Digraph k-Coloring Games

From Theory to Practice

UNIVERSITÀ DEGLI STUDI DELL'AQUILA

A. D'Ascenzo[1], M. D'Emidio[1], M. Flammini[2], G. Monaco[1]
[1] University of L'Aquila, Italy
[2] Gran Sasso Science Institute, Italy

SEA 25 - 27 July 2022

Problem Introduction

- A directed unweighted graph $G = (V, E)$, $|V| = n$, $|E| = m$, and a set of $k \geq 2$ colors:
  - Vertices represent **autonomous agents**;
  - Arcs represent mutual **undirectional conflicts**;
  - Colors denote **agents' available strategies**;
- Each agent aims at **maximizing her own payoff**, defined as the number of outgoing neighbors with a color different from hers.

Motivations

- **Wireless networks**: radio stations wish to select the transmission frequency not used by the maximum number of neighboring stations within their range;
- **Social networks**: members must be split in groups and want to maximize the number of enemies they do not end together with;
- **Markets**: sellers aim to locate their activities as far as possible from their direct competitors.

Notions of Equilibrium

- A **solution** to an instance of the digraph k-coloring game corresponds to a **state** $C = (c_1, \dots, c_n)$, where c_i is the color chosen by vertex i;
- A solution is said to be a (pure) **Nash Equilibrium** (NE) if no agent can improve her payoff by changing strategy (i.e. color);
- Unfortunately, it is known [KPR13 (SAGT)] that **the problem of understanding whether digraph k-coloring games admit a NE is NP-complete**, for any $k \geq 2$;
- Carosi et al. [CFM17 (AAMAS)] focused on γ-Nash Equilibrium (γ-NE), which is a state where no agent can strictly improve her payoff by a **multiplicative factor of** γ by changing color, for some $\gamma \geq 1$, and developed algorithms for this equilibrium notion.

Related Problems

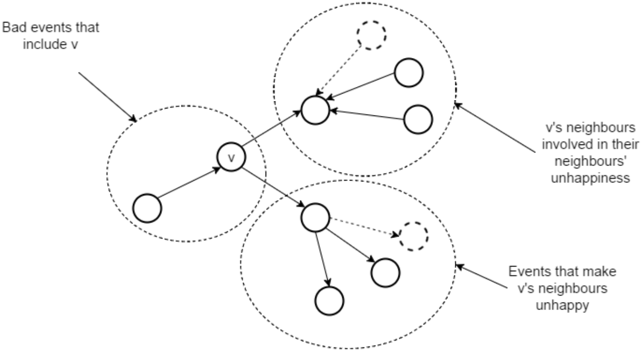
- *k*-coloring game in undirected graphs **always admits a NE**, and can be **computed in polynomial time** if the graph is **unweighted** [Ho07, KPR13(SAGT)];
- If the graph is **weighted**, a NE always exists but it is **PLS-complete** to compute it, also for $k = 2$ [SY91(JComput)];
- These results exploit the *potential function method*:
  - digraph *k*-coloring games **do not admit a potential function**;
- Related class of problems are *graphical games* [KLS01(UAI), BFFM11] and *hedonic games* [AS16(HCSC)].

Algorithms for Digraph k-Coloring Game: AP1

- AP1 is a deterministic, polynomial time algorithm, running in $O(\Delta_o nm)$;
- Given a digraph G and $k \geq 3$, returns a k-coloring such that every vertex with positive degree has **payoff at least 1**;
- Corresponds to a Δ_o-NE, since Δ_o is the maximum payoff an agent can achieve;
- Iterative algorithm: at each iteration visits the uncolored vertices and **detects a cycle or a path**;
- It colors the vertices by **alternating either three or two colors**;

Algorithms for Digraph k-Coloring Game: LLL-SPE

- LLL-SPE is based on the *Lovász Local Lemma* (LLL);
- A random assignment of k colors **has positive probability** of returning a constant approximate NE;
- Works for any $k \geq 2$ and any digraph G such that the minimum outgoing degree of any vertex $v \in V(G)$ is $\delta_o^v = \Omega(\log \Delta_o + log \Delta_i)$;
- Constructive version of LLL runs in **polynomial expected running time**;
- Starts from a random assignment and iteratively **resamples** the colors of γ-unhappy vertices and of vertices in their **dependency set**:
  - a γ-unhappy vertex v is an agent that can improve her payoff by a multiplicative factor of γ by changing color;
  - a resample operation consists of changing color to the vertices in the dependency set of a γ-unhappy vertex;
  - Dependency set of v is made of vertices that whose status is influenced / influences that of v;



Graphical representation of all the types of event that can influence a vertex's behaviour

Experimental Analysis

- AP1 and LLL-SPE have different nature: the former is a **deterministic** algorithm with an approximation guarantee dependent on the **graph size**; the latter is a **probabilistic** algorithm having a **constant** approximation guarantee, working on a restricted class of graphs;
- Up to now, it was not clear which method should be adopted in **practice**
- Moreover, by a first analysis, both algorithms resulted to produce the same results as a naive random assignment
- We performed an **experimental study** seeking for the most appropriate algorithm for digraph k-coloring games;
- In order to conduct a complete experimental analysis, we have:
  - extended LLL-SPE to LLL-GEN, since LLL-SPE applicability is restricted to a class of graphs which rarely appears in practice;
  - considered a naive fully-random algorithm RANDOM that assigns colors to agents uniformly at random;
  - casted the well-known **best-response dynamics** to this class of problems, devising the BEST-RESP algorithm.

