Game theory and its applications

Levente Buttyán
Laboratory of Cryptography and System Security (CrySyS)
Budapest University of Technology and Economics
buttyan@crysys.hu

Brief introduction to Game Theory

- Discipline aiming at modeling situations in which actors have to make decisions which have mutual, **possibly conflicting**, consequences
- Classical applications: **economics**, but also politics, biology, and recently, networking protocols!
- Example: should a company invest in a new plant, or enter a new market, considering that the **competition** *may* make similar moves?
- Most widespread kind of game: **non-cooperative** (meaning that the players do not attempt to find an agreement about their possible moves)

The Prisoner's Dilemma

- game formulation: G = (P, S, U)
  - P: set of players
  - S: set of strategies
  - U: set of utility (payoff) functions
- players are ***rational*** → they try to maximize their payoff
- strategic-form representation:

Blue \ Green	Confess	Don't confess
Confess	(-7, -7)	(0, -10)
Don't confess	(-10, 0)	(-1, -1)

Solving the Prisoner's Dilemma

- strict dominance:

  a strategy s_i of player i is strictly dominant, if for any other strategy s_i', we have

  $$u_i(s_i, s_{-i}) > u_i(s_i', s_{-i}) \quad \text{for all } s_{-i} \text{ in } S_{-i}$$

  where $u_i()$ is player i's payoff function, and s_{-i} is a strategy profile containing strategies for all players except i

- in the Prisoner's Dilemma, Confess strictly dominates Don't Confess for both players

Blue \ Green	Confess	Don't confess
Confess	(-7, -7)	(0, -10)
Don't confess	(-10, 0)	(-1, -1)

Hawks and Doves

Blue \ Green	Fight	Surrender
Fight	(-100, -100)	(1, -1)
Surrender	(-1, 1)	(0, 0)

- no strictly dominant strategy
- but consider the following:
  - if Blue fights, then the *best response* of Green is to surrender
  - if Green surrenders, then the *best response* of Blue is to fight
  - → (**Fight**, **Surrender**) is an equilibrium in the sense that no player has an incentive to unilaterally deviate (Nash equilibrium)

Solving the Hawks and Doves game

- Nash equilibrium:

  a strategy profile (s_i^*) is a Nash equilibrium, if for every player i and for any strategy $s_i \neq s_i^*$, we have

  $$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$$

- the Hawks and Doves game has two Nash equilibria

Blue \ Green	Fight	Surrender
Fight	(-100, -100)	(1, -1)
Surrender	(-1, 1)	(0, 0)

Pareto optimality and stability

- Pareto optimality
  - an outcome of the game is Pareto optimal, if no player can increase its payoff without hurting some other player
- stability
  - an equilibrium is stable if a change in any player's strategy leads to a situation where:
    - the player who did not change has no better strategy in the new circumstance
    - the player who did change is now playing with a strictly worse strategy

    (if these cases are both met, then the player who changed his strategy will return immediately to the previous equilibrium)
- when there are multiple Nash equilibria, Pareto optimality and stability may be considered as selection criteria
- example: in the Hawks and Doves game, both NEs are Pareto optimal and instable

The Jamming game

Blue \ Green	C_1	C_2
C_1	(-1, 1)	(1, -1)
C_2	(1, -1)	(-1, 1)

- Green is a jammer who wants to destroy Blue's transmission
- there are two channels C1 and C2
- the game is a zero-sum game: successful jamming is good for Green and bad for Blue, while successful transmission is bad for Green and good for Blue

Solving the Jamming game

- there's no pure strategy Nash equilibrium
- mixed strategies:
  - a mixed strategy is defined by a probability distribution $p(s_i)$ that assigns a probability to each strategy of player i
  - when player i plays a mixed strategy it chooses strategy s_i with probability $p(s_i)$
  - in this case, we are interested in the expected payoff of the players
- in the Jamming game, the mixed strategy profile ((1/2, 1/2), (1/2, 1/2)) is a Nash equilibrium
  - when Blue chooses the channel uniformly at random, the jammer Green has no better move than choosing his channel uniformly at random, and vice versa
- Nash theorem (1950): Every finite game has a mixed strategy Nash equilibrium.

