Playing Dominoes Is Hard, Except by Yourself

Erik D. Demaine, Fermi Ma, Erik Waingarten

Presentation by Chio Ge, Tassilo Schwarz

Why should I care?

- Winning strategy: complexity
- Useful for reductions
- Reduction from instances of job scheduling problem [1]

[1] two-machine no-idle/no-wait shop scheduling [BDCST19]

Domino Game: Classical Version

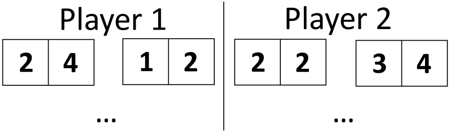
- Domino set: $\left\{(a, b) \middle| a \leq b \wedge a, b \in \{1, \ldots, 7\}\right\}$

Domino Game: Classical Version

- Domino set: $\left\{(a, b) \middle| a \leq b \wedge a, b \in \left\{1, \ldots, 7\right\}\right\}$
- Evenly distributed among players

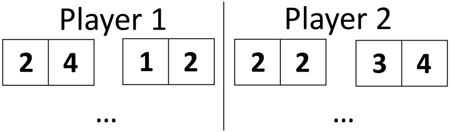
Domino Game: Classical Version

- Domino set: $\{(a, b) | a \leq b \wedge a, b \in \{1, \ldots, 7\}\}$
- Evenly distributed among players



Domino Game: Classical Version

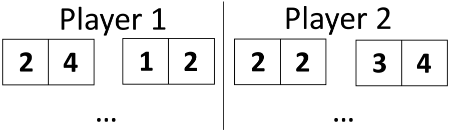
- Domino set: $\{(a,b) | a \leq b \wedge a,b \in \{1,\ldots,7\}\}$
- Evenly distributed among players



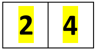
- Players take turn building a *valid* chain

Domino Game: Classical Version

- Domino set: $\{(a, b) | a \leq b \wedge a, b \in \{1, \ldots, 7\}\}$
- Evenly distributed among players



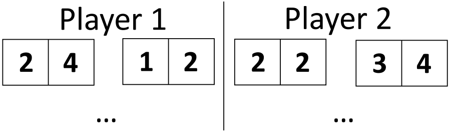
- Players take turn building a *valid* chain



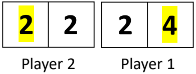
Player 1

Domino Game: Classical Version

- Domino set: $\{(a, b) | a \leq b \wedge a, b \in \{1, \dots, 7\}\}$
- Evenly distributed among players



- Players take turn building a *valid* chain



Domino Game: Classical Version

- Domino set: $\{(a, b) | a \leq b \wedge a, b \in \{1, \ldots, 7\}\}$
- Evenly distributed among players

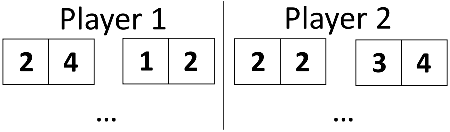
Player 1		Player 2	
2 | 4	1 | 2	2 | 2	3 | 4
…		…	

- Players take turn building a *valid* chain

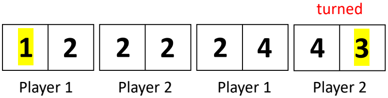
1 | 2	2 | 2	2 | 4
Player 1	Player 2	Player 1

Domino Game: Classical Version

- Domino set: $\{(a, b) \mid a \leq b \wedge a, b \in \{1, \ldots, 7\}\}$
- Evenly distributed among players

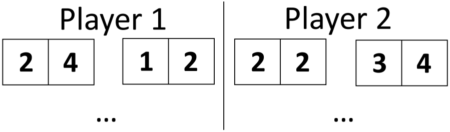


- Players take turn building a *valid* chain

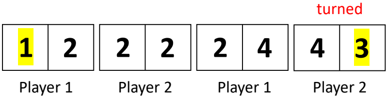


Domino Game: Classical Version

- Domino set: $\{(a,b) | a \leq b \wedge a, b \in \{1, \ldots, 7\}\}$
- Evenly distributed among players



- Players take turn building a *valid* chain



- Winner:
  - Player, first out of dominoes
  - Opponent stuck

Domino Game: Generalized Version

Instance of Domino Game:

- multiset of n dominoes
- arbitrary distribution among players

Winner:

- Rid of all dominoes
- Opponent stuck

Domino Game: Generalized Version

Instance of Domino Game:

- multiset of n dominoes
- arbitrary distribution among players

Winner:

- Rid of all dominoes
- Opponent stuck

Goal

Complexity to decide: Given a Dominoes instance, does a winning strategy exist?

