How Hard is Competition for Rank?

Paul W. Goldberg

[1]Department of Computer Science
University of Liverpool, U. K.

University of Warwick
April 2010

Easy and hard problems

Input/output problems (e.g.: given a game, compute a NE): Some inputs are larger than others. Of course, larger inputs should be allowed to use more processing time.

Easy and hard problems

Input/output problems (e.g.: given a game, compute a NE): Some inputs are larger than others. Of course, larger inputs should be allowed to use more processing time.

Efficient algorithms

If the time taken by an algorithm is proportional to n (the input size), or n^2 or n^3 etc then the algorithm is "efficient" or "fast"; if it is something like 2^n then it is not efficient.

Easy and hard problems

Input/output problems (e.g.: given a game, compute a NE): Some inputs are larger than others. Of course, larger inputs should be allowed to use more processing time.

Efficient algorithms

If the time taken by an algorithm is proportional to n (the input size), or n^2 or n^3 etc then the algorithm is "efficient" or "fast"; if it is something like 2^n then it is not efficient.

A problem is "tractable" if it has an efficient algorithm, otherwise it is "hard" or "intractable".

Easy and hard problems

Input/output problems (e.g.: given a game, compute a NE): Some inputs are larger than others. Of course, larger inputs should be allowed to use more processing time.

Efficient algorithms

If the time taken by an algorithm is proportional to n (the input size), or n^2 or n^3 etc then the algorithm is "efficient" or "fast"; if it is something like 2^n then it is not efficient.

A problem is "tractable" if it has an efficient algorithm, otherwise it is "hard" or "intractable".

To prove problem P is hard, take a problem H that is already believed to be hard, and "efficiently encode" instances of H in terms of P so that the answer to P tells you the answer to H ...

Easy and hard problems

NP-complete problems: hard problems that encode *CIRCUIT SAT* (given a boolean circuit with one output, find an input vector that causes the output to be TRUE)

Easy and hard problems

NP-complete problems: hard problems that encode *CIRCUIT SAT* (given a boolean circuit with one output, find an input vector that causes the output to be TRUE)

NASH

Given a game, find a Nash equilibrium

We believe NASH is hard, but it is due to Nash's theorem that we "can't" encode CIRCUIT SAT in terms of NASH!

Easy and hard problems

NP-complete problems: hard problems that encode *CIRCUIT SAT* (given a boolean circuit with one output, find an input vector that causes the output to be TRUE)

NASH

Given a game, find a Nash equilibrium

We believe NASH is hard, but it is due to Nash's theorem that we "can't" encode CIRCUIT SAT in terms of NASH!We settle for **PPAD**-completeness...

END OF (THE) LINE (Papadimitriou 1991)

Given a graph G of indegree/outdegree at most 1, and a vertex of degree 1, find another vertex of degree 1.

Easy and hard problems

NP-complete problems: hard problems that encode *CIRCUIT SAT* (given a boolean circuit with one output, find an input vector that causes the output to be TRUE)

NASH

Given a game, find a Nash equilibrium

We believe NASH is hard, but it is due to Nash's theorem that we "can't" encode CIRCUIT SAT in terms of NASH!We settle for **PPAD**-completeness...

END OF (THE) LINE (Papadimitriou 1991)

Given a graph G of indegree/outdegree at most 1, and a vertex of degree 1, find another vertex of degree 1.The catch is, G 's edges are represented by boolean circuits that take any pair of endpoints in $\{0,1\}^n$ and output whether an edge is present between them.

Overview

- Nash equilibria are "hard" to find

Overview

- Nash equilibria are “hard” to find
- Try looking for other solution concepts, e.g. correlated equilibria, approximate Nash equilibria
- Or, look for algorithms that are efficient and apply to limited kinds of game

Overview

- Nash equilibria are “hard” to find
- Try looking for other solution concepts, e.g. correlated equilibria, approximate Nash equilibria
- Or, look for algorithms that are efficient and apply to limited kinds of game

This talk

- Some intuition on the hardness of unrestricted NE
- A class of games that appears to be “realistic” for which we so far have some positive results

The "Dragons' Den" Game

Two entrepreneurs, Alice and Bob, want to raise £100,000 from a venture capitalist. Each of them may decide to spend £2,000 on image consulting. Alice has a better business idea, and the only way Bob will receive the investment is if he buys the image consulting and Alice does not.

Question: which of them will buy the image consulting?

The "Dragons' Den" Game

Two entrepreneurs, Alice and Bob, want to raise £100,000 from a venture capitalist. Each of them may decide to spend £2,000 on image consulting. Alice has a better business idea, and the only way Bob will receive the investment is if he buys the image consulting and Alice does not.

Question: which of them will buy the image consulting?

look for *mixed (randomised) strategies*; the problem becomes: *compute the probabilities*

		Bob	
		don't spend 5000	spend 5000
Alice	spend 5000	0 45	-5 45
	don't spend 5000	0 50	45 0

Numbers are multiples of £5,000; assume it is worth £50,000 to win the investment.

