

Optimizing your hero pick in Dota 2

Erick Martins Ratamero

Department of Physics/School of Life Sciences
University of Warwick
www.warwick.ac.uk/erickratamero

Let's talk about Dota.

- It's a MOBA
- 5x5
- Teamwork is essential
- Paid over \$18 million in prizes at the last world championships



 Abaddon	 Alchemist	 Ancient Apparition	 Anti-Mage	 Arc Warden	 Axe	 Bane	 Batrider
 Beastmaster	 Bloodseeker	 Bounty Hunter	 Brewmaster	 Bristleback	 Broodmother	 Centaur Warrunner	 Chaos Knight
 Chen	 Clinkz	 Clockwerk	 Crystal Maiden	 Dark Seer	 Dazzle	 Death Prophet	 Disruptor
 Doom	 Dragon Knight	 Drow Ranger	 Earth Spirit	 Earthshaker	 Elder Titan	 Ember Spirit	 Enchantress
 Enigma	 Faceless Void	 Gyrocopter	 Huskar	 Invoker	 Io	 Jakiro	 Juggernaut
 Keeper of the Light	 Kunkka	 Legion Commander	 Leshrac	 Lich	 Lifestealer	 Lina	 Lion
 Lone Druid	 Luna	 Lycan	 Magnus	 Medusa	 Meepo	 Mirana	 Morphling
 Naga Siren	 Nature's Prophet	 Necrophos	 Night Stalker	 Nyx Assassin	 Ogre Magi	 Omniknight	 Oracle
 Outworld Devourer	 Phantom Assassin	 Phantom Lancer	 Phoenix	 Puck	 Pudge	 Pugna	 Queen of Pain
 Razor	 Riki	 Rubick	 Sand King	 Shadow Demon	 Shadow Fiend	 Shadow Shaman	 Silencer
 Skywrath Mage	 Slardar	 Slark	 Sniper	 Spectre	 Spirit Breaker	 Storm Spirit	 Sven
 Techies	 Templar Assassin	 Terrorblade	 Tidehunter	 Timbersaw	 Tinker	 Tiny	 Treant Protector
 Troll Warlord	 Tusk	 Undying	 Ursa	 Vengeful Spirit	 Venomancer	 Viper	 Visage
 Warlock	 Weaver	 Windranger	 Winter Wyvern	 Witch Doctor	 Wraith King	 Zeus	

Quick back of the envelope

number of possible teams: $\sim 10^{10}$

number of possible matches: $\sim 10^{20}$

Let's just look at individual heroes, shall we?

Gathering Data



Techies

Matchups

88th

POPULARITY

47.68%

WIN RATE



[Overview](#) [Guides](#) [Items](#) **[Matchups](#)** [Ability Builds](#) [Abilities](#) [Trends](#) [Player Rankings](#)

MOST SUCCESSFUL MATCHUPS, TECHIES, THIS MONTH

This Month

Hero	Advantage ?	Techies Win Rate	Matches Played
Undying	3.13%	47.47%	48,405
Weaver	3.05%	48.90%	70,114
Ember Spirit	2.49%	55.73%	106,548
Riki	2.11%	48.50%	113,214
Shadow Fiend	2.07%	51.73%	129,432
Elder Titan	1.96%	46.41%	16,556
Pugna	1.93%	53.19%	34,328
Dark Seer	1.75%	49.74%	30,655
Bounty Hunter	1.61%	51.27%	117,459
Bloodseeker	1.59%	51.86%	89,836
Alchemist	1.45%	49.71%	110,506

- Simple Python scraper
- (requests + BeautifulSoup)
- Data gathered for latest patch (6.86d, about 2 weeks old)
- Matrix of payoffs generated from "advantage" stat

Everyone knows Game Theory, right? Good.