Variants of Dominoes

Cooperative Dominoes

p-COOP-DOM = $\{\ \langle$Instance of Dominoes$\rangle \mid$All players can help P1 win the game$\}$

- $p \geq 1$ players
- Can all players help Player 1 win the game?

Variants of Dominoes

Cooperative Dominoes

p-$COOP$-$DOM = \{\, \langle \text{Instance of Dominoes} \rangle \,|\, \text{All players can help P1 win the game} \}$

- $p \geq 1$ players
- Can all players help Player 1 win the game?

Competitive Dominoes

p-$COMP$-$DOM = \{\, \langle \text{Instance of Dominoes} \rangle \,|\, \text{P1 has a winning strategy} \}$

- $p \geq 2$ players
- Does player 1 have a winning strategy?

Variants of Dominoes

Cooperative Dominoes

p-COOP-DOM = $\left\{ \langle \text{Instance of Dominoes} \rangle \middle| \text{All players can help P1 win the game} \right\}$

- $p \geq 1$ players
- Can all players help Player 1 win the game?

Competitive Dominoes

p-COMP-DOM = $\left\{ \langle \text{Instance of Dominoes} \rangle \middle| \text{P1 has a winning strategy} \right\}$

- $p \geq 2$ players
- Does player 1 have a winning strategy?

Perfect information: all dominoes visible

Player 1 starts.

We will show

1. One-player dominoes is in P
2. Two-player cooperative dominoes is NP-complete
3. Two-player competitive dominoes is PSPACE-complete
4. (p-player cooperative dominoes is NP-complete)
5. (p-player competitive dominoes is PSPACE-complete)

Theorem

One-player dominoes $\in$ P

Proof.

One-player dominoes $\leq_p$ Eulerian Path[1]

Numbers on dominoes $\rightarrow$ Vertices

Dominoes $\rightarrow$ Edges

1	2
2	3
3	2
2	2
2	4

Eulerian path[1] exists in G $\iff$ valid domino chain exists

[1] allowing for revisiting vertices

Cooperative Dominoes

2 players; can all players help Player 1 win the game?

Theorem

2-player cooperative dominoes (2P-COOP-DOM) is NP-complete.

Proof.

1) 2P-COOP-DOM $\in$ *NP*:

Given a move sequence, verify whether Player 1 wins.

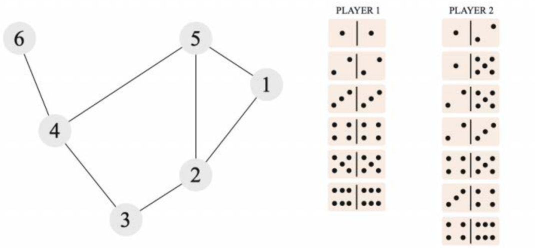
2) 2P-COOP-DOM is NP-hard:

Reduction from Hamilton Path Problem (HP) $\square$

HP $\leq_p$ 2*P-COOP-DOM*

Given (connected) graph $G = (V, E)$, construct dominoes instance:

$$T_1 = \{\{i, i\} | i \in V\} \text{ and } T_2 = \{\{i, j\} | (i, j) \in E\} \cup \{\{*, *\}\}$$



(a) An instance of Hamiltonian path

(b) Corresponding instance of 2-player cooperative dominoes

Fig. 1. Reduction from Hamiltonian path to 2-player cooperative dominoes

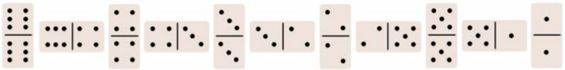
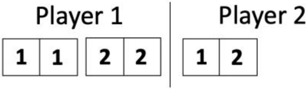


Fig. 2. Hamiltonian path represented as domino chain

Edge Case

$$T_1 = \{\{i, i\} | i \in V\} \text{ and } T_2 = \{\{i, j\} | (i, j) \in E\} \cup \{\{*, *\}\}$$



G is connected so there are at least $|V| - 1$ edges $\Rightarrow$ Player 2 cannot get rid of his dominoes before Player 1 with dummy domino

Competitive Dominoes

2 players; does player 1 have a winning strategy?

Theorem

2-player competitive dominoes (2P-COMP-DOM) is in PSPACE.

Proof.

Convert into instance of the Formula Game Problem □

Formula Game

Given quantified Boolean formula in prenex normal form

$$\phi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 ... Q x_k [\psi]$$

and two Players E, A

Definition (Formula Game (associated with ϕ))

Player E ($\exists$) and Player A ($\forall$) take turns selecting variables $x_1, ..., x_k$

If ψ evaluates to *TRUE*, E wins. Else, A Wins.