Extensive games

<u>Definition</u>: $G = < P, Q, p, (\mathcal{I}_i)_{i \in P}, (\leq_i)_{i \in P} >$

P : set of players

Q : set of action sequences (set of terminal action sequences is Z)

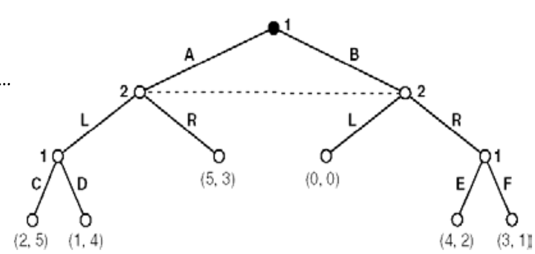
p : player function ($p : Q \setminus Z \rightarrow P$)

$\mathcal{I}_i$: information partition of player i

$\leq_i$: preference relation of player i on Z (often represented by payoffs)

<u>Example</u>

$P = \{1, 2\}$
$Q = \{\varepsilon, A, B, A.L, A.R, \ldots\}$
$p(\varepsilon) = 1, p(A) = p(B) = 2, \ldots$
$\mathcal{I}_1 = \{\{\varepsilon\}, \{A.L\}, \{B.R\}\}$
$\mathcal{I}_2 = \{\{A, B\}\}$
B.L $\leq_1$ A.L.D $\leq_1$ A.L.C …
B.L $\leq_2$ B.R.F $\leq_2$ B.R.E …



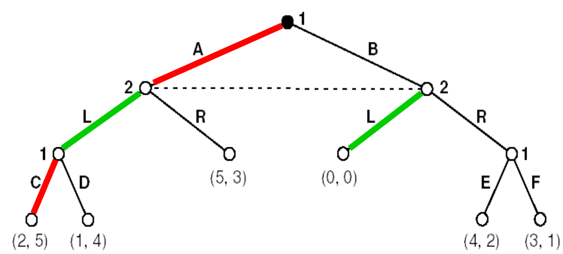
Strategy of player i

Definition: A strategy of player i is a function that assigns an action to every non-terminal action sequence q such that

- it is i's turn to move after q (i.e., $p(q) = i$)
- q is consistent with earlier moves defined by the strategy

with the restriction that the same action must be assigned to q and q' whenever q and q' belong to the same information set of i

Example



Example applications

- modeling software protection as game
  - it turns out that in certain cases, software firms can achieve higher payoff by not protecting their software against piracy
- modeling exchange protocols
  - the concept of rational exchange – strongly related to the concept of Nash equilibrium – can yield efficient exchange protocols with similar properties to fairness

Model

- there are two firms, A and B
- they produce two software packages for price p_A and p_B
- consumers gain extra utility σ from the support provided by the software firms to those customers who pay for the software
- illegal software users cannot obtain support from an independent supplier
- consumers are of two types:
  - type 1 – support-oriented consumers
  - type 2 – support-independent consumers
- the populations of support-oriented and support-independent consumers have the same size, and the total population size is 2 units
- in addition, consumers rank the two software packages differently
  - ranking is represented by a value x between 0 and 1, where a value closer to 0 means preference for software A, and a value closer to 1 means preference for software B
- the distribution of consumers is uniform over the set of all possible ranks

Possible moves and their payoff

- each consumer has 5 possible moves:
  - buy software A
  - buy software B
  - pirate software A
  - pirate software B
  - do not use any software
- number of consumers using software A (legally and illegally) is n_A similarly, number of consumers using software B is n_B
- payoff is increased with an increase in the number of other consumers using the same software package (network externality)

$$U(x,i) \equiv \begin{cases} -x + \mu n_A - p_A + s_i & \text{if buys software } A, \\ -x + \mu n_A & \text{if pirates software } A, \\ -(1-x) + \mu n_B - p_B + s_i & \text{if buys software } B, \\ -(1-x) + \mu n_B & \text{if pirates software } B, \\ 0 & \text{if does not use software,} \end{cases}$$

$$\text{where} \quad s_i \equiv \begin{cases} \sigma, & i = 1, \\ 0 & i = 2, \end{cases} \quad (1)$$