Into SageMath we go!

```
g=build_game("advantages.csv")
```

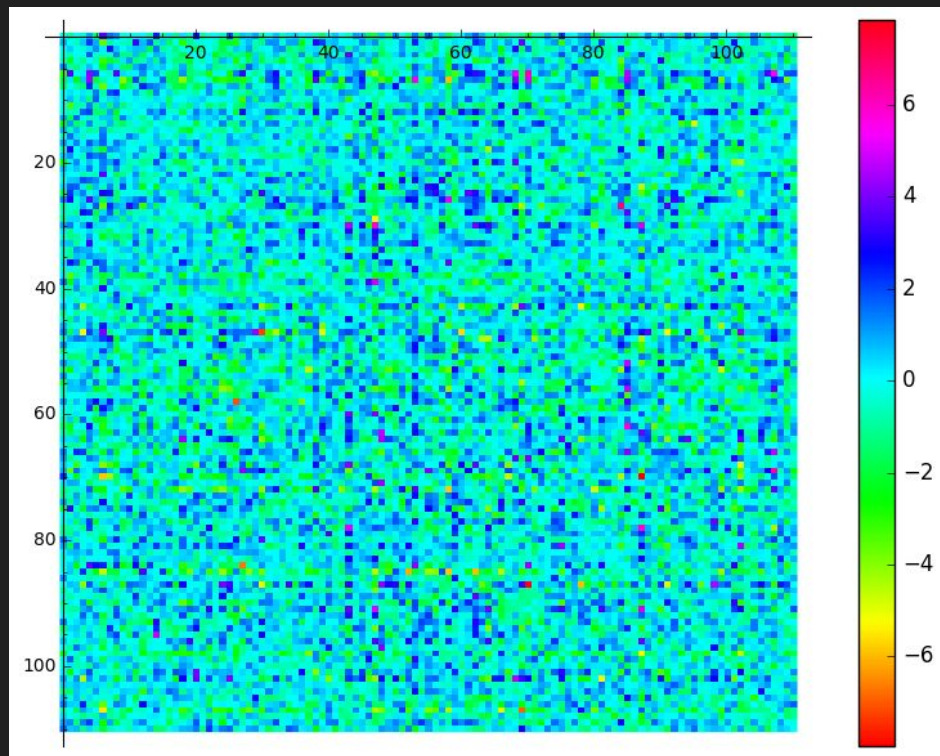
```
m = g.payoff_matrices()[0].n(digits=2)
```

```
sum_of_threats = [sum([row[k] for row in m]) for  
k in range(111)]
```

```
print [hero_names[i] for i, score in  
enumerate(sum_of_threats) if score ==  
min(sum_of_threats)]
```

(it's lo!)

