Studio sul Gioco del Domino a due giocatori

Alex Lorenzato

6 marzo 2025

Indice

1 Introduzione **2**
1.1 Il Domino 2
1.2 Storia del Domino 3
1.3 Variante "Block" 4
1.4 Altre varianti 5

2 Teoria dei giochi **6**
2.1 L'albero di gioco 6
2.2 Minimax 8
2.3 Minimax a informazione incompleta 10
2.4 Algoritmi Greedy e Random 13

3 Statistiche **14**
3.1 Statistiche Globali 15
3.2 Statistiche di set specifici 16
3.3 Partite con una sola tessera giocata 20
3.4 Partite con tutte le foglie a segno uguale 22
3.5 Partite a foglia singola (deterministiche) 23

4 Risultati partite **26**
4.1 Modalità con mani selezionate 26
4.2 Modalità Torneo 26

5 Bibliografia **28**

6 Conclusioni **29**

Capitolo 1

Introduzione

1.1 Il Domino

Il domino è un antico gioco da tavolo le cui origini risalgono alla Cina imperiale. Le tessere di gioco sono rettangolari e suddivise in due metà: su ciascuna metà è presente un valore numerico che va da 0 a 6, rappresentato tipicamente con dei pallini, o "pip", che indicano il punteggio della tessera. Le tessere formano una combinazione unica di numeri, con 28 tessere totali in un set "double-six" (rappresentato in figura 1.1), ovvero la cui tessera di maggior valore è quella in cui appaiono due 6.

L'obiettivo generale del gioco, sebbene esistano diverse varianti, è di liberarsi di quante più tessere possibili dalla propria mano prima della fine della partita.

Il gioco procede a turni in senso orario: ogni giocatore, nel suo turno, ha la possibilità di aggiungere una tessera dalla propria mano all'inizio o alla fine della sequenza di tessere presenti sul tavolo, rispettando la condizione che il valore su una delle estremità della tessera coincida con il valore della tessera a cui si collega.

Nella variante classica, la partita inizia con il giocatore che possiede la tessera doppia con il valore più alto, di conseguenza il doppio-6 è la tessera che garantisce di partire per primi. Questa versione è particolarmente popolare nella modalità a 2 giocatori, ma può essere giocata anche da un massimo di 4 o 6 giocatori, con regole e adattamenti per gestire il numero di tessere e i turni.

Nella figura 1.2 è rappresentato un piccolo esempio di partita in corso:

- Il giocatore 1 è il verde, che ha iniziato giocando la tessera 6|6
- Il giocatore 2, rosso, ha scelto di giocare la tessera 6|5, le alternative erano la 6|2 e 6|3
- Il giocatore 1 risponde scegliendo la tessera 5|2, l'alternativa era la 4|5
- Ora è il turno del giocatore 2 che ha 3 mosse dispnibili: la 6|2 può andare sia in testa che in coda, oppure la 6|3