"Incentive direction" of the players

Bob

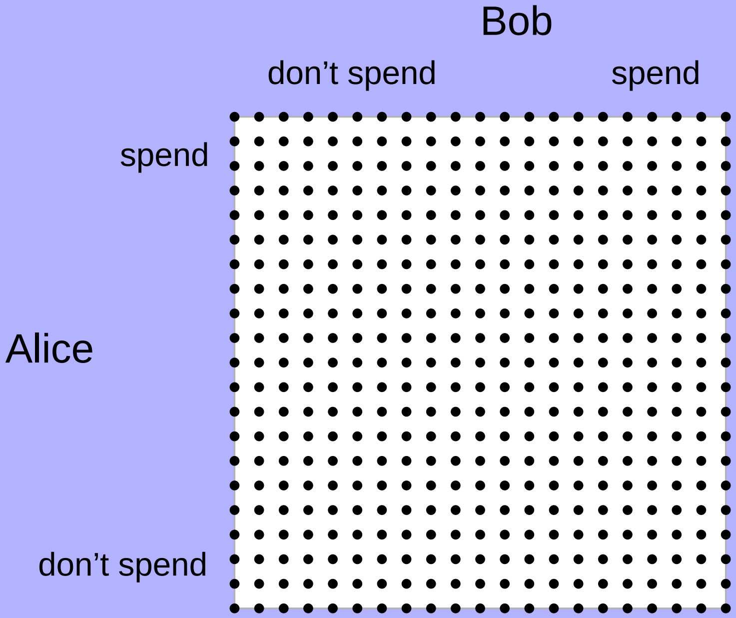
don't spend spend

Alice

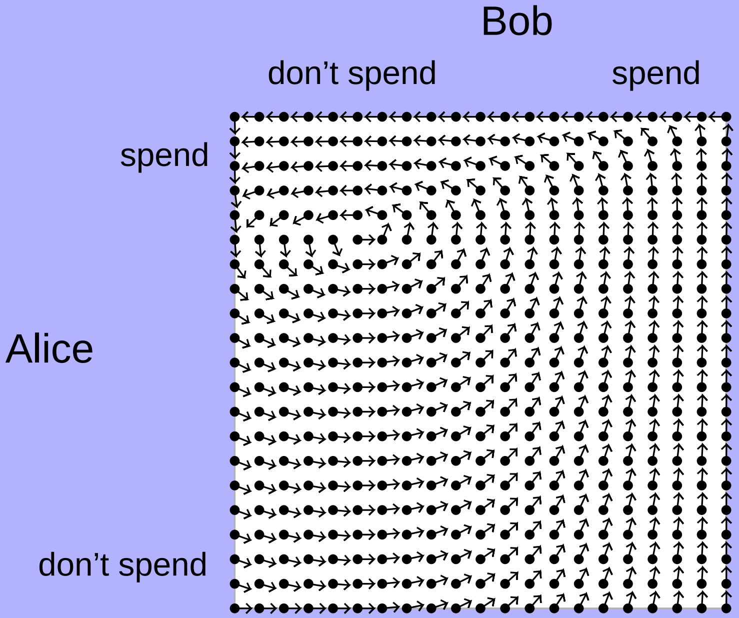
spend

don't spend

“Incentive direction” of the players



"Incentive direction" of the players



Nash equilibrium

Brouwer's fixpoint theorem: continuous functions from a compact domain to itself, have fixpoints. A *non-constructive* proof.

L.E.J. Brouwer (1881-1966)

Nash equilibrium

Brouwer's fixpoint theorem: continuous functions from a compact domain to itself, have fixpoints.
A *non-constructive* proof.
Nash's theorem: using Brouwer's FPT, there always exists a solution,
provided that players may randomize (any number of players, any number of actions).

John Forbes Nash

Nash equilibrium

Brouwer's fixpoint theorem: continuous functions from a compact domain to itself, have fixpoints.
A *non-constructive* proof.
Nash's theorem: using Brouwer's FPT, there always exists a solution,
provided that players may randomize (any number of players, any number of actions).

- standard notion of "outcome of the game"

John Forbes Nash

Nash equilibrium

Brouwer's fixpoint theorem: continuous functions from a compact domain to itself, have fixpoints.
A *non-constructive* proof.
Nash's theorem: using Brouwer's FPT, there always exists a solution,
provided that players may randomize (any number of players, any number of actions).

- standard notion of "outcome of the game"
- each player is receiving optimal expected payoff in the context of the other players' choices.