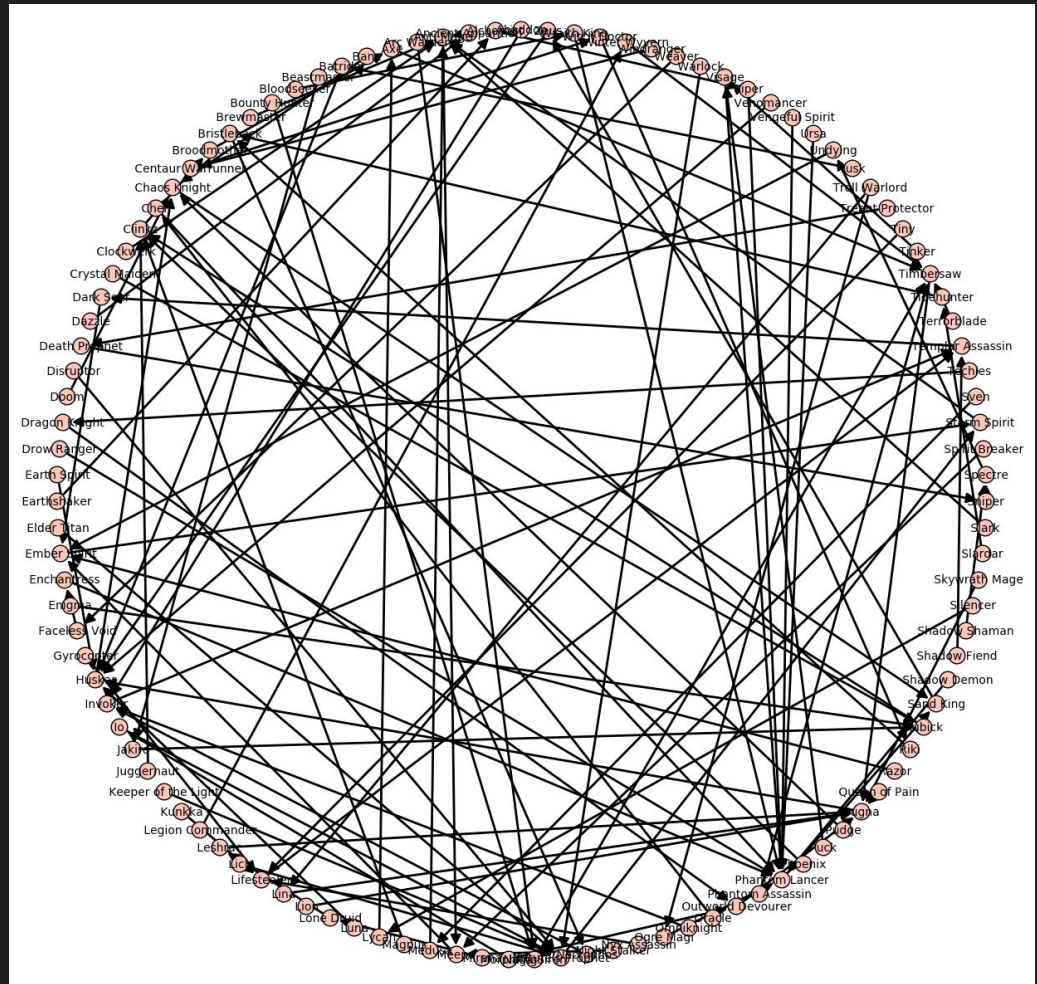
```
plot(g.payoff_matrices()[0],  
colorbar=True,cmap='hsv')
```



```
d = get_best_response_dictionary(g)
```

```
G = digraph_representation(g)
```

```
#G.show(layout='circular',figsize=[12,12])
```



```
NEs = g.obtain_nash(algorithm='lrs')
```

NOPE

111x111 matrix is WAY too much for my laptop.

We could try a subset of heroes for fun,
though...

Let's take the first 20 heroes!

```
g=build_game("advantages_20.csv")
```

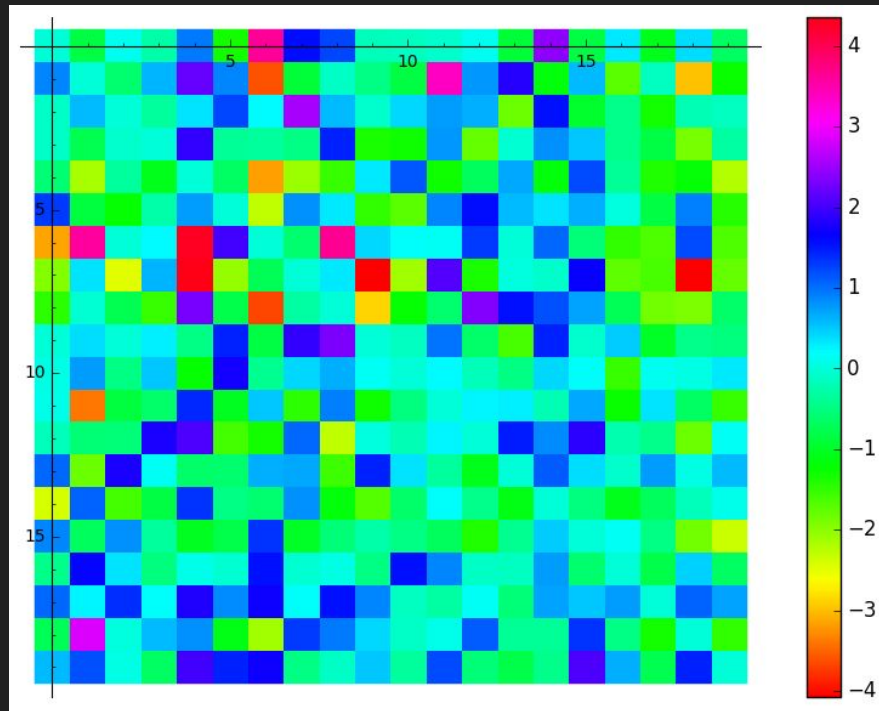
```
m = g.payoff_matrices()[0].n(digits=2)
```

```
sum_of_threats = [sum([row[k] for row in m]) for  
k in range(20)]
```

```
print [hero_names[i] for i, score in  
enumerate(sum_of_threats) if score ==  
min(sum_of_threats)]
```

