The Complexity of Pure Nash Equilibria

Alex Fabrikant
Christos Papadimitriou
Kunal Talwar

CS Division, UC Berkeley

Definitions

- A **game**: a set of *n* players, a set of actions S_i for each player, and a payoff function u_i mapping **states** (combinations of actions) to integers for each player
- A **pure Nash equilibrium**: a state such that no player has an incentive to unilaterally change his action
- A **randomized (or mixed) Nash equilibrium**: for each player, a distribution over his states such that no player can improve his expected payoff by changing his action
- A **symmetric game**: a game with all S_i's equal, and all u_i's identical and symmetric as functions of the other $n-1$ players

Context

- Lots of work studying Nash equilibria:
  - Whether they exist
  - What are their properties
  - How they compare to other notions of equilibria
  - etc.
- **But how hard is it to actually find one?**

Complexity: Randomized NE

- Nash's theorem guarantees existence of randomized NE, so “find a randomized NE” is a total function, and NP-completeness is out of the question, but:
  - Various slight variations on the problem quickly become NP-Complete [Conitzer&Sandholm '03]
  - The two-person case has an interesting combinatorial construction, but with exponential counter-examples [von Stengel '02; Savani&von Stengel '03]
  - It has an “inefficient proof of existence”, placing it in PPAD; other related problems are complete for PPAD, although NE is not known to be [Papadimitriou '94]

Complexity: Pure NE

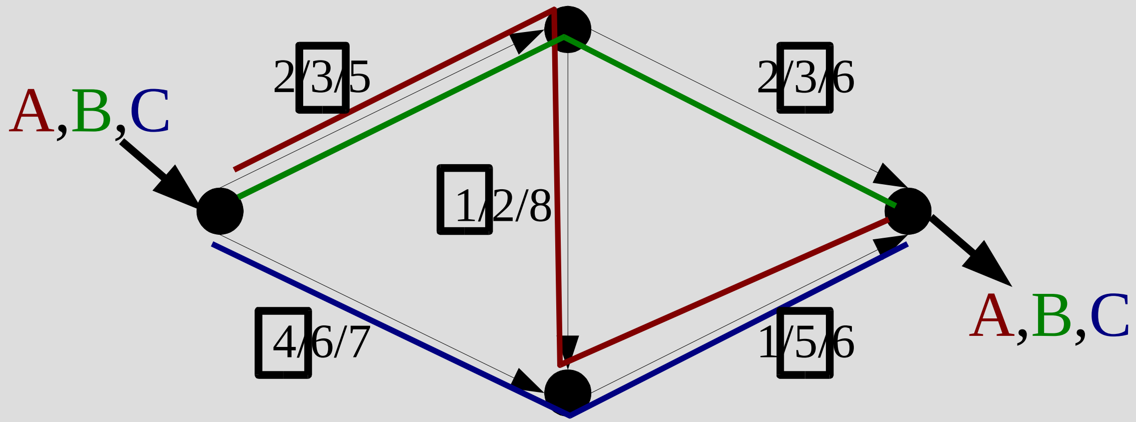
- Natural question: what about pure equilibria?
  - When do they exist?
  - How hard are they to find?

- Immediate problem: with *n* players, explicit representations of the payoff functions are exponential in *n*; brute-force search for pure NE is then linear (on the other hand, fixed #players $\Rightarrow$ boring)

- **Our focus**: The complexity of finding a pure Nash equilibrium in broad *concisely-representable* classes of games

Congestion games

- Well-studied class of games with clear affinity to networks [Roughgarden&Tardos '02, inter alia]



Congestion games (cont)

- **General congestion game**:
  - finite set E of resources
  - non-decreasing delay function: $d: E \times \{1, \ldots, n\} \to \mathbb{Z}$
  - S_i's are subsets of E
  - Cost for a player:

(**number** of players using resource e in state s)

$$\sum_{e \in s_i} d_e(f_s(e))$$

(delay function for resource e)

- **Network congestion game**: each edge is a resource, and each player has a source and a sink, with paths forming allowed strategies

Congestion games & potential functions

- Congestion games have a **potential function**:

