. xsum{i in J}: sum{j in I} a[j,i]*x[j] >= V;
s.t. xprob: sum{i in I} x[i] = 1;
```

```
s.t. ysum{i in I}: sum{j in J} a[i,j]*y[j] <= V;

s.t. yprob: sum{i in J} y[i] = 1;

solve;

printf "\n";
printf "V is %10g\n",V;
printf "X distribution is:\n";
printf{i in I} " (%d %10g)\n",i,x[i];
printf "\n";
printf "Y distribution is:\n";
printf{i in J} " (%d %10g)\n",i,y[i];

end;
```

```
# rock paper scissors

param m := 3;

param n := 3;

param a : 1  2  3 :=
      1   0 -1  1
      2   1  0 -1
      3  -1  1  0 ;

end;
```

```
V is          0
X distribution is:
 (1   0.333333)
 (2   0.333333)
 (3   0.333333)

Y distribution is:
 (1   0.333333)
 (2   0.333333)
 (3   0.333333)
Model has been successfully processed
```

Single Iteration of Optimal Basket of Goods via Network Flow

optBasket.mod   optBasket.2nd.1.dat

```
# Solves one iteration of optimal basket of goods from Nisan, Chapter 1, p. 23 - 26.
# See Figure 1.4 on p. 24.

# Loosely based on glpk-5.0/examples/maxflow.mod

# glpsol --model optBasket.mod --data optBasket.trivial.dat

# using --exact fixes issues with "flowToSink[j] < money[j]" later

param numGoods, integer, > 0;

param numBuyers, integer > 0;

set good := 0..numGoods-1; # Left column A

set buyer := 0..numBuyers-1; # Right column B

param amount{i in good}, >= 0;

param money{i in buyer}, >= 0;

set accessible, within good cross buyer;

var price, >= 0;

var flowFromSource{i in good}, >= 0;

var flowGoodToBuyer{(i,j) in accessible}, >= 0;

var flowToSink{i in buyer}, >= 0;

s.t. checkFlowFrom{i in good}: flowFromSource[i] = price*amount[i];

s.t. checkFlowTo{i in buyer}: flowToSink[i] <= money[i];

s.t. priceCheck: price <= (sum{i in buyer} money[i]) / (sum{j in good} amount[j]);
```

```
s.t. flowConservationGood{i in good}: flowFromSource[i] = (sum{j in buyer} if (i,j) in accessible
then flowGoodToBuyer[i,j]);

s.t. flowConservationBuyer{i in buyer}: flowToSink[i] = (sum{j in good} if (j,i) in accessible
then flowGoodToBuyer[j,i]);

maximize obj: price;

solve;

printf "price is %10g\n",price;

printf "Data for next iteration\n";

# If a good or buyer does not appear, then their respective amount or money should be zeroed.

printf "set accessible :=";
for {(i,j) in accessible: flowToSink[j] < money[j]}
  printf "\n%d %d",i,j;
# The line below reveals numeric issue that can be fixed with --exact
#  printf "\n%d %d %10g %10g",i,j,flowToSink[j],money[j];
printf " ;\n";

end;


# Second optimal basket of goods example in notes01.gtconcepts.doc.

# Input to first iteration for optimalBasket.mod.

param numGoods := 4;

param numBuyers := 5;

set accessible :=
  0 0
  0 1
  1 1
  1 2
  2 2
  2 3
  3 3
  3 4 ;

param amount :=
  0 10
  1 50
  2 200
  3 1000 ;

param money :=
  0 2000
  1 1000
  2 500
  3 200
  4 100 ;

end;

OPTIMAL SOLUTION FOUND
Time used:   0.0 secs
Memory used: 0.1 Mb (129206 bytes)
price is        0.3
Data for next iteration
set accessible :=
0 0
0 1
1 1
1 2
2 2 ;
Model has been successfully processed
```