(it's Tiny!)

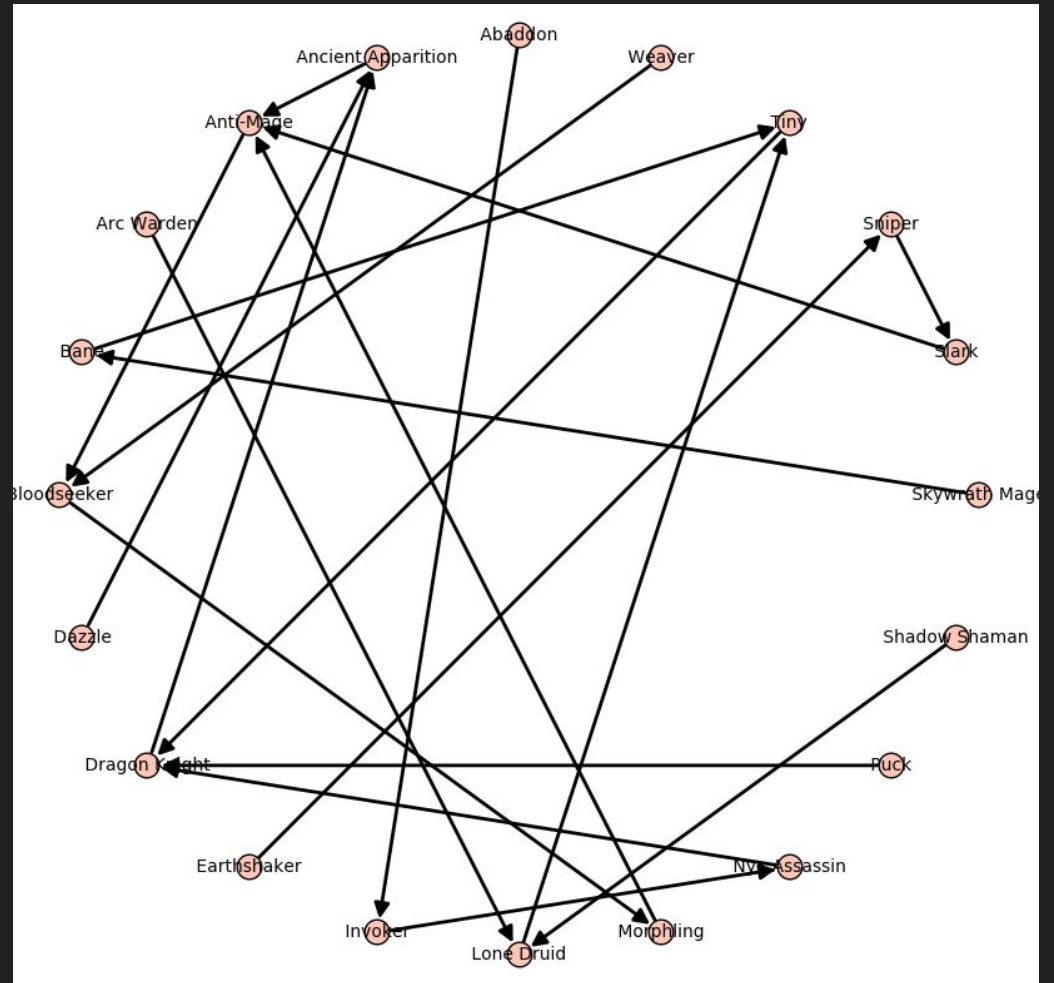
```
plot(g.payoff_matrices()[0],  
colorbar=True,cmap='hsv')
```




```
d = get_best_response_dictionary(g)
```

```
G = digraph_representation(g)
```

```
#G.show(layout='circular',figsize=[8,8])
```