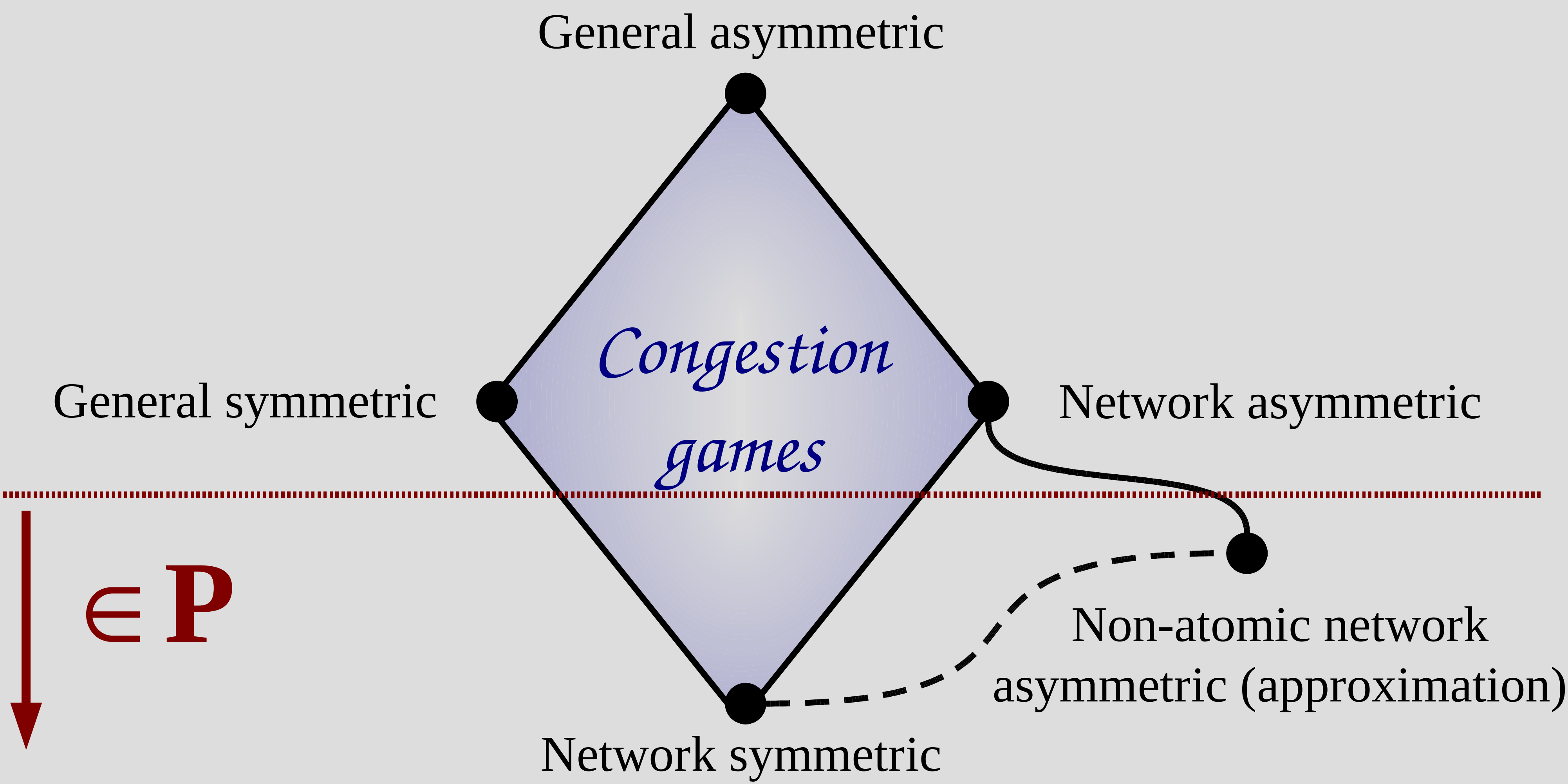
$$\phi(s) = \sum_{e} \sum_{j=1}^{f_s(e)} d_e(j)$$