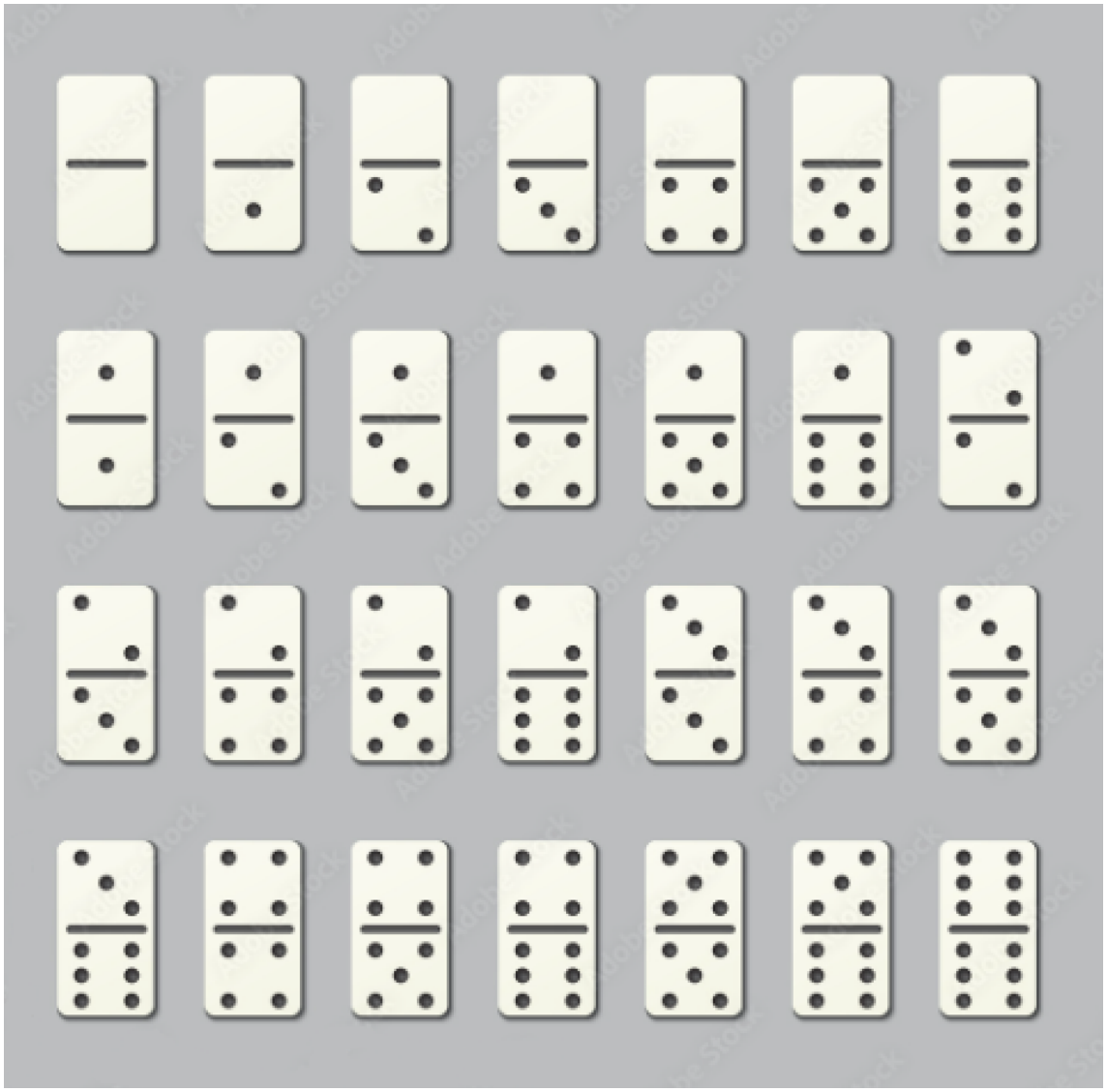

Figura 1.1: Set base di tessere del domino

1.2 Storia del Domino

Il domino con tessere nacque in Cina intorno al XII secolo e si ritiene che sia stato sviluppato da uno statista nel 1120, come dono per l'imperatore Hui Tsung.

Inizialmente, il domino aveva anche un ruolo come strumento di divinazione intorno al XIII secolo, diventando poi un gioco da tavolo vero e proprio.

Nel corso del tempo, il gioco del domino si diffuse dal continente asiatico all'Europa, probabilmente grazie agli scambi culturali mediati dal mondo arabo. Giunto in Italia, il gioco si diffuse rapidamente in Francia e successivamente in tutta Europa. Il termine "domino" deriva dal latino *dominus*, che significa "padrone", e sembra essere legato al senso di controllo e strategia che il gioco richiede.[1]

[1]Fonte: Wikipedia, "Domino", https://it.wikipedia.org/wiki/Domino (accesso il 25 ottobre 2024).