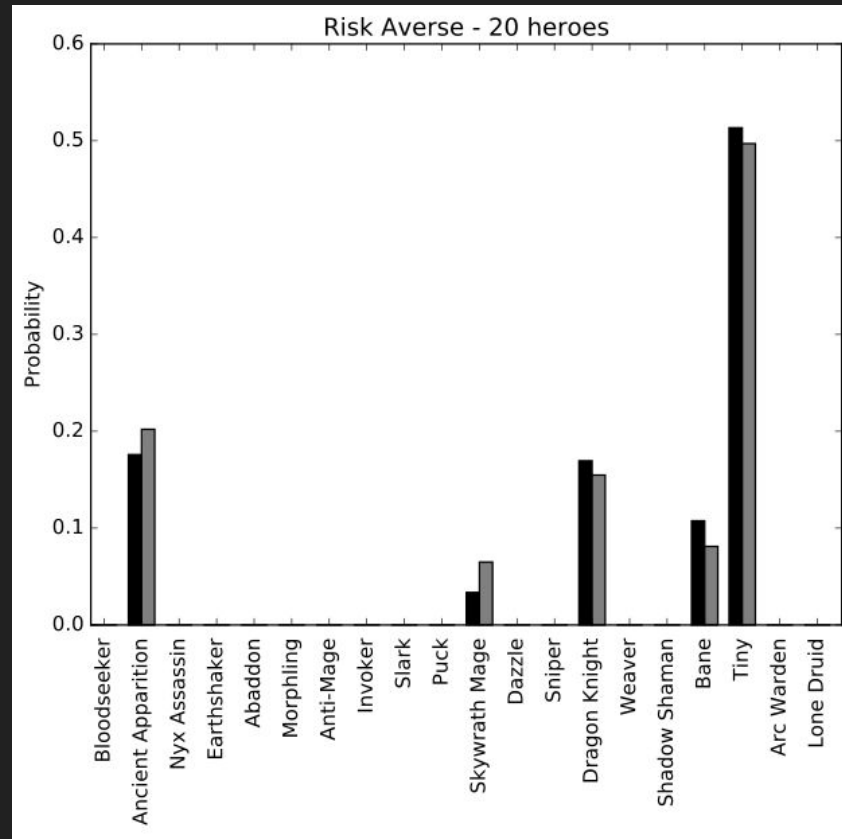
```
NEs = g.obtain_nash(algorithm='lrs')
```

1 equilibrium!

Player 1 plays: {0., 0.175942, 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0.0335806, 0., 0., 0.169511, 0., 0., 0.107597,
0.51337, 0., 0.}

Player 2 plays: {0., 0.202066, 0., 0., 0., 0., 0., 0., 0.,
0., 0.0649503, 0., 0., 0.154792, 0., 0., 0.081129,
0.497062, 0., 0.}

```
mean_ne = plot_mean_NE()
```



```
played = heroes_that_are_played(mean_ne)

print played
```

```
not_played =
heroes_that_are_not_played(mean_ne)



print not_played
```

Played heroes: [' Ancient Apparition ', ' Skywrath Mage ', ' Dragon Knight ', ' Bane ', ' Tiny ']

Not played heroes: [' Bloodseeker ', ' Nyx Assassin ', ' Earthshaker ', ' Abaddon ', ' Morphling ', ' Anti-Mage ', ' Invoker ', ' Slark ', ' Puck ', ' Dazzle ', ' Sniper ', ' Weaver ', ' Shadow Shaman ', ' Arc Warden ', ' Lone Druid ']