  If a player changes his strategy, the change in the potential function is **equal** to the change in his payoff

- Local search on potential function guaranteed to converge to a local optimum – an pure NE [Rosenthal '73]

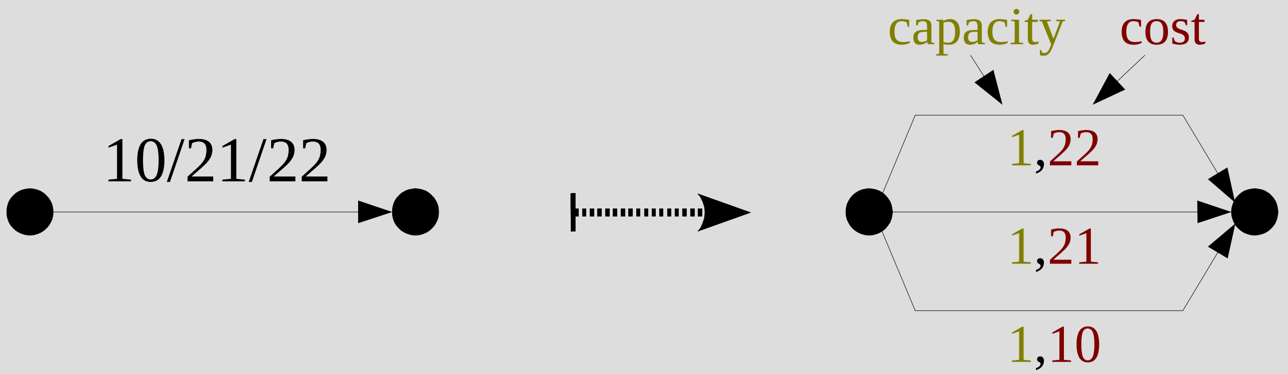
- Note: the potential is *not* the social cost

Our results: upper bounds



Algorithm: symmetric network games

- Reduction to min-cost-flow: transform each edge into n edges, with capacities 1, costs $d_e(1),\ldots,d_e(n)$:



- Integral min-cost flow $\Rightarrow$ local minimum of potential function

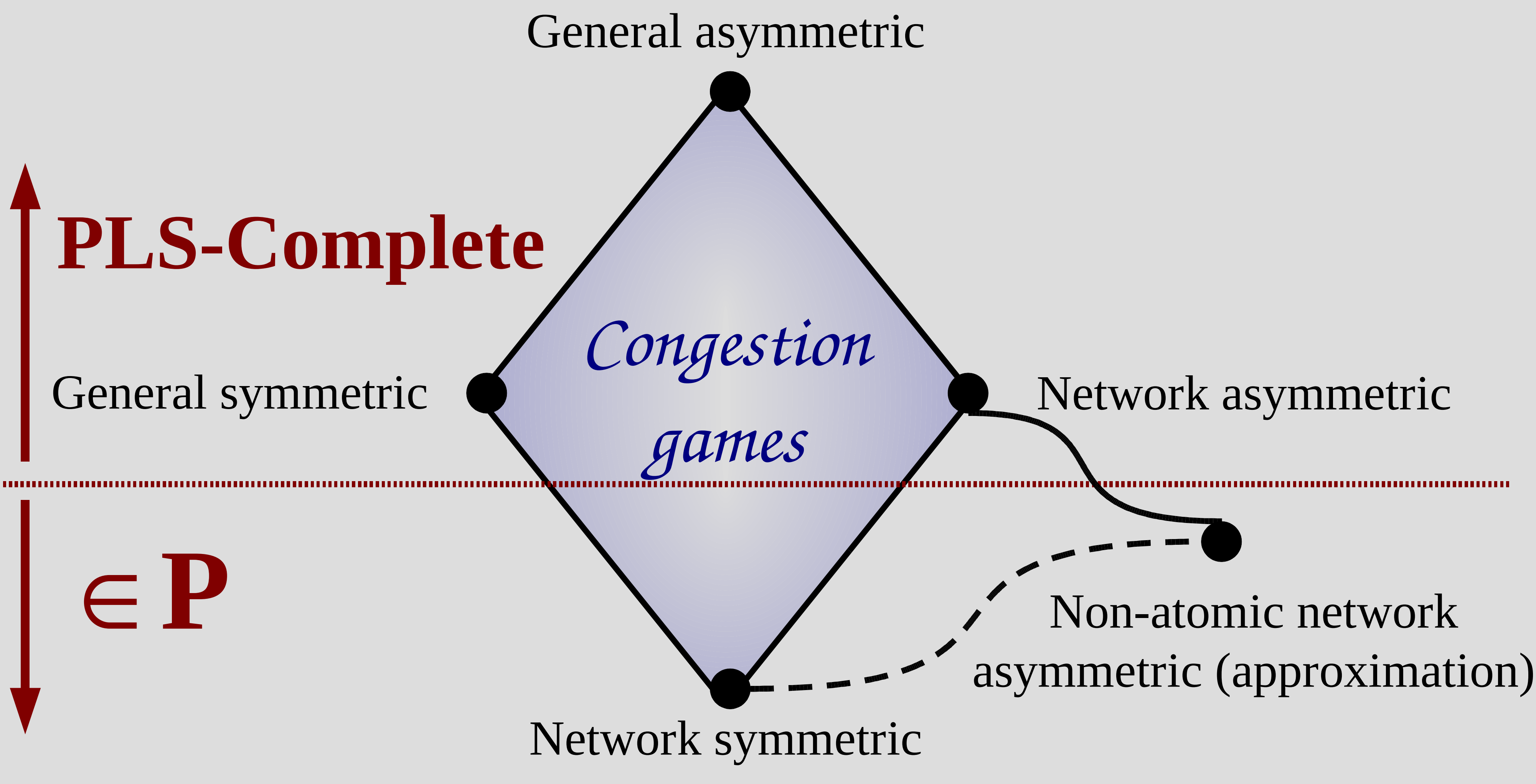
Algorithm: non-atomic games

- [Roughgarden&Tardos '02] studied *non-atomic* congestion games: what happens when $n \rightarrow \infty$ (with continuous delay functions)? Can cast as convex optimization, and thus approximate in polynomial time by the ellipsoid method.

- We modify the above to get, in *strongly* polynomial time, approximate pure Nash equilibria (no player can benefit by $>\varepsilon$) in the *non-atomic asymmetric network case*

- N.B.: Another strongly-polynomial approximation scheme follows from the OR literature, but it is not clear that it produces approximate Nash equilibria

Our results: Lower bounds



P...what?

- **PLS** (polynomial local search [Johnson, et al '88]) – "find *some* local minimum in a reasonable search space":
  - A problem with a search space (a set of feasible solutions which has a neighborhood structure)
  - A poly-time cost function *c(x,s)* on the search space
  - A poly-time function that *g(x,s)*, given an instance *x* and a feasible solution *s*, either returns another one in its neighborhood with lower cost or "none" if there are none
- E.g.: "Find a local optimum of a congestion game's potential function under single-player strategy changes"
- Membership in PLS is an inefficient proof of existence

PLS-Completeness

- PLS reduction:
  $(\text{instance}_A, \text{search space}_A) \rightarrow (\text{instance}_B, \text{search space}_B)$
  Local optima of A must map to local optima of B

- Basic PLS-Complete problem: weighted CIRCUIT-SAT under input bitflips; since [JPY'88], local-optimum relatives of TSP, MAXCUT, SAT shown PLS-Complete

- We mostly use POS-NAE-3SAT (under input bitflips): NAE-3SAT with positive literals only; very complex PLS reduction from CIRCUIT-SAT due to [Schaeffer&Yannakakis '91]

PLS-Completeness: general asymmetric

- POS-NAE-3SAT $\leq_{PLS}$ General Asymmetric CG:

variable x ➠ player x

clause c ➠ resources e_c, e_c'

$$S_x = \{ \overset{x=\text{True}}{\boxed{\{e_c \mid c \ni x\}}}, \overset{x=\text{False}}{\boxed{\{e_c' \mid c \ni x\}}} \}$$

$$d_{e_c}(1) = d_{e_c}(2) = 0 \;\; ; \;\; d_{e_c}(3) = w(c)$$

- Input bitflip maps to a single-player strategy change, with the same change in cost, so search space structure preserved