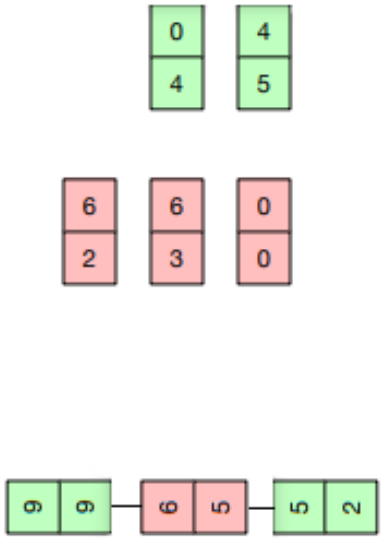

Figura 1.2: Esempio di partita in corso

1.3 Variante "Block"

La variante "Block" è la versione del domino utilizzata come oggetto di studio in questa tesi, nonché la variante più diffusa e popolare.

Questa versione utilizza un set di tessere "double-six", ovvero tessere con valori che vanno da 0 a 6 su ciascuna metà.

Ogni giocatore pesca sette tessere all'inizio del gioco, e le rimanenti 14 tessere vengono scartate e non utilizzate per tutta la durata della partita.

La partita termina in due casi:

1. Quando un giocatore riesce a liberarsi di tutte le tessere in mano, vincendo automaticamente.

2. Quando la partita è "bloccata", ossia nessuno dei due giocatori può posizionare una tessera valida.

In caso di "blocco", vince il giocatore con il minor punteggio totale in mano, dove il punteggio è calcolato come la sommatoria dei valori presenti sulle tessere rimanenti in mano ai giocatori.

Se un giocatore riesce a esaurire tutte le sue tessere, è dichiarato automaticamente vincitore.

Un esempio: il giocatore 1 a fine partita ha in mano le tessere $0|4, 6|5$ e il giocatore 2 la tessera $3|4$; allora il giocatore 1 accumula 15 punti mentre il giocatore 2 solo 7 punti; lo scopo quindi è avere un punteggio il più basso possibile.

1.4 Altre varianti

Assieme alla variante "Block", esiste la variante "Draw", che differisce per l'uso delle tessere non distribuite ai giocatori.

Mentre nella "Block" le tessere avanzate vengono rimosse definitivamente dalla partita, nella variante "Draw" i giocatori possono pescare tessere aggiuntive quando non riescono a fare una mossa valida.

Le modalità di pesca variano: in alcune versioni il giocatore è obbligato a pescare fino a trovare una tessera giocabile, mentre in altre può pescare solo una tessera per turno.

Queste varianti danno origine a una vasta gamma di regole aggiuntive e variazioni del gioco.

Esistono inoltre numerose versioni del domino fan-made e competitive, ognuna con modifiche più o meno significative alle regole delle varianti "Block" e "Draw".

Per esempio, oltre all'obiettivo tradizionale di esaurire la propria mano, alcune modalità premiano configurazioni particolari di tessere sul tavolo, rendendo il gioco ancora più strategico e vario.[2]

[2]Fonte: dominorules.it, *Varianti nel domino*, `https://www.dominorules.com/the-basics` (accesso il 17 Febbraio 2025).

Capitolo 2

Teoria dei giochi

La teoria dei giochi è una disciplina che studia modelli matematici di interazione strategica tra agenti razionali.

La teoria dei giochi ha applicazioni in vari campi delle scienze sociali, così come nella logica, nella teoria dei sistemi e nell'informatica.

Sebbene originariamente si sia focalizzata sui giochi a somma zero, in cui i guadagni o le perdite di ogni partecipante sono perfettamente bilanciati da quelli degli altri, la teoria dei giochi contemporanea si applica ad una vasta gamma di relazioni comportamentali e indica ormai genericamente la scienza delle decisioni logiche negli esseri umani, negli animali e nei calcolatori.[1]

Lo studio del gioco del domino ricade pienamente nella disciplina della teoria dei giochi, nelle prossime sezioni ne verranno approfonditi alcuni aspetti tecnici. [2]

2.1 L'albero di gioco

Un albero di gioco è una struttura utilizzata per rappresentare tutte le possibili sequenze di mosse in un gioco a turni, in questo caso il domino, partendo da una situazione iniziale fino a tutte le configurazioni finali possibili, dette "foglie" dell'albero.