Further notation

- for a given price pair (p_A, p_B) let
  - x_A^ be the support-oriented consumer who is indifferent between buying software A and not buying any software

$$U(\hat{x}_A, 1) = -\hat{x}_A + \mu n_A - p_A + \sigma = 0$$

  - x_B^ be defined similarly
  - y_A^ be the support-independent consumer who is indifferent between pirating software A and not using any software

$$U(\hat{y}_A, 2) = -\hat{y}_A + \mu n_A = 0$$

  - y_B^ be defined similarly
  - x^ be the support-oriented consumer indifferent between software A and B

$$-x + \mu n_A - p_A + \sigma = -(1 - x) + \mu n_B - p_B + \sigma,$$

$$\hat{x} = \frac{1 + \mu(n_A - n_B) + p_B - p_A}{2}$$

The game

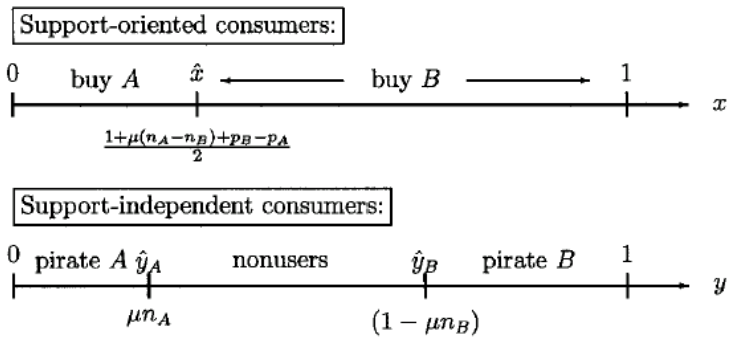
- two stages:
  - stage 1: the two firms set their software price
  - stage 2: consumers make their moves
- solution concept: subgame perfect Nash equilibrium
  - we are looking for strategy profiles that induce a Nash equilibrium in each subgame of the game
- an equilibrium of the second stage subgame is a partition between those who buy software A, who buy software B, who pirate software A, who pirate software B, and who don't use any software, such that no individual would be better off by changing his behavior

LEMMA 1: *Let p_A and p_B be any pair of prices satisfying $p_A, p_B \leq \sigma$. If $\mu < \frac{1}{2}$, then there is an adoption equilibrium such that all support-oriented consumers buy software.*

note: if $\mu > 1/2$, then there's no pure strategy Nash equilibrium in software prices in which both firms sell strictly positive amounts and earn strictly positive profits

→ hence, we will assume that $\mu < 1/2$

Solving the subgame



Lemma 2: *When neither firm protects its software, (a) some support-independent users pirate software A and some pirate software B, and (b) some support-independent consumers do not use any software.*

Solving the subgame

$$n_A = \hat{x} + \hat{y}_A = \frac{1 - \mu n_B - p_A + p_B}{2 - 3\mu},$$

$$n_B = (1-\hat{x}) + (1-\hat{y}_B) = \frac{1 - \mu n_A - p_B + p_A}{2 - 3\mu}$$

solving for n_A and n_B:

$$n_A = \frac{\mu(p_A - p_B - 2) - p_A + p_B + 1}{2(2\mu^2 - 3\mu + 1)}$$

$$n_B = \frac{\mu(p_B - p_A - 2) + p_A - p_B + 1}{2(2\mu^2 - 3\mu + 1)}$$

substituting into the expression of x^:

$$\hat{x}(p_A, p_B) = \frac{\mu(p_A - p_B - 2) - p_A + p_B + 1}{2(1 - 2\mu)}$$

Nash equilibrium in software prices

- firm A chooses p_A to maximize $\pi_A = p_A \hat{x}(p_A, p_B)$
- firm B chooses p_B to maximize $\pi_B = p_B[1 - \hat{x}(p_A, p_B)]$