John Forbes Nash

Nash equilibrium

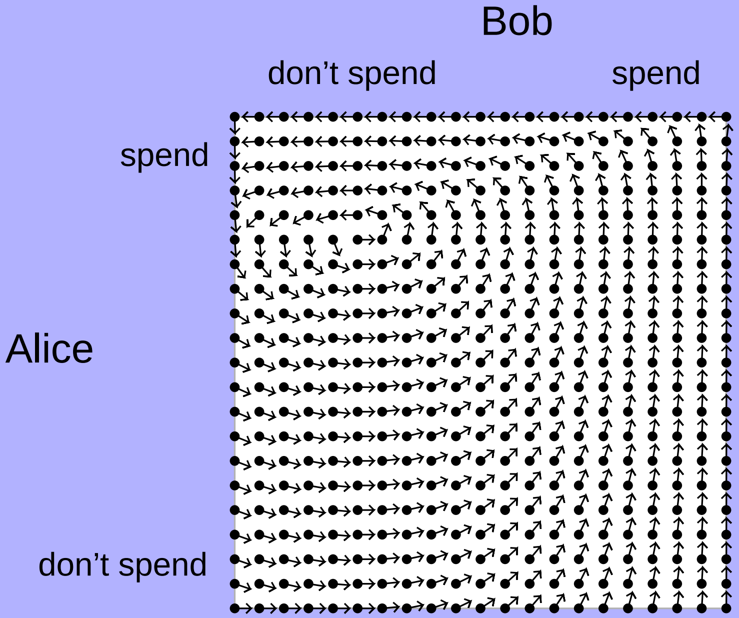
Brouwer's fixpoint theorem: continuous functions from a compact domain to itself, have fixpoints.
A *non-constructive* proof.
Nash's theorem: using Brouwer's FPT, there always exists a solution,
provided that players may randomize (any number of players, any number of actions).

- standard notion of "outcome of the game"
- each player is receiving optimal expected payoff in the context of the other players' choices.

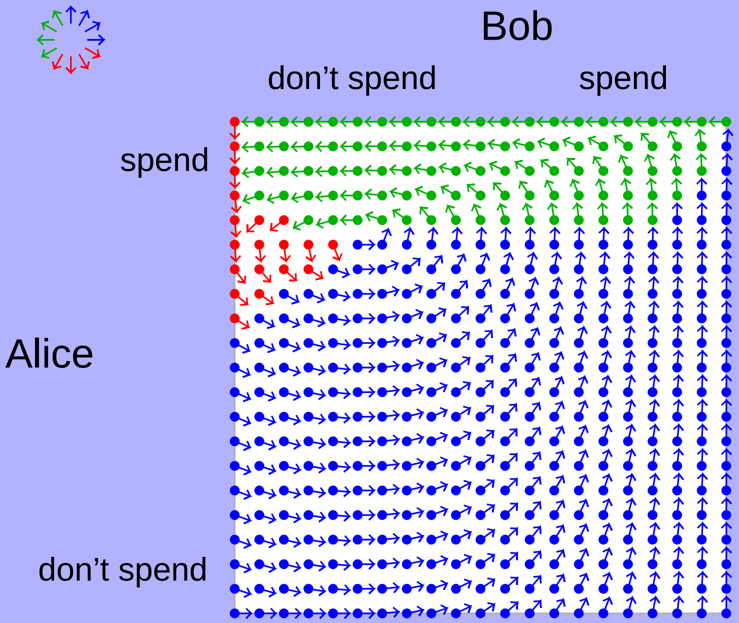
But, how to compute the probabilities? We would like an "efficient algorithm". **Next: how search for NE relates to search on large graphs**