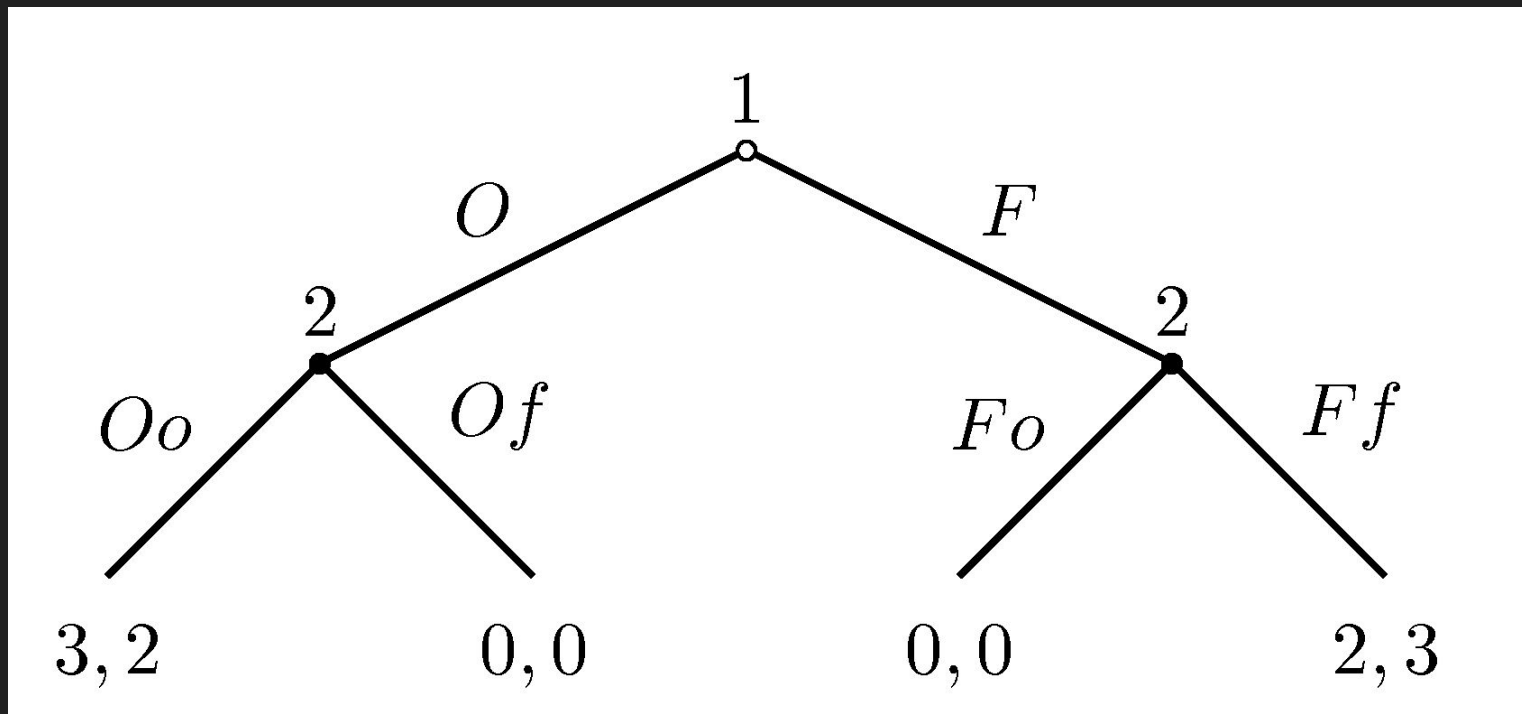
An Alternative approach: Captain's Mode

Captains Mode picking order

(This example assumes  *Radiant* is the starting team. Invert teams if  *Dire* is the starting team.)



...which is pretty much an extended-form game



- Payoffs are necessary for each possible combination of teams
- (remember there are 10^{20} of those)
- to make things worse, team combinations depend on banned heroes
- $\sim 10^{40}$ possible leaves!
- Even a game with only 20 heroes would have 20! ($\sim 10^{18}$) possible leaves

Acknowledgements

Dr. Vincent Knight (Cardiff University) for pretty much the whole SageMath code and the original idea

Matthew Lewis for his Dotabuff scraping code

Dotabuff for being a great resource



Funded by
the European Union