- LLL-SPE guarantees' rely on the constraint that $\delta_o^v = \Omega(\log \Delta_o + log \Delta_i)$, which rarely happens in real world networks;
- Convergence results are not guaranteed on general graphs;
- LLL-GEN is a **generalization** of LLL-SPE that takes in input also an integer I which specifies a threshold to the number of iterations the algorithm has to run for, and casts the γ approximation value to general graphs;
- LLL-GEN running time is $O(I(n + \Delta_o + \Delta_i + \Delta_o \Delta_i))$;
- RANDOM is the procedure of **uniformly assigning** at random a color to each agent in the graph;
- RANDOM running time is $\Theta(n)$.

Algorithms for Digraph k-Coloring Game: BEST-RESP

- BEST-RESP is based on the classical concept of **best-response dynamics**;
- Starts with a **random coloring**;
- Selects a 1-**unhappy vertex** v, if any;
- Assigns the color c_v to v that **maximizes** v's **payoff**;
- The process stop when all the vertices are happy, or when a maximum number of iterations I is reached;
- Runs in $O(n\Delta_o I)$.

Experimental Setting

- The analysis has been carried on an **ample and heterogeneous** set of graph instances;
- Various values of k have been chosen in order to **magnify the dependency** on k in a reasonable number of tests;
- Considered metrics of interest are (given a coloring C):
  - **Approximation ratio** $\gamma(G, C)$: maximum γ-value over all the agents in the graph;
  - **Average payoff** $\bar{P}(G, C)$: arithmetic mean of the vertices' payoff;
  - **Fraction of unhappy vertices** $U(G, C)$: number of unhappy vertices divided by the order of G;
  - **Running time** $T(G, C)$: running time spent on G to compute the coloring c;
- The algorithms have been implemented in Python 3.8, exploiting *NetworKit* as graph library.

Graph Dataset

Dataset	Short	Type	$\|V\|$	$\|A\|$	$\overline{d_o}$	$\overline{\overline{d_o}}$	Δ_o	S	LLL
Twitter	TWI	DIGITAL SOCIAL	23370	33101	1.42	0	238	○	○
FACEBOOK	FAC	DIGITAL SOCIAL	309717	472792	1.53	0	358	○	○
AMAZON	AMA	RATINGS	80679	135336	1.68	2	9	○	○
Flight	FLT	INFRASTRUCTURE	1226	2613	2.13	1	24	○	○
Peer2Peer	P2P	INTERNET	62586	147892	2.36	0	78	○	○
Luxembourg	LUX	ROAD	30647	75546	2.47	3	9	○	○
RAND3	RR3	RANDOM	10000	30000	3	3	3	●	●
RAND4	RR4	RANDOM	10000	40000	4	4	4	●	●
Oregon-AS	ORE	AUTONOMOUS SYSTEM	10670	44004	4.12	2	2312	○	○
RAND5	RR5	RANDOM	10000	50000	5	5	5	●	●
Health	HEA	HUMAN SOCIAL	2539	12969	5.11	5	10	○	○
RELATIVITY	REL	COLLABORATION	5242	28968	5.53	3	81	○	○
Linux	LIN	COMMUNITY	30834	213424	6.92	5	243	○	○
Peer2PeerSm	SPP	INTERNET	10876	79988	7.35	5	103	○	○
Google	GOO	HYPERLINKS (LOCAL)	15763	170335	10.81	8	852	○	○
Erdős-Rényi A	ERA	RANDOM	1000	12460	12.46	12	27	●	○
Blog	BLG	INTERACTION	1224	19022	15.54	7	256	○	○
Erdős-Rényi B	ERB	RANDOM	1000	24943	24.94	25	45	●	●
wiki-Vote	WVT	VOTING	7115	201524	28.32	4	1065	○	○
Email	EMA	INTERACTION	1005	32128	31.97	21	345	○	○
Erdős-Rényi C	ERC	RANDOM	1000	49924	49.92	50	74	●	●
Erdős-Rényi D	ERD	RANDOM	1000	100025	100.03	100	134	●	●
Erdős-Rényi E	ERE	RANDOM	1000	199443	199.44	199	238	●	●
PALEY601	PL1	RANDOM	601	180300	300	300	300	●	●
PALEY1181	PL2	RANDOM	1181	696790	590	590	590	●	●

Overview of used input digraphs. The first three columns contain dataset name, acronym, and type; the 4th and 5th columns show number of vertices and arcs of the digraph; the 6th, 7th and 8th columns report average, median and maximum outgoing degree. Finally, the 9th column highlights whether the graph is synthetic or real-world (● = true, ○ = false), while the last column specifies whether the LLL holds in the given graph (● = true, ○ = false). Inputs are sorted by $\overline{d_o}$.