John Forbes Nash

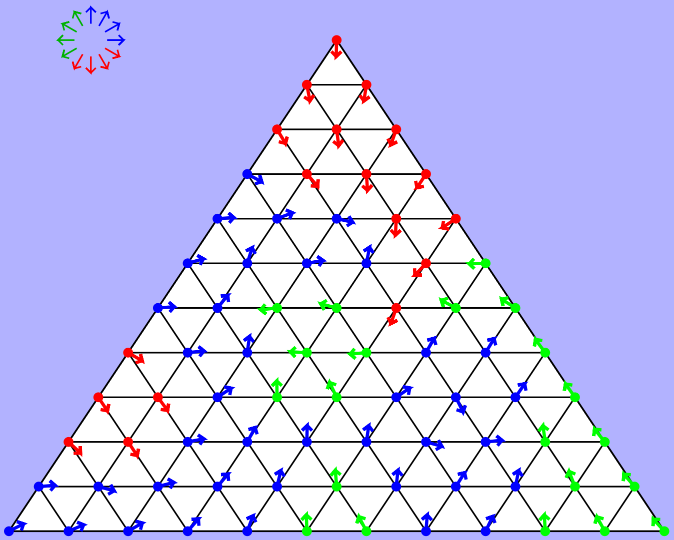
"Incentive direction" of the players



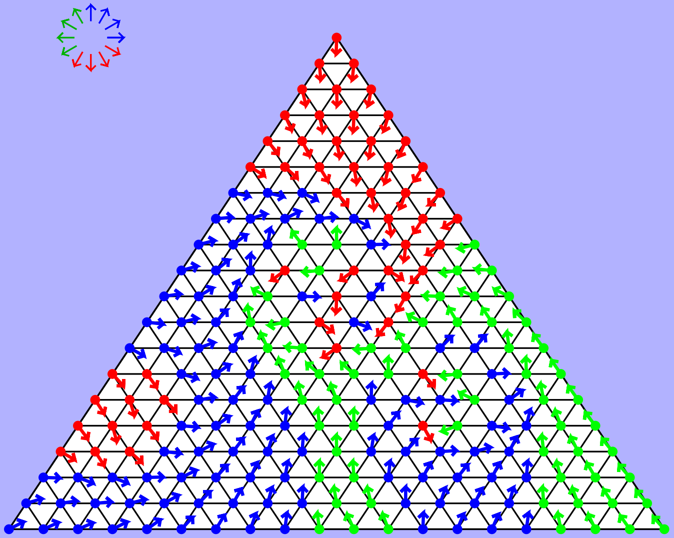
“Incentive direction”, colour-coded



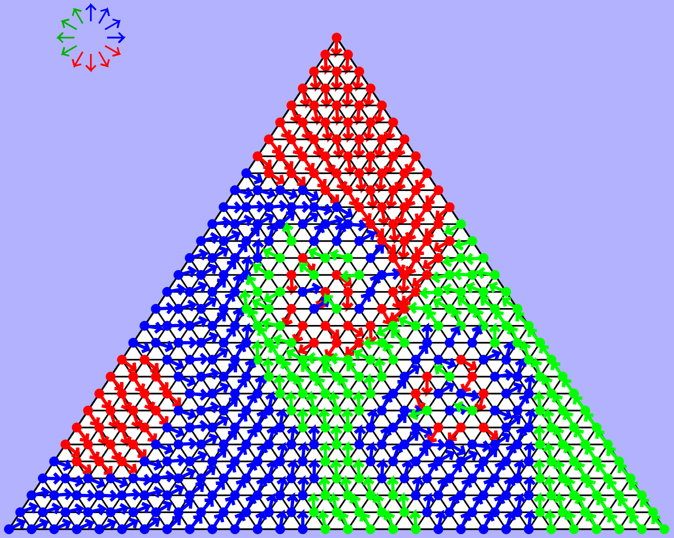
Now, pretend this triangle is high-dimension domain



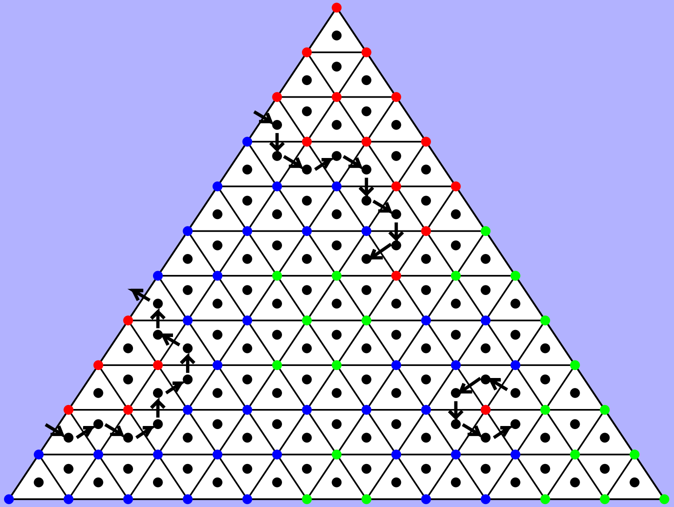
Search for “trichromatic triangles” at higher resolution...



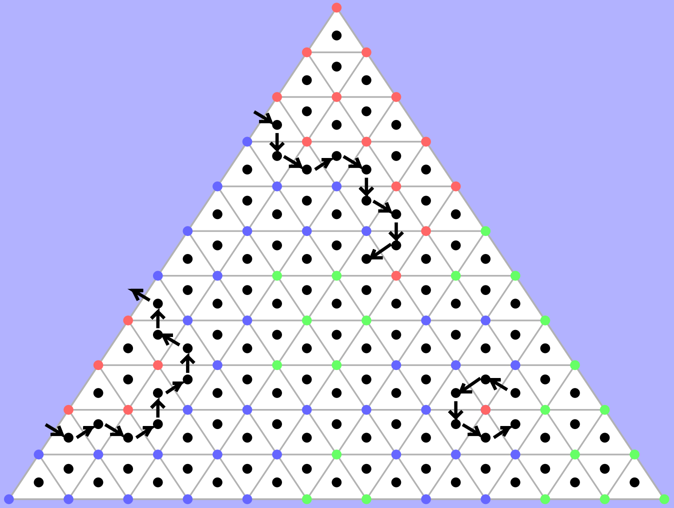
...converges to Brouwer fixpoint



The corresponding graph



The corresponding graph



From graph search to NE computation

- Papadimitriou (1991): generic "END OF LINE" graph search problems seem to be hard

From graph search to NE computation

- Papadimitriou (1991): generic "END OF LINE" graph search problems seem to be hard
- They can encode/represent the difficulty of finding fixpoints of certain Brouwer functions.