Ogni possibile situazione di gioco è rappresentata da un "nodo" dell'albero e ciascun "ramo" rappresenta una possibile mossa effettuata dal giocatore corrente.

Il nodo radice rappresenta lo stato iniziale, con le mani iniziali dei giocatori e il turno del primo giocatore.

Ogni nodo discendente rappresenta uno stato raggiungibile in base alla tessera che un giocatore gioca.

L'albero di gioco ha lo scopo di esplorare tutte le mosse possibili per individuare le mosse ottimali, in modo da massimizzare la probabilità di vittoria.

Accade molto facilmente che un albero di gioco si allarghi moltissimo, questo comporta un peso computazionale maggiore, ed è per questo che si possono adottare delle ottimizzazioni algoritmiche per ridurne la complessità e di conseguenza le tempistiche di calcolo.

[1] Fonte: Wikipedia, *Teoria dei giochi*, `https://it.wikipedia.org/wiki/Teoria_dei_giochi` (accesso il 12 Dicembre 2024).

[2] Fonte: Game Theory 4th edition, *Teoria dei giochi*

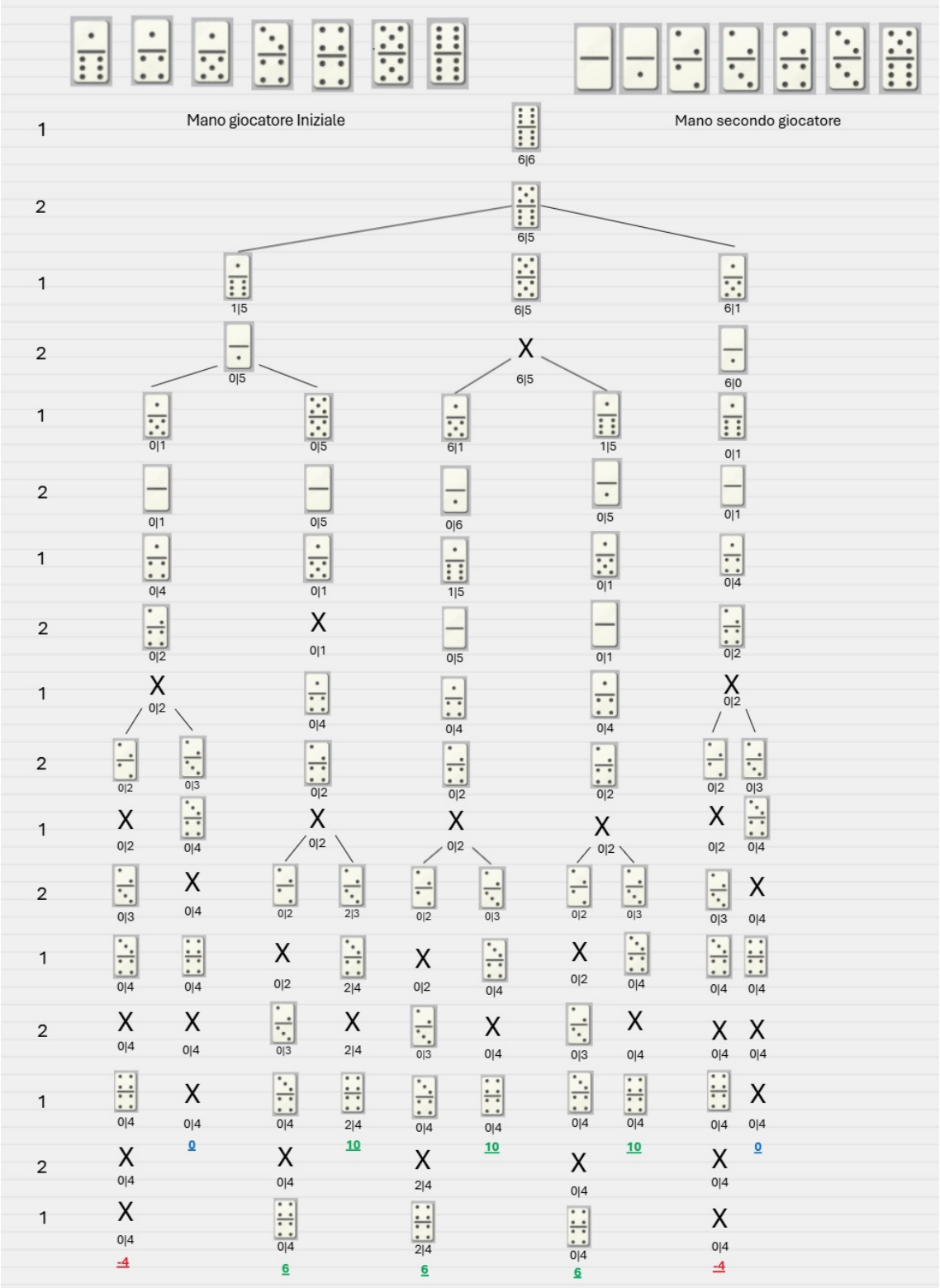

Figura 2.1: Esempio di albero di gioco

In figura 2.1 si può vedere un esempio di albero da gioco, gli elementi sono i seguenti:

1. i set di tessere rappresentati in cima sono le mani iniziali dei due giocatori
2. il 6|6 in cima all'albero è il nodo radice
3. i numeri rappresentati subito sotto ad ogni tessera sono il valore della testa e della coda nel tavolo facendo quella mossa
4. le foglie sono le tessere dove l'albero si interrompe, ciascuna rappresenta un fine partita differente
5. il numero sotto alle foglie rappresenta il punteggio di fine partita: positivo se vince il primo giocatore e viceversa

Il valore delle foglie è calcolato con la formula:
$punti_giocatore_iniziale - punti_secondo_giocatore$

2.2 Minimax

Il principio del minimax, nella teoria dei giochi, consiste nel massimizzare il proprio punteggio minimizzando nel contempo quello dell'avversario.