Summary Results

metric	algorithm	best	2nd	3rd	worst	total
$\gamma(G,c)$	RND	4 (2.3 %)	33 (18.9 %)	81 (46.3 %)	57 (32.5 %)	175 (100 %)
	AP1	1 (0.6 %)	69 (39.4 %)	51 (29.1 %)	54 (30.9 %)	175 (100 %)
	LLG	7 (4.0 %)	62 (35.4 %)	43 (24.6 %)	63 (36.0 %)	175 (100 %)
	BR	163 (93.1 %)	11 (6.3 %)	0 (0.0 %)	1 (0.6 %)	175 (100 %)
$U(G,c)$	RND	1 (0.6 %)	42 (24.0 %)	106 (60.6 %)	26 (14.8 %)	175 (100 %)
	AP1	0 (0.0 %)	13 (7.4 %)	13 (7.4 %)	149 (85.1 %)	175 (100 %)
	LLG	6 (3.4 %)	114 (65.1 %)	55 (31.5 %)	0 (0.0 %)	175 (100 %)
	BR	168 (96.0 %)	6 (3.4 %)	1 (0.6 %)	0 (0.0 %)	175 (100 %)
$\overline{P}(G,c)$	RND	5 (2.9 %)	56 (32.0 %)	96 (54.9 %)	18 (10.2 %)	175 (100 %)
	AP1	0 (0.0 %)	7 (4.0 %)	13 (7.4 %)	155 (88.6 %)	175 (100 %)
	LLG	7 (4.0 %)	106 (60.6 %)	60 (34.3 %)	2 (1.1 %)	175 (100 %)
	BR	163 (93.2 %)	6 (3.4 %)	6 (3.4 %)	0 (0.0 %)	175 (100 %)
T (G,c)	RND	173 (98.9 %)	2 (1.1 %)	0 (0.0 %)	0 (0.0 %)	175 (100 %)
	AP1	2 (1.1 %)	152 (86.9 %)	14 (8.0 %)	7 (4.0 %)	175 (100 %)
	LLG	0 (0.0 %)	20 (11.4 %)	101 (57.7 %)	54 (30.9 %)	175 (100 %)
	BR	0 (0.0 %)	1 (0.6 %)	60 (34.3 %)	114 (65.1 %)	175 (100 %)

- Aggregate statistics for all tested algorithms with respect to the four metrics, for all combinations of inputs and values of k.
- Data highlights that BR is the **best performing one**, globally, and that it has been able to find **pure NE** in almost all instances!

Results for $k = 3$

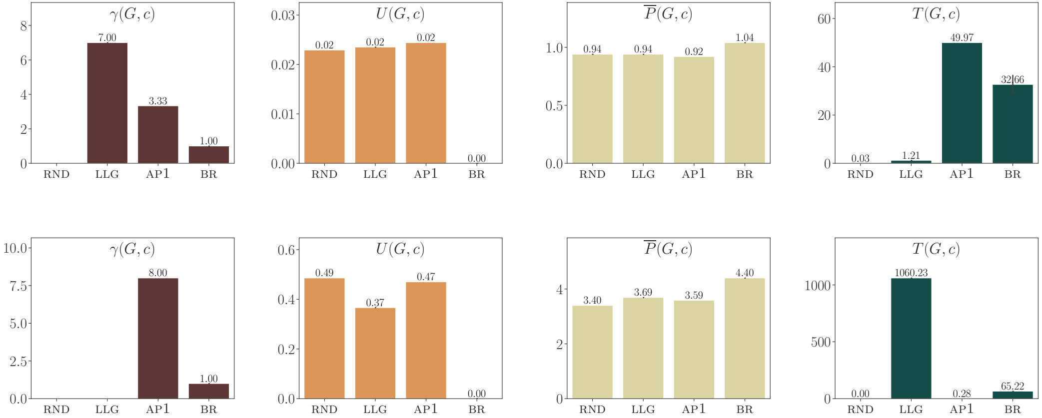


Figure: Performance of algorithms RND, LLG, AP1 and BR, resp., in graphs TWI (top) and HEA (bottom), with $k = 3$.