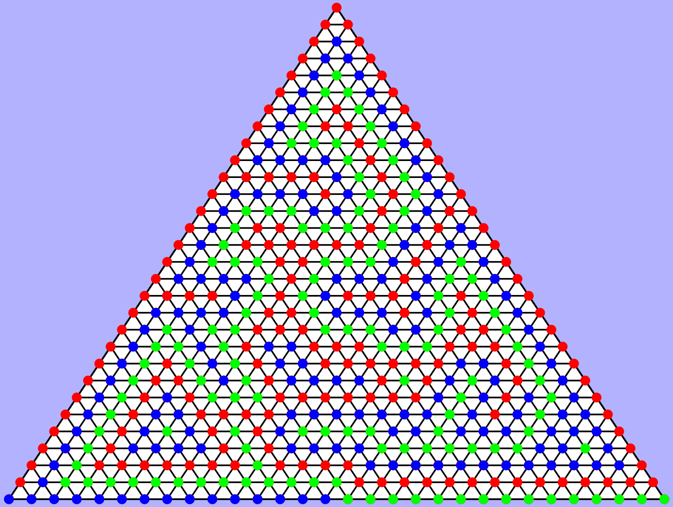
From graph search to NE computation

- Papadimitriou (1991): generic "END OF LINE" graph search problems seem to be hard
- They can encode/represent the difficulty of finding fixpoints of certain Brouwer functions.
- Daskalakis, G and Papadimitriou (2005-6) show that games can also represent/encode a class of Brouwer functions which themselves encode END OF LINE graph search.
  Basically, solving a game is equivalent to finding your way around a very large graph, one that allows efficient local exploration and consists of long paths.

From graph search to NE computation

- Papadimitriou (1991): generic "END OF LINE" graph search problems seem to be hard
- They can encode/represent the difficulty of finding fixpoints of certain Brouwer functions.
- Daskalakis, G and Papadimitriou (2005-6) show that games can also represent/encode a class of Brouwer functions which themselves encode END OF LINE graph search.
  Basically, solving a game is equivalent to finding your way around a very large graph, one that allows efficient local exploration and consists of long paths.
- 2-players (Chen, Deng and Teng '06); 2-players, 0/1-valued payoffs (Abbott, Kane and Valiant '05)

How to make a hard case of the problem



Chen and Deng ('06, '09): 2D-Brouwer

coming back to “Dragons’ Den”

(Current work with colleagues at Liverpool)

(Current work with colleagues at Liverpool)
What if there are

- more than 2 competitors?

coming back to “Dragons’ Den”

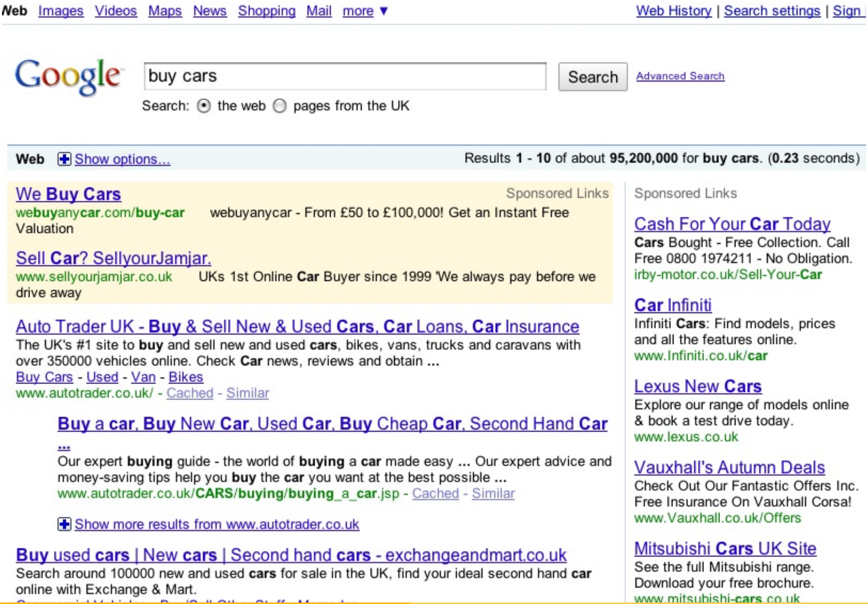
(Current work with colleagues at Liverpool)

What if there are

- more than 2 competitors?
- many choices per competitor?
- more than one “prize” for winning?

Players compete for *rank*.

Competition for rank



Competition for rank

- Who's Top Ten for their Subject 2010
- University League Table
- Methodology & Notes
- Universities by region
- University League Table 2009
- University League Table 2008

www.unistats.com

www.cem.ac.uk

University League Table 2010

Published April 30th 2009

Create your **own customised ranking**, see the device below the main table.