Formula Game

$$\mathit{FORMULA\text{-}GAME} = \{\langle\phi\rangle \mid \text{Player E has a winning strategy in the formula game associated with } \phi\}$$

Example

$$\exists x_1 \forall x_2 \exists x_3 [(x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3})]$$

Player E has a winning strategy: Set $x_1 = 1$ and $x_3 = \neg x_2$

Formula Game

Theorem

FORMULA-GAME is in PSPACE

Proof (Sketch).

Recursively check for all variables if subformula has a winning strategy by trying out all assignments. Only needs to save one possible assignment at a time.[1] □

[1] ∈ PSPACE proven by Sipser via QBF [Sip97]

2P-COMP-DOM $\leq_p$ *FORMULA-GAME*

Every instance of dominoes can be transformed into a formula game problem in PSPACE.

Define variables $X_{i,j,k,l}$ representing i-th domino, placed in j-th direction, at k-th position at the l-th turn ($\mathcal{O}(n^3)$ many variables).

- Generate $\psi = (F_1 \wedge F_2 \wedge F_3) \vee \neg F_4$ which is satisfied iff Player 1 wins a valid game OR if Player 2 makes a wrong move.

  F1: Satisfied iff Player 1 moves correctly.

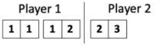
  F2: Satisfied iff domino chain is correct.

  F3: Satisfied iff Player 1 won.

  F4: Satisfied iff Player 2 moves correctly.

- Generate $\phi = \exists X_{\_,\_,\_,1} \forall X_{\_,\_,\_,2} ... QX_{\_,\_,\_,n}[\psi]$

Formula Game: example



Polynomial Space?

- All constraints/checks in ϕ consist of one variable or a pair of variables
- $\mathcal{O}(n^3)$ many variables $\Rightarrow \binom{n^3}{2}$ many possible pairs $\Rightarrow \psi$ is at most of length $\mathcal{O}(n^6)$
- We can write the formula down in polynomial space

PSPACE completeness

Theorem

2-player competitive dominoes is PSPACE-complete

To show:

$\in$ PSPACE: Lemma 4

hardness: This section

PSPACE hardness

Theorem

2-player competitive dominoes (2P-COMP-DOM) is PSPACE-hard

$$\text{BIPARTITE-GG} \leq_p 2P\text{-}COMP\text{-}DOM \tag{1}$$

[1] Bipart-GG: PSPACE hardness proven by Lichtenstein & Sipser via QBF [LS80]

PSPACE hardness

Theorem

2-player competitive dominoes (2P-COMP-DOM) is PSPACE-hard

$$\text{BIPARTITE-GG} \leq_p 2\textit{P-COMP-DOM} \tag{1}$$

Bipartite Generalized geography (Bipart-GG):[1]:

[1] Bipart-GG: PSPACE hardness proven by Lichtenstein & Sipser via QBF [LS80]

PSPACE hardness

Theorem

2-player competitive dominoes (2P-COMP-DOM) is PSPACE-hard

$$\text{BIPARTITE-GG} \leq_p 2P\text{-}COMP\text{-}DOM \tag{1}$$

Bipartite Generalized geography (Bipart-GG):[1]:

- given: $G = A \dot{\bigcup} B$, directed, bipartite: $E \subseteq A \times B$; $a^* \in A$

[1] Bipart-GG: PSPACE hardness proven by Lichtenstein & Sipser via QBF [LS80]

PSPACE hardness

Theorem

2-player competitive dominoes (2P-COMP-DOM) is PSPACE-hard

$$\text{BIPARTITE-GG} \leq_p 2P\text{-}COMP\text{-}DOM \tag{1}$$

Bipartite Generalized geography (Bipart-GG):[1]:

- given: $G = A \dot{\bigcup} B$, directed, bipartite: $E \subseteq A \times B$; $a^* \in A$
- 2 players alternate moving token along edges, to unvisited nodes

[1] Bipart-GG: PSPACE hardness proven by Lichtenstein & Sipser via QBF [LS80]

PSPACE hardness

Theorem

2-player competitive dominoes (2P-COMP-DOM) is PSPACE-hard

$$\text{BIPARTITE-GG} \leq_p 2P\text{-}COMP\text{-}DOM \qquad (1)$$

Bipartite Generalized geography (Bipart-GG):[1]:

- given: $G = A \dot{\bigcup} B$, directed, bipartite: $E \subseteq A \times B$; $a^* \in A$
- 2 players alternate moving token along edges, to unvisited nodes
- Start: Player A at vertex a^*

[1] Bipart-GG: PSPACE hardness proven by Lichtenstein & Sipser via QBF [LS80]

PSPACE hardness

Theorem

2-player competitive dominoes (2P-COMP-DOM) is PSPACE-hard

$$\text{BIPARTITE-GG} \leq_p 2P\text{-}COMP\text{-}DOM \tag{1}$$

Bipartite Generalized geography (Bipart-GG):[1]:

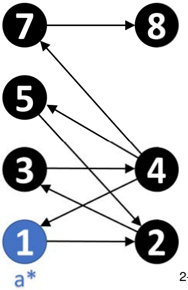
- given: $G = A \dot{\bigcup} B$, directed, bipartite: $E \subseteq A \times B$; $a^* \in A$
- 2 players alternate moving token along edges, to unvisited nodes
- Start: Player A at vertex a^*
- Player who cannot move loses

[1] Bipart-GG: PSPACE hardness proven by Lichtenstein & Sipser via QBF [LS80]

Bipart-GG: example

Rules

- given: $G = A \dot{\bigcup} B$, directed, bipartite: $E \subseteq A \times B$; $a^* \in A$
- 2 players alternate moving token along edges, to unvisited nodes
- Start: Player A at vertex a^*
- Player who cannot move loses



Given $G = A \dot{\bigcup} B, \quad \forall (a, b) \in E \quad (b, a) \in E$:

Edge (a,b) $\longrightarrow$ Domino [a|b] to player A

Edge (b,a) $\longrightarrow$ Domino [a|b] to player B

(Direction encoded by who gets which domino)

Given $G = A \dot{\bigcup} B$, $\forall (a, b) \in E \quad (b, a) \in E$:

Edge (a,b) $\longrightarrow$ Domino [a|b] to player A

Edge (b,a) $\longrightarrow$ Domino [a|b] to player B

(Direction encoded by who gets which domino)

$\hookrightarrow$ Differences to care about:

BIPARTITE-GG	$\hat{=}$	**2p-COMP-DOM**
Blocking opponent (cannot move token)	$\hat{=}$	Blocking opponent or getting rid of all dominoes
One-sided chain	$\hat{=}$	Two-sided chain (queue)
Start at a^*	$\hat{=}$	A starts with any of his dominoes

Eliminate Win by Getting Rid

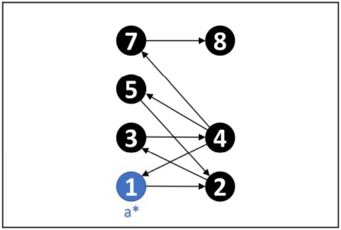
Nonsense domino [#|#] to both players:

Player 1	Player 2
[#|#]	[#|#]

- can only be connected to other player’s nonsense [#|#]
- nobody would play *nonsense* in first turn:

$\Rightarrow$ nobody runs out of dominoes $\Rightarrow$ only win by blocking opponent

Make Chain One-Sided



Player 2 starts in domino

- Give Player 2 *start-domino* [a^*| :-)] (":-)" unique)

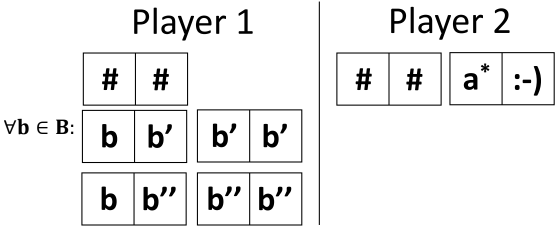
Player 1

#	#

Player 2

#	#	a^*	:-)

Force Start-Domino



- Give Player 1 *garbage dominoes* $\forall b \in B$: [b,b’], [b’,b’], [b,b”], [b”,b”]

Lemma: this domino distribution enforces P2 to play *start domino* first [2]

---

[2] Assuming P2 makes first move.

Lemma: this domino distribution enforces P2 to play *start domino* first [2]

Proof.

P2 cannot start with nonsense domino (see before).

P2 cannot start with an edge Domino:

Case 1: P2 starts with [a|b], $a \neq a^*$.

Case 2: P2 starts with [a|b], $a = a^*$.

□

[2] Assuming P2 makes first move.

PSPACE hardness

winning strategy in Bipartite-GG $\iff$ winning strategy in (transformed) domino

Proof sketch.

PSPACE hardness

winning strategy in Bipartite-GG $\Longleftrightarrow$ winning strategy in (transformed) domino

Proof sketch.