- best response functions:

$$p_A = R_A(p_B) = \frac{1 - 2\mu}{2(1-\mu)} + \frac{p_B}{2}$$

$$p_B = R_B(p_A) = \frac{1 - 2\mu}{2(1-\mu)} + \frac{p_A}{2}$$

- equilibrium prices and profit levels:

$$p_A^u = p_B^u = \frac{1 - 2\mu}{1-\mu} > 0 \quad \text{and} \quad \pi_A^u = \pi_B^u = \frac{1 - 2\mu}{2(1-\mu)} > 0$$

The game with software protection

- piracy is not possible → consumers must choose between buying the software or not using it

$$\hat{x} = \frac{1 + \mu(n_A - n_B) + p_B - p_A}{2}$$

$$U(\breve{y}_A, 2) = -y_A + \mu n_A - p_A = \breve{0} \quad \rightarrow \hat{y}_A = \mu n_A - p_A$$

$$U(y_B, \breve{2}) = -(1 - y_B) + \mu n_B - \bar{p}_B = \breve{0} \rightarrow \hat{y}_B = 1 - \mu n_B + p_B$$

$$n_A = \hat{x} + \hat{y}_A$$

$$n_B = (1 - \hat{x}) + (1 - \hat{y}_B)$$

$$n_A = \frac{2\mu(2p_A - 1) - 3p_A + p_B + 1}{2(2\mu^2 - 3\mu + 1)} \quad n_B = \frac{2\mu(2p_B - 1) - 3p_B + p_A + 1}{2(2\mu^2 - 3\mu + 1)}$$

Nash equilibrium in software prices

- firm A chooses p_A to maximize $p_A n_A$
- firm B chooses p_B to maximize $p_B n_B$

- best response functions:

$$p_A = R_A(p_B) = \frac{1 - 2\mu + p_B}{2(3 - 4\mu)}$$

$$p_B = R_B(p_A) = \frac{1 - 2\mu + p_A}{2(3 - 4\mu)}$$

- equilibrium prices and profit levels:

$$p_A^p = p_B^p = \frac{1 - 2\mu}{5 - 8\mu} \qquad \pi_A^p = \pi_B^p = \frac{(1 - 2\mu)(3 - 4\mu)}{2(1 - \mu)(5 - 8\mu)^2}$$

Comparison of profit levels

- no protection:

$$\pi_A^u = \pi_B^u = \frac{1 - 2\mu}{2(1 - \mu)}$$

- protection:

$$\pi_A^p = \pi_B^p = \frac{(1 - 2\mu)(3 - 4\mu)}{2(1 - \mu)(5 - 8\mu)^2}$$

- when $\mu < ½$ (as this was assumed), the firms make less profit if they use software protection ($(3\text{-}4\mu) < (5\text{-}8\mu)^2$)

Rational exchange – informal definition

- A misbehaving party cannot gain any advantages
  → Misbehavior is uninteresting and should happen only rarely.

- few rational exchange protocols proposed in the literature
  - Jakobsson's coin ripping protocol
  - Sandholm's unenforced exchange
  - Syverson's rational exchange protocol
- they seem to provide weaker guarantees than fair exchange protocols, but …
- they are usually less complex than fair exchange protocols

→ trade off between complexity and fairness

→ interesting solutions to the exchange problem

Rational exchange – formal definition

Rationality ~ Nash equilibrium

- Rationality: a misbehaving party cannot gain any advantages
- Nash equilibrium: a deviating party cannot gain a higher payoff (given that the other parties do not deviate)