Compare | Clear

To compare 2 or more universities, select the box next to the name and click Compare.
To create your own ranking see below

Rank	Institution	Student Satisfaction	Research Assesment	Entry Standards	Student-Staff Ratio	Academic Services Spend	Facilities Spend	Good Honours	Graduate Prospects
1	The original Complete University Guide ranking	4.1	3.0	523.9	10.9	2,639	587	91.1	82.3
2	Cambridge	4.2	3.0	538.5	11.6	1,755	657	87.0	85.5
3	Imperial College	3.7	2.9	489.4	10.3	3,036	575	68.5	88.4
4	London School of Economics	3.8	3.0	483.0	14.1	1,391	265	76.0	90.6
5	Durham	4.0	2.7	459.0	15.2	1,036	682	77.5	78.3
6	Warwick	3.9	2.8	463.0	13.5	1,630	354	79.7	79.2
7	St Andrews	4.2	2.7	467.9	12.3	1,152	357	85.1	77.8

Search

CITY UNIV LONDON

- Economi
- Internati Politics
- Psycholo
- Sociolog

Web | ettings | Sign

Web .23 seconds)

We web Valu

Sel www drive away

Today lection. Call Obligation. our-Car

[Auto Trader UK - **Buy** & Sell New & Used **Cars**, **Car** Loans, **Car** Insurance](#)
The UK's #1 site to **buy** and sell new and used **cars**, bikes, vans, trucks and caravans with over 350000 vehicles online. Check **Car** news, reviews and obtain ...
Buy Cars - Used - Van - Bikes
www.autotrader.co.uk/ - Cached - Similar

[**Buy** a **car**, **Buy** New **Car**, Used **Car**, **Buy** Cheap **Car**, Second Hand **Car** ...](#)
Our expert **buying** guide - the world of **buying** a **car** made easy ... Our expert advice and money-saving tips help you **buy** the **car** you want at the best possible ...
www.autotrader.co.uk/**CARS**/**buying**/**buying**_a_**car**.jsp - Cached - Similar

Show more results from www.autotrader.co.uk

[**Buy** used **cars** | New **cars** | Second hand **cars** - exchangeandmart.co.uk](#)
Search around 100000 new and used **cars** for sale in the UK, find your ideal second hand **car** online with Exchange & Mart.

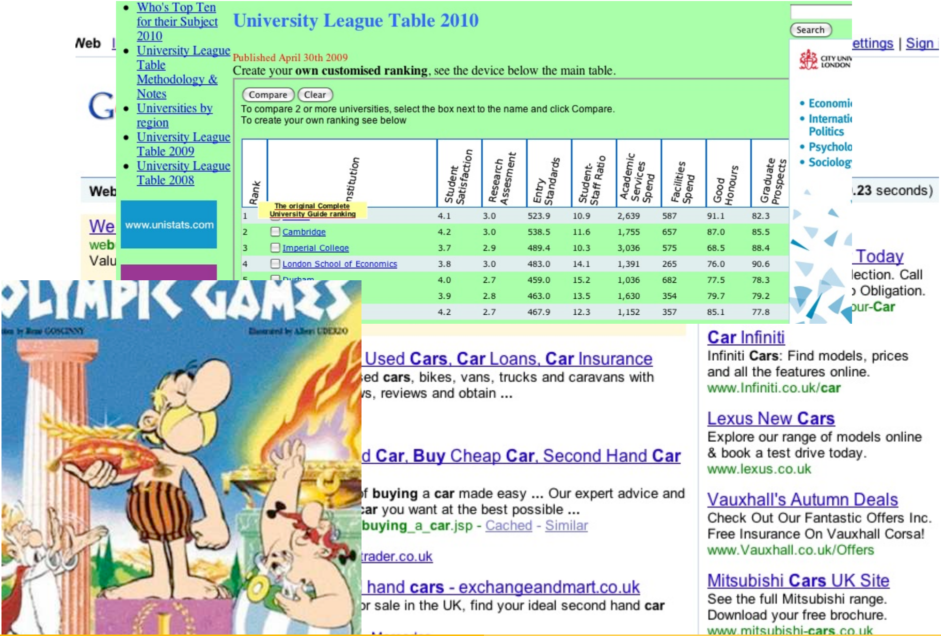
Car Infiniti
Infiniti **Cars**: Find models, prices and all the features online.
www.Infiniti.co.uk/**car**

Lexus New **Cars**
Explore our range of models online & book a test drive today.
www.lexus.co.uk

Vauxhall's Autumn Deals
Check Out Our Fantastic Offers Inc. Free Insurance On Vauxhall Corsa!
www.Vauxhall.co.uk/Offers

Mitsubishi **Cars** UK Site
See the full Mitsubishi range. Download your free brochure.
www.mitsubishi-**cars**.co.uk

Competition for rank



Competition for rank

Rank	Institution	Student Satisfaction	Research Assessment	Entry Standards	Student-Staff Ratio	Academic Services Spend	Facilities Spend	Good Honours	Graduate Prospects
1		4.1	3.0	523.9	10.9	2,639	587	91.1	82.3
2	Cambridge	4.2	3.0	538.5	11.6	1,755	657	87.0	85.5
3	Imperial College	3.7	2.9	489.4	10.3	3,036	575	68.5	88.4
4	London School of Economics	3.8	3.0	483.0	14.1	1,391	265	76.0	90.6
5	Durham	4.0	2.7	459.0	15.2	1,036	682	77.5	78.3
		3.9	2.8	463.0	13.5	1,630	354	79.7	79.2
		4.2	2.7	467.9	12.3	1,152	357	85.1	77.8

Competition for rank

Competition for rank

Telegraph.co.uk

Home | News | Sport | Finance | Lifestyle | Comment | Travel | Culture | Te

News by Sector | Comment | Personal Finance | Markets | Economics | Your Business

Money-Saving Tips | Investing | Savings | Gold | Insurance | Pensions | Funds

HOME > FINANCE > PERSONAL FINANCE

Britain's quality of life worse than former Communist countries

Britain's has fallen to 25th position on a list of best places in the world to live.