"$\Rightarrow$" winning strategy in bipartite-GG. $\Rightarrow$ edges $\hat{=}$ dominoes $\Rightarrow$ strategy in domino game

PSPACE hardness

winning strategy in Bipartite-GG $\Longleftrightarrow$ winning strategy in (transformed) domino

Proof sketch.

"$\Rightarrow$" winning strategy in bipartite-GG. $\Rightarrow$ edges $\hat{=}$ dominoes $\Rightarrow$ strategy in domino game

"$\Leftarrow$" P1 winning strategy in dominoes $\Rightarrow$ dominoes $\hat{=}$ edges $\Rightarrow$ strategy bipartite GG

$\square$

PSPACE hardness

winning strategy in Bipartite-GG $\iff$ winning strategy in (transformed) domino

Proof sketch.

"$\Rightarrow$" winning strategy in bipartite-GG. $\Rightarrow$ edges $\hat{=}$ dominoes $\Rightarrow$ strategy in domino game

"$\Leftarrow$" P1 winning strategy in dominoes $\Rightarrow$ dominoes $\hat{=}$ edges $\Rightarrow$ strategy bipartite GG

$\square$

Polynomial time ✓

BIPARTITE-GG $\leq_p$ 2P-COMP-DOM

$\square$

Corollary

p-player cooperative dominoes is NP-complete for any fixed $p \geq 2$

- $\in$ *NP*: certificate
- *NP*-hard: p-player can simulate any 2-player game

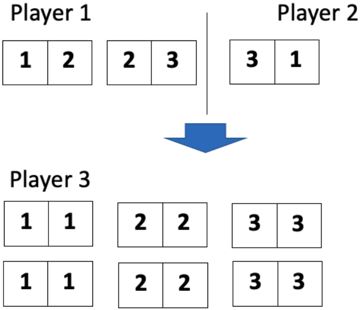
Corollary

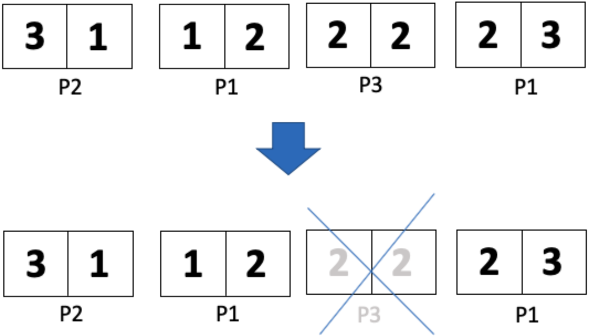
p-player competitive dominoes is PSPACE-complete for any fixed $p \geq 2$

- $\in$ *PSPACE* as in Lemma 4
- *PSPACE*-hard: p-player can simulate any 2-player game

- p-player can simulate any 2-player game by introducing p-2 "null" players
- give each null player a domino [*a* | *a*] for each face *a* that appears in the set of dominos of player 1 and 2

Example (p = 3)





3-player game corresponds to 2-player game

Conclusion

- Determined complexity of domino game under different variants
- Single-Player is easy (in P)
- All other variants are intractable
- Caveats:
  - No Passing
  - Uneven number of dominoes
  - Multisets
- Can the model be extended?

Appendix: GG is PSPACE hard (and complete)

GO is Polynominal-Space Hard 395

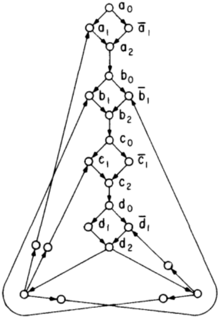


FIGURE 1

representing universally quantified variables), and the other player chooses which path to take through ∃-diamonds. After all diamonds have been traversed, the ∀-player chooses a clause, and the ∃-player then chooses a variable from that clause. ∃ then wins immediately if the chosen variable satisfies the clause; otherwise, ∀ wins on the next move. It follows easily that ∃ wins iff B is true, and we leave the details to the reader. □

Figure: From [LS80]

Bibliography

- J-C Billaut, Federico Della Croce, Fabio Salassa, and Vincent T'kindt, *No-idle, no-wait: when shop scheduling meets dominoes, eulerian paths and hamiltonian paths*, Journal of Scheduling **22** (2019), no. 1, 59–68.
- David Lichtenstein and Michael Sipser, *Go is polynomial-space hard*, Journal of the ACM (JACM) **27** (1980), no. 2, 393–401.
- Michael Sipser, *Introduction to the theory of computation*, 314.