Results for $k = 3$ (cont'd)

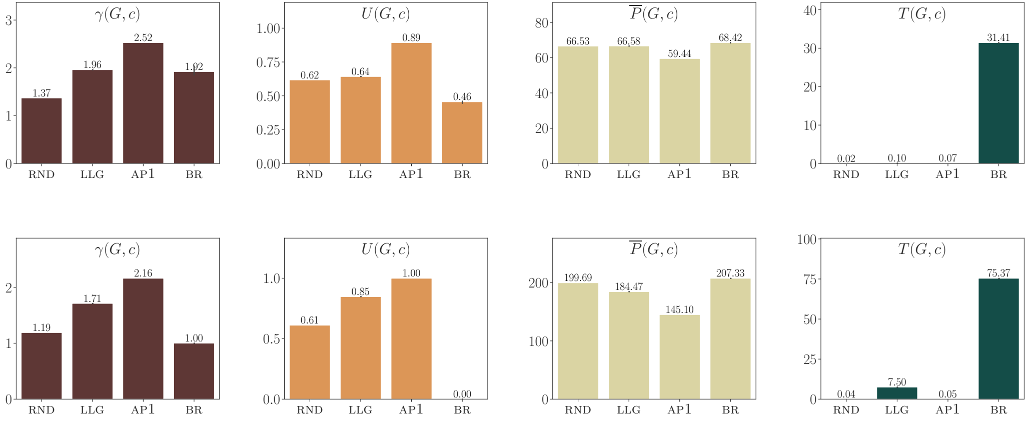


Figure: Performance of algorithms RND, LLG, AP1 and BR, resp., in graphs ERD (top) and PL1 (bottom), with $k = 3$.

Results for $k > 3$

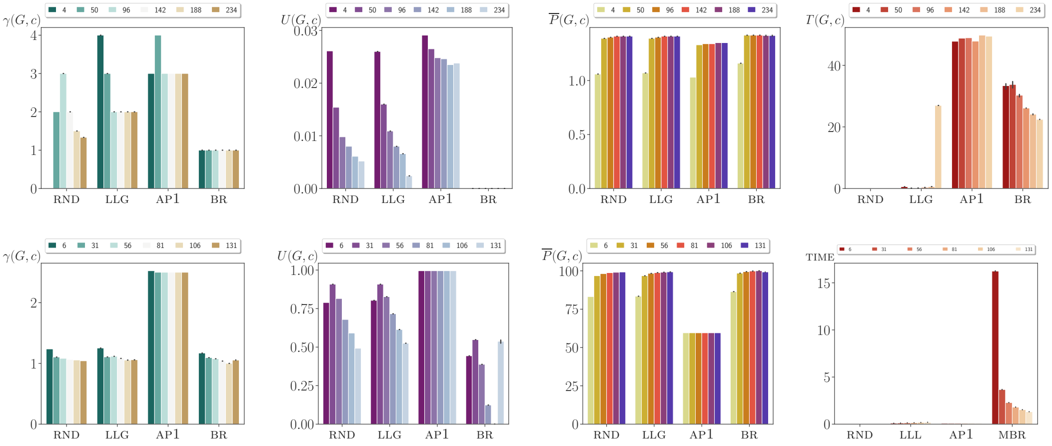


Figure: Performance of algorithms RND, LLG, AP1 and BR, resp., in graphs TWI (top) and ERD (bottom) , with increasing values of k.

Future Research

- Our analysis provides empirical evidence of the following facts:
  - AP1 and LLL-GEN performs badly in practice;
  - **best response dynamics outperforms** algorithms with guarantees, providing **pure NE** in almost all tested graph instances;
- Motivates research efforts towards proving the **existence of NE** in specific graph classes;
- Even when a pure NE is not reached, γ **values result to be close to 1**, suggesting that algorithms with **better theoretical guarantees** may be devised.

Thanks for you attention!

Any question?

References

KPR13 J. Kun, B. Powers, and L. Reyzin. *Anti-coordination games and stable graph colorings*, SAGT13

CFM17 R. Carosi, M. Flammini, and G. Monaco. *Computing approximate pure nash equilibria in digraph k-coloring games*,AAMAS17

Ho07 Martin Hoefer. *Cost sharing and clustering under distributed competition.* PhD thesis, University of Konstanz, 2007;

SY91 A. A. Schäffer and M. Yannakakis. *Simple local search problems that are hard to solve.* SIAM J. Comput., 20(1):56-87, 1991

KLS01 M. J. Kearns, M. L. Littman, and S. P. Singh. *Graphical models for game theory*, UAI01

BFFM11 V. Bilò, A. Fanelli, M. Flammini, and L. Moscardelli. *Graphical congestion games.* Algorithmica, 61(2):274-297, 2011

AS16 H. Aziz and R. Savani. *Hedonic games.* Handbook of Computational Social Choice, pages 356-376, 2016