By Myra Butterworth, Personal Finance Correspondent
Published: 6:54PM GMT 06 Jan 2010

Share

Digg sub

20 re

Email

Our poor climate, soaring unemployment and congested roads means we are now ranked behind

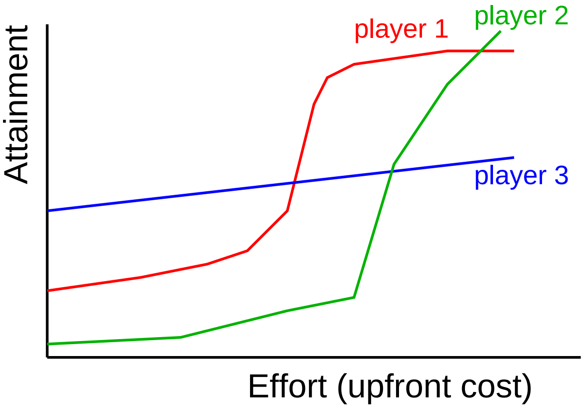
Some background on ranking games

"Ranking games" (Brandt, Fisher, Harrenstein and Shoham)

each combination of strategies results in a ranking of the players; every player has a monotonically decreasing function from rank to utility.

Problem: unrestricted ranking games are still hard: a 3-player ranking game can easily encode an unrestricted 2-player 0/1 game. (as noted earlier, hard to solve)
Our idea: assume strategies are correlated with "competitiveness"

The model



Each player has his own function from effort to performance.

The model

Player i has actions (pure strategies) $a^i_1, \ldots, a^i_n$

The model

Player i has actions (pure strategies) $a^i_1, \ldots, a^i_n$
a^i_j has associated quantities c^i_j (the cost) and r^i_j (the "return", or level of performance).

Players get ranked on the r^i_j -values they obtain.

The model

Player i has actions (pure strategies) $a^i_1, \ldots, a^i_n$
a^i_j has associated quantities c^i_j (the cost) and r^i_j (the "return", or level of performance).

Players get ranked on the r^i_j -values they obtain.

$c^i_i \leq c^i_{i+1}$ and $r^i_i \leq r^i_{i+1}$, i.e. lower-indexed strategies are less competitive.

The model

Player i has actions (pure strategies) $a_1^i, \ldots, a_n^i$
a_j^i has associated quantities c_j^i (the cost) and r_j^i (the "return", or level of performance).

Players get ranked on the r_j^i -values they obtain.

$c_i^i \leq c_{i+1}^i$ and $r_i^i \leq r_{i+1}^i$, i.e. lower-indexed strategies are less competitive.
There are "prizes" awarded to players according to rank; the k -th prize has value u_k .

If a player plays a_j and wins the k -th prize, his overall utility is $u_k - c_j$.

The model

Player i has actions (pure strategies) $a^i_1, \ldots, a^i_n$
a^i_j has associated quantities c^i_j (the cost) and r^i_j (the "return", or level of performance).

Players get ranked on the r^i_j -values they obtain.

$c^i_i \leq c^i_{i+1}$ and $r^i_i \leq r^i_{i+1}$, i.e. lower-indexed strategies are less competitive.
There are "prizes" awarded to players according to rank; the k -th prize has value u_k .

If a player plays a_j and wins the k -th prize, his overall utility is $u_k - c_j$.

Observation

We can concisely represent games with many players/strategies, in contrast with unrestricted ranking games.

Some results

We can pre-process a d -player game so as to assume that $u_1 = 1$, $u_d = 0$; all costs c_j^i lie in range $[0, 1]$; costs and returns are *strictly* monotonic in j , else we would have dominated actions; each player's weakest action has cost 0.

Theorem

Suppose there is just one prize ($u_1 > 1$; $u_j = 0$ for $j > 1$). Suppose ties are impossible (if all r_j^i -values are distinct, or equivalently there is a tie-breaking rule). Then there is just one player who gets positive payoff (all others get zero); namely the player who has the strongest action.

Some results

Proof.

- If a_n^1 is the strongest action in the game, note that player 1 can ensure a payoff of $u_1 - c_n^1$.

Some results

Proof.

- If a_n^1 is the strongest action in the game, note that player 1 can ensure a payoff of $u_1 - c_n^1$.
- In a NE, for each player i let a_W^i be the weakest action that i plays with positive probability. All but one of these actions are guaranteed to lose (payoff: $-c_W^i$)

Some results

Proof.