Questo approccio presuppone che entrambi i giocatori siano perfettamente razionali, ossia giochino sempre la mossa migliore possibile a loro disposizione. In un contesto reale questo è difficile che accada proprio per la complessità di calcolo derivante dallo studiare la mossa perfetta.

Nei giochi a turni, il principio del minimax assume la forma di algoritmo minimax: un algoritmo ricorsivo per la ricerca della mossa migliore in una determinata situazione.

L'algoritmo minimax è costituito da una funzione di valutazione posizionale che misura la bontà di una posizione (o stato del gioco) e indica quanto è desiderabile per il dato giocatore raggiungere quella posizione.

Il giocatore fa poi la mossa che minimizza il valore della migliore posizione raggiungibile dall'altro giocatore. [3]

Minimax solitamente è rappresentato da una struttura ad albero il cui nodo radice è la situazione iniziale, le diramazioni portano ai nodi successivi che rappresentano le possibili mosse e di conseguenza le situazioni che ne derivano.

Reiterando questo processo si arriva fino alle foglie, dove l'albero smette di crescere in quanto il fine partita è stato raggiunto.

Una volta calcolate tutte le foglie, che ricordo rappresentano situazioni di fine partita, il Minimax valuta il punteggio di ciascuna di esse e "risale" l'albero partendo dal punteggio delle foglie, associando di volta in volta ai rami e ai nodi intermedi dei valori forniti da una funzione di valutazione, fino ad arrivare al nodo radice dove ora può scegliere la mossa migliore per Max visto che conosce i punteggi che ogni ramo può garantirgli.

[3] Fonte: Wikipedia, *Minimax*, `https://it.wikipedia.org/wiki/Minimax` (accesso il 12 Dicembre 2024).

Ogni foglia dell'albero è associata a un punteggio tendente al positivo se vince il giocatore iniziale e viceversa, che rappresenta l'esito finale della partita per i due giocatori.