Formal definition of rationality

- protocol: $\pi = \{ \pi_1, \pi_2, \pi_3 \}$
- protocol game: G_π
- each program π_i is represented by a strategy s_i^* in G_π
- the network has a single strategy s_{net}^*
- we consider the restricted protocol game $G_{\pi|s}$,
  where $s = (s_3^*, s_{net}^*)$
- the protocol is rational iff
  - $(s_{1|s}^*, s_{2|s}^*)$ is a Nash equilibrium in $G_{\pi|s}$
  - both players 1 and 2 prefer the outcome of $(s_{1|s}^*, s_{2|s}^*)$ to any other Nash equilibrium in $G_{\pi|s}$

An example: a rational payment protocol

$U \rightarrow V: m_1 = U, V, tid, val, h(rnd), Sig_U(U, V, tid, val, h(rnd))$

$V \rightarrow U: m_2 = srv$

$U \rightarrow V: m_3 = rnd$

if V received m_1 and m_3: $V \rightarrow B: m_4 = m_1, m_3, Sig_V(m_1, m_3)$	B: charges U with val credits V with val
if V received only m_1: $V \rightarrow B: m'_4 = m_1, Sig_V(m_1)$	B: charges U with val

<u>Brief informal analysis</u>

- no fairness, but …
- none of the parties gain any financial advantages by cheating
- needs a TTP (the bank), but …
- the bank is needed anyway to maintain accounts
- it performs the same operations as in any check based payment system

An application: micropayment schemes

PayWord

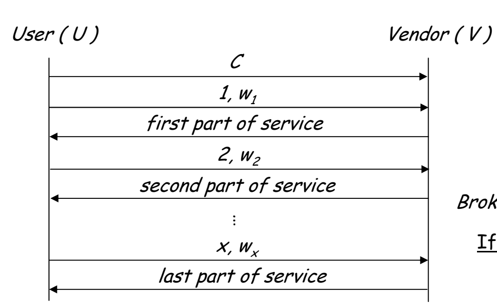
- chain of hash values

$$w_n \longrightarrow w_{n-1} = h(w_n) \longrightarrow w_{n-2} = h(w_{n-1}) \longrightarrow \quad \dots \quad \longrightarrow w_0 = h(w_1)$$

- commitment

$$C = \{ V, cert_U, w_0, exp, data \}_{PrK_U}$$

- protocol



Broker (B):

<u>If</u> (C, x, w_x) <u>then</u>
- charge U with x units;
- pay V x units;

An application: micropayment schemes

Our improvement to PayWord

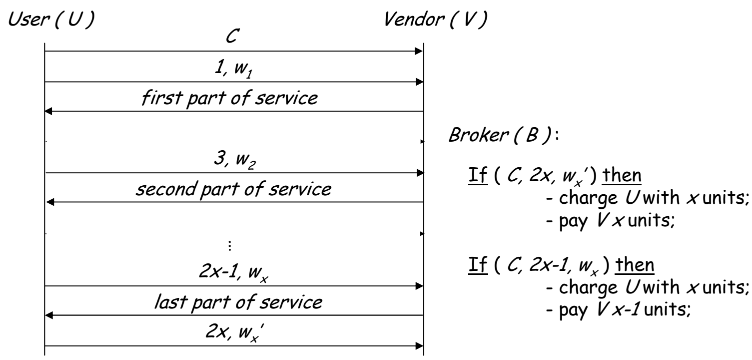
- hash chain size is doubled

$$(w_n' \longrightarrow w_n) \longrightarrow (w_{n-1}' \longrightarrow w_{n-1}) \longrightarrow \ \dots \ \longrightarrow (w_1' \longrightarrow w_1) \longrightarrow w_0'$$

- commitment

$$C = \{ V, cert_U, w_0', exp, data \}_{PrK_U}$$

- protocol



Summary

- Game Theory was invented to analyze situations where parties with potentially conflicting interests are interacting

→ this is the case in many e-commerce applications

- Game Theory has been successfully used to analyze incentives and explain some phenomena in the field of security engineering (see Anderson's work on the Economics of Security)

- a related field is Mechanism Design (Reverse Game Theory) which is concerned with designing games with certain properties (e.g., truthfulness) → interesting direction for research on e-commerce protocol design