- If a_n^1 is the strongest action in the game, note that player 1 can ensure a payoff of $u_1 - c_n^1$.
- In a NE, for each player i let a_W^i be the weakest action that i plays with positive probability. All but one of these actions are guaranteed to lose (payoff: $-c_W^i$)
- So, all but one player get a non-positive payoff (since a player's payoff is his expected payoff for any action he uses with positive probability. i can get payoff 0 by playing a_1^i, so presumably his overall payoff is 0.

Some results

Proof.

- If a_n^1 is the strongest action in the game, note that player 1 can ensure a payoff of $u_1 - c_n^1$.
- In a NE, for each player i let a_W^i be the weakest action that i plays with positive probability. All but one of these actions are guaranteed to lose (payoff: $-c_W^i$)
- So, all but one player get a non-positive payoff (since a player's payoff is his expected payoff for any action he uses with positive probability. i can get payoff 0 by playing a_1^i, so presumably his overall payoff is 0.
- Finally, we found precisely one player who can get positive payoff.

□

What if the strongest action has cost 1? What about > 1 prizes?

Some results

Theorem

Suppose there is just one prize ($u_1 > 1$; $u_j = 0$ for $j > 1$). Suppose ties are impossible (if all r^i_j-values are distinct, or equivalently there is a tie-breaking rule).
Then if you know the support of the solution, you con compute it easily; also, the solution is all in rational numbers.

Some results

Theorem

Suppose there is just one prize ($u_1 > 1$; $u_j = 0$ for $j > 1$). Suppose ties are impossible (if all r^i_j-values are distinct, or equivalently there is a tie-breaking rule).
Then if you know the support of the solution, you con compute it easily; also, the solution is all in rational numbers.

(So, that's like 2-player normal-form games! Is that interesting?)

Some results

- How about poly-time algorithms? We have some for special cases...

Some results

- How about poly-time algorithms? We have some for special cases...
- 2-player games are easy; no, they are not zero-sum; it's quite a cute algorithm

Some results

- How about poly-time algorithms? We have some for special cases...
- 2-player games are easy; no, they are not zero-sum; it's quite a cute algorithm
- d-players, n actions, where d is constant: Approximate NE can be found in poly-in-n time by brute-force approach.

Some results

- How about poly-time algorithms? We have some for special cases...
- 2-player games are easy; no, they are not zero-sum; it's quite a cute algorithm
- d-players, n actions, where d is constant: Approximate NE can be found in poly-in-n time by brute-force approach.
- FPTAS for d players, 1 prize (in the paper, done for just 2 players) Dynamic programming approach

Linear-prize ranking games

Suppose the k -th prize has value $a - bk$ where a and b are positive constants. We can solve as follows.

Linear-prize ranking games

Suppose the k -th prize has value $a - bk$ where a and b are positive constants. We can solve as follows.

Each player gains b for every other player he beats. So, express his payoff as the negation of the cost of his action, plus the sum of payoffs from a bunch of zero-sum 2-player games.

Linear-prize ranking games

Suppose the k -th prize has value $a - bk$ where a and b are positive constants. We can solve as follows.

Each player gains b for every other player he beats. So, express his payoff as the negation of the cost of his action, plus the sum of payoffs from a bunch of zero-sum 2-player games.

His payment of that cost can be considered as a 2-player game against "nature" (a dummy player) who collects the cost but does not influence the player.

Linear-prize ranking games

Suppose the k -th prize has value $a - bk$ where a and b are positive constants. We can solve as follows.

Each player gains b for every other player he beats. So, express his payoff as the negation of the cost of his action, plus the sum of payoffs from a bunch of zero-sum 2-player games.

His payment of that cost can be considered as a 2-player game against "nature" (a dummy player) who collects the cost but does not influence the player.

So, we have reduced the game to a zero-sum polymatrix game, which is known to be solvable in poly-time (Daskalakis and Papadimitriou '09).

Conclusions and further work

- unrestricted games (designed by a notional adversary to be difficult to solve) indeed “cannot” be solved by efficient algorithms.

Conclusions and further work

- unrestricted games (designed by a notional adversary to be difficult to solve) indeed "cannot" be solved by efficient algorithms.
- focus on "natural" types of more tractable games

Conclusions and further work

- unrestricted games (designed by a notional adversary to be difficult to solve) indeed "cannot" be solved by efficient algorithms.
- focus on "natural" types of more tractable games
- For these games, continue by looking for *decentralised* algorithms (a solution is implausible if it needs to be found centrally and then handed out to the players).

Conclusions and further work

- unrestricted games (designed by a notional adversary to be difficult to solve) indeed "cannot" be solved by efficient algorithms.
- focus on "natural" types of more tractable games
- For these games, continue by looking for *decentralised* algorithms (a solution is implausible if it needs to be found centrally and then handed out to the players).
- Another direction: weaken the objective – "approximate equilibria" replace "no incentive for a player to change" with "only a small incentive to change" — an interesting and challenging problem, both for centralised and decentralised algorithms!