Per rendere il Minimax più efficiente si può utilizzare la tecnica di potatura alpha-beta, che consente di eliminare nodi dell'albero che non influenzano la decisione finale, riducendo drasticamente il numero di nodi da esaminare.

Il calcolo di questa stima può essere migliorato se è possibile fornire una funzione di valutazione euristica che valuti le posizioni non terminali del gioco senza dover conoscere tutte le mosse successive: in questo caso si possono considerare solo un certo numero di mosse future. Questo numero è detto "profondità" dell'algoritmo ed è misurato in mosse.

```
Funzione Minimax(depth, is_maximizing):
    Se checkEndGame() == true O depth == MAX_DEPTH:
        Restituisci evaluateGameScoring()

    moves = availableMoves(current_player)

    Se moves e vuoto:
        playTile(null)
        eval = Minimax(depth + 1, !is_maximizing)
        undo()
        Restituisci eval

    Se is_maximizing:
        max_eval = -INF
        last_move = null
        unswap_flag = false

        Per ogni move in moves:
            tile_to_play = p_hands[current_player][move]

            Se tileNeedsSwap(last_move, tile_to_play):
                tile_to_play.swapTile()
                unswap_flag = true

            playTile(tile_to_play)
            eval = Minimax(depth + 1, false)
            undo()

            Se unswap_flag == true:
                tile_to_play.swapTile()
                unswap_flag = false

            max_eval = max(max_eval, eval)
            last_move = tile_to_play

        Restituisci max_eval

    Altrimenti:
        min_eval = +INF
        last_move = null
```

```
        unswap_flag = false

        Per ogni move in moves:
            tile_to_play = p_hands[current_player][move]

            Se tileNeedsSwap(last_move, tile_to_play):
                tile_to_play.swapTile()
                unswap_flag = true

            playTile(tile_to_play)
            eval = Minimax(depth + 1, true)
            undo()

            Se unswap_flag == true:
                tile_to_play.swapTile()
                unswap_flag = false

            min_eval = min(min_eval, eval)
            last_move = tile_to_play

        Restituisci min_eval
```

Listing 2.1: Pseudocodice del giocatore Minimax ad informazione completa

2.3 Minimax a informazione incompleta

L'algoritmo Minimax descritto precedentemente assume che entrambi i giocatori abbiano informazione perfetta, ovvero che conoscano esattamente lo stato del gioco, comprese le tessere in mano all'avversario e tutte le possibili mosse disponibili.

Tuttavia, in molti giochi, tra cui il Domino, i giocatori non conoscono le tessere dell'avversario, il che introduce una situazione di informazione incompleta. In questo contesto, l'algoritmo deve essere in grado di ragionare su un insieme ipotetico di mosse possibili per il secondo giocatore.

Per gestire questa situazione, è necessario:

1. Definire un insieme di tessere candidate per il secondo giocatore, ossia tutte le tessere non possedute dal primo giocatore. Poiché non si conosce la sua mano, si assume che possa avere qualsiasi tessera ancora in gioco.

2. Considerare il caso in cui l'avversario debba passare il turno. Se il secondo giocatore non può effettuare alcuna mossa valida, significa che non possiede nessuna delle tessere attualmente giocabili. Di conseguenza, nel sottoalbero generato da questa situazione, tali tessere possono essere escluse dal set candidato, poiché è certo che non le possiede (altrimenti le avrebbe giocate).

Nella pratica, questo Minimax segue la struttura dell'algoritmo classico, con la differenza che, oltre a esplorare i sottoalberi derivanti dalle mosse che l'avversario potrebbe giocare, si aggiunge la possibilità di esplorare un sottoalbero in cui l'avversario passa senza giocare nulla, se esistono le condizioni per farlo.