- General Asymmetric CG $\leq_{PLS}$ General Symmetric CG:
  - "Anonymous" players arbitrarily take on the roles of "non-anonymous" players in the asymmetric game

PLS-Completeness: general symmetric

- General Asymmetric CG $\leq_{PLS}$ General Symmetric CG:
  - Introduce an extra resource r_x for each player x
  - $d_r(1)=0$, $d_r(n>1)=\infty$
  - $S = \bigcup_x \{ s \cup \{r_x\} \mid s \in S_x \}$
- Same number of players, so any solution that uses an r_x twice has an unused r_x, so can't be a local minimum
- Otherwise, players arbitrarily take on the “roles” of players in the original game

PLS-Completeness: network asymmetric

- First guess: make a network following the idea of the general asymmetric reduction – each POS-NAE-3SAT clause becomes two edges, add extra edges so each variable-player traverses either all e_c edges, or all the e_c' edges
- Problem: How do we prevent a player from taking a path that doesn't correspond to a consistent assignment?
- For a dense instance of POS-NAE-3SAT, this appears unavoidable

PLS-Completeness: network asymmetric (cont.)

- But: the Schaeffer-Yannakakis reduction produces a very structured, sparse instance of POS-NAE-3SAT
- Our approach:
  - tweak formulae produced by the S-Y reduction
  - carefully arrange the network so “non-canonical” paths are never a good choice
- Details:
  - 39 variable types
  - 124 clause types
  - 3 more talks today
  - full reduction and a sketch of the proof are in the paper

More on PLS-completeness

- “Clean” PLS reductions: an edge in the original search space corresponds to a short path in the new search space (holds for ours)

- A clean PLS reduction preserves interesting complexity properties (shared by CIRCUIT-SAT, POS-NAE-3SAT, etc):
  - Finding the local optimum reachable from a specific state is PSPACE-complete
  - There are instances with states exponentially far from *any* local optimum

More on potential functions

- Potential functions clearly relevant to equilibria, so: *How applicable is this method?*
- [Monderer&Shapley '96] If *any* game has a potential function, it's equivalent to a (slightly generalized) congestion game
- Party affiliation game: n players, actions: {-1,1}, "friendliness" matrix $\{w_{ij}\}$. Payoff: $p(i) = \mathrm{sgn} \sum_j s_i \cdot s_j \cdot w_{ij}$
- Follow the gradient of $\Phi(s) = \sum_{i,j} s_i \cdot s_j \cdot w_{ij}$ – terminates at a pure NE; but agrees with payoff changes only in sign (and is not a congestion game)

General potential functions

- Define a *general potential function* as one that agrees just in sign with payoff changes under single-player strategy changes (if one exists, there is a pure NE)
- The problem of finding a pure NE in the presence of such a function is clearly in PLS
- **Theorem**: *Any* problem in PLS corresponds to a family of general potential games with polynomially many players; the set of pure Nash equilibria corresponds exactly to the set of local optima

Conclusions

- We have:
  1. Given an efficient algorithm for symmetric network congestion games (and an approximation scheme for the non-atomic asymmetric case)
  2. Shown PLS-completeness of both extensions (asymmetry and general congestion game form); “clean” reductions imply other complexity results
  3. Characterized a link between PLS and general potential games

- Congestion games are thus as hard as any other game with pure NEs guaranteed by a general potential function

Open problems

- Other classes of games where the Nash dynamics converges:
  - Via general potential functions:
    - Basic utility games in [Vetta '02]
    - Congestion games with player-specific delays [Fotakis, et al '02]
  - An algebraic argument shows that the union of 2 games with pure NE's, under some conditions, retains pure NE's
- Acyclic Nash dynamics guarantees *some* potential function (toposort the solution space), but is there always a *tractable* one?
- Pointed out yesterday [Wigderson, yesterday]: complexity classification of games?