1 Complexity, Games, Polytopes and Gale Strings

1.1 Some Complexity Classes

A *computational problem* is given by the combination of an *input* and a related *output*. A specific input gives an *instance* of the problem.

Computational problems can be classified according to the form of their output: for instance, the output of *decision problems* is either "YES" or "NO". A *function problem*, on the other hand, has a more generic output y related to the input x by a $R(x, y)$ given by the problem.

An example of decision problem could be "(input) given a graph, (question) is it possible to find an *Euler tour*, that is, a path through the input graph that starts and end at the same vertex that traverses each edge exactly once?" On the other hand, a function problem could be "(input) given a graph, (output) return an Euler tour."

check: search problems - def

Search problems return either an output y satisfying a given relation $R(x, y)$, where x is the input of the problem, or "NO", if it's not possible to find any such y. If y is guaranteed to exist, the problem is called a *total function problem*. *Counting problems*, finally, return the *number* of y's that satisfy $R(x, y)$; given a problem R we denote the associated counting problem $\#R$.

Computational problems are also classified according to their *computational complexity*, given by the *reducibility* from each other.

Turing machines: here

Let P_1 be a computational problem. For an instance x of P_1, let $|x|$ be the the number of bits needed to encode x. *P_1 reduces to the problem P_2 in polynomial time*, denoted $P_1 \leq_P P_2$, if there exists a *polynomial-time reduction*, that is, a function $f : \{0,1\}^* \to \{0,1\}^*$ and a *Turing machine* $\mathcal{M}$ such that for all $x \in \{0,1\}^*$

- $x \in P_1 \iff f(x) \in P_2$

- $\mathcal{M}$ computes $f(x)$
- $\mathcal{M}$ stops after $p(|x|)$ steps, where p is a polynomial

The complexity class P contains all the polynomially decidable problems, that is, all problems P such that there exists a Turing machine $\mathcal{M}$ that outputs either "YES" or "NO" for all inputs $x \in \{0,1\}^*$ of P after $p(|x|)$ steps, where p is a polynomial. problems in P are often described as *efficient*. The class FP of all the function problems that can be solved in polynomial time is analogously defined.

NP - finding certificates
counting certificates of NP: # P
FNP / TFNP $\Rightarrow$ PPA(D)

1.2 Normal Form Games and Nash Equilibria

until here

1.3 Bimatrix Games and Best Response Polytopes

file: polytopes-subsection

1.4 Cyclic Polytopes and Gale Strings

1.5 Labeling and the Problem Another Gale

file: gale-def-subsection

References

[1] M. M. Casetti, J. Merschen, B. von Stengel (2010). Finding Gale Strings. *Electronic Notes in Discrete Mathematics*

issue, pp. n–m.

[2] X. Chen, X. Deng (2006). Settling the complexity of two-player Nash equilibrium. *Proc. 47th FOCS*, pp. 261–272.

[3] C. Daskalakis, P. W. Goldberg, C. H. Papadimitriou (2006). The complexity of computing a Nash equilibrium. *Proc. Ann. 38th STOC*, pp. 71–78.

change ref to econometrica(?)

[4] J. Edmonds (1965). Paths, trees, and flowers. *Canad. J. Math.* 17, pp. 449–467.

[5] D. Gale (1963), Neighborly and cyclic polytopes. *Convexity, Proc. Symposia in Pure Math.*, Vol. 7, ed. V. Klee, American Math. Soc., Providence, Rhode Island, pp. 225–232.

check if right typography

[6] J. Merschen (2012).

thesis

[7] C. E. Lemke, J. T. Howson, Jr. (1964). Equilibrium points of bimatrix games. *J. Soc. Indust. Appl. Mathematics* 12, pp. 413–423.

[8] C. H. Papadimitriou (1994). On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. System Sci.* 48, pp. 498–532.

[9] R. Savani, B. von Stengel (2006). Hard-to-solve bimatrix games. *Econometrica* 74, pp. 397–429.

[10] L. Végh, B. von Stengel

ref