La condizione che permette di esplorare questa possibilità dipende dalla dimensione del set candidato rispetto alla mano dell'avversario:

- All'inizio della partita, è quasi sempre necessario considerare il caso in cui l'avversario passi, poiché il set candidato è molto ampio e potrebbero esserci molte tessere che non può giocare.
- Con il progredire del gioco, alcune tessere vengono scartate dal set candidato (ad esempio, se l'avversario passa quando le estremità del tavolo sono 5—6, è possibile escludere tutte le tessere contenenti 5 o 6 dal suo set candidato). Man mano che il set candidato si restringe, diventa più precisa la stima delle tessere effettivamente in suo possesso.
- Se il numero di tessere candidate si riduce fino a eguagliare la dimensione della mano dell'avversario, significa che non vi è più incertezza: egli possiede esattamente quelle tessere e quindi non può passare. In questo caso, l'opzione di passare viene esclusa dall'analisi dell'algoritmo.

Dopo aver esplorato il caso della mossa vuota, l'algoritmo risale l'albero e ripristina le tessere precedentemente rimosse per garantire la correttezza dell'analisi nei livelli successivi della ricerca.

```
    Funzione AlphaBeta(brd, opp_hand_size, alpha, beta):
    best_value = -INF
    move_value = +INF
    moves = brd.getCurrentPlayerMoves()

    Se brd.getCurrentState() == ENDED O opp_hand_size == 0:
    Restituisci brd.getScoreDifference()

    Se brd.getCurrentPlayer() == FIRST:  # Turno gioc. principale (
       Max)
        Per ogni move in moves:
            brd.playTile(move)
            move_value = AlphaBeta(brd, opp_hand_size, alpha, beta)
            brd.unplayTile()

            best_value = max(best_value, move_value)
            alpha = max(alpha, best_value)

            Se alpha >= beta:
                Interrompi ciclo

        Restituisci best_value

    Altrimenti:   Turno dell avversario
    move_value = -INF
    best_value = +INF

    Per ogni move in moves:
        brd.playTile(move)

        Se move e vuoto:
            move_value = AlphaBeta(brd, opp_hand_size, alpha, beta)
        Altrimenti:
```

```
33              move_value = AlphaBeta(brd, opp_hand_size - 1, alpha,
                    beta)
34
35              brd.unplayTile()
36
37              best_value = min(best_value, move_value)
38              beta = min(beta, best_value)
39
40              Se beta <= alpha:
41                  Interrompi ciclo
42
43          Se moves.size() > 1 O moves[0] != vuoto:
44              new_excluded_list = brd.removeMatches()
45
46              Se brd.getNumOfTiles(SECOND) >= opp_hand_size:
47                  brd.playTile(TILE_VUOTO)
48                  move_value = AlphaBeta(brd, opp_hand_size, alpha,
                        beta)
49                  brd.unplayTile()
50
51                  best_value = min(best_value, move_value)
52
53              brd.restoreMatches(new_excluded_list)
54
55          Restituisci best_value
```

Listing 2.2: Pseudocodice del giocatore Minimax ad informazione incompleta

Differenze tra il codice ad informazione completa e incompleta: l'algoritmo Minimax a informazione completa assume che entrambi i giocatori conoscano esattamente tutte le tessere in gioco, comprese quelle nella mano dell'avversario. Questo permette di simulare l'intero albero delle possibilità con certezza, valutando ogni mossa in base alle configurazioni effettive.

Al contrario, l'algoritmo Minimax a informazione incompleta introduce l'incertezza sulle tessere possedute dall'avversario. Per affrontare questa mancanza di informazione, l'algoritmo ipotizza che l'avversario abbia in mano tutte le tessere possibili e gestisce il caso in cui un giocatore sia costretto a passare il turno. Un elemento distintivo di questa implementazione è il controllo sulla dimensione della mano dell'avversario, che viene ridotta progressivamente man mano che si effettuano ipotesi sulle tessere giocate.

Un'ulteriore differenza chiave è che il Minimax a informazione incompleta deve gestire le situazioni in cui l'avversario non può giocare alcuna tessera, esplorando scenari alternativi e aggiornando la lista delle tessere possibili. Questo introduce una maggiore complessità computazionale rispetto al Minimax a informazione completa, che non ha bisogno di tali ipotesi.

In sintesi, mentre il Minimax a informazione completa fornisce decisioni ottimali basate su dati certi, il Minimax a informazione incompleta deve gestire un livello di incertezza, rendendo necessaria una gestione più avanzata della ricerca delle mosse e una modellizzazione delle possibili mani avversarie.

2.4 Algoritmi Greedy e Random

Per testare l'efficacia del Minimax sono stati utilizzati anche altri due tipi di giocatori molto semplici: Greedy e Random.

L'algoritmo Random è banale, gioca in maniera casuale limitandosi a seguire le regole del gioco.

Il Greedy invece è un approccio molto diffuso nella teoria dei giochi e negli algoritmi, gioca la mossa più vantaggiosa nell'immediato, ma senza considerare sviluppi futuri. Nel caso del Domino questo si traduce nel giocare la tessera, ovviamente tra quelle permesse, con il valore più alto in modo da "liberarsi" di più punti possibile. In apparenza è una buona strategia, i limiti sono dati dal non guardare alle contromosse avversarie o da possibili sviluppi futuri; una mossa avida (traduzione di Greedy) nel momento sbagliato può aprire possibilità all'avversario che altrimenti non avrebbe avuto.

Capitolo 3

Statistiche

In questa sezione vengono analizzati i dati raccolti da uno studio su un ampio numero di partite simulate utilizzando l'algoritmo Minimax. L'obiettivo è identificare statistiche e comportamenti rilevanti in ottica di sviluppare un giocatore perfetto.

Un singolo test consiste nel generare due mani da 7 tessere ciascuna e simulare una partita tramite l'algoritmo Minimax che andrà a formare l'intero albero di gioco, da radice a foglie, di fatto simulando ogni possibile scenario che potrebbe presentarsi usando le mani iniziali generate; l'output, corrispondente in larga parte ai valori trovati nelle foglie generate da minimax, viene scritto in un file di testo e contiene:

1. le mani dei due giocatori
2. i valori delle foglie trovati tramite la simulazione (se la partita termina senza che i due giocatori abbiano svuotato la mano il valore è contrassegnato da una X)
3. il valore maggiore tra le foglie
4. la durata in millisecondi della partita

Un esempio di output è:
6|6 1|6 2|4 2|6 3|3 3|4 3|6 0|0 0|1 0|2 0|3 0|5 0|6 1|5 -3X 2 2 -3X 1 -3X 1 1 -19 86

Complessivamente sono stati eseguiti 40.000.320 test, sebbene le partite possibili siano molte di più.

Il primo calcolo da affrontare difatti è stato quello del totale di partite diverse esistenti nel domino usando il set da 28 tessere.

Questo valore è dato dal numero di mani iniziali diverse che si possono distribuire tra i 2 giocatori.

Il numero totale di configurazioni delle mani iniziali, corrispondente al numero di partite diverse giocabili nel domino è: 137.281.098.240; più precisamente, questo è il numero di set da 14 tessere formabili senza che ci sia mai un set che sia formato dalle stesse tessere di un altro, che si traduce nel distribuire queste 14 tessere tra i 2 giocatori avendo sempre delle mani differenti.

Questo numero è calcolato nel seguente modo:

1. Possibili set da 14 tessere: $\binom{28}{14} = 40.116.600$, ai quali bisogna sottrarre tutti i set che non contengono nessuna tessera doppia.

2. Set che non contengono nessuna tessera doppia: $\binom{21}{14} = 116.280$.

3. Possibili set da 14 tessere validi per poter iniziare una partita: $40.116.600 - 116.280 = 40.000.320$, ovvero il totale di set possibili esclusi quelli senza una tessera doppia.

4. Ognuna delle 40.000.320 possibili partite è costituita da un set di 14 tile, che però possono essere distrubuite in $\binom{14}{7} = 3432$ modi diversi tra i 2 giocatori

5. Il numero totale di partite esistenti con un set da 28 tile e mani da 7 tile ciascuna quindi è: $40.000.320 * 3432 = 137.281.098.240$

Testare l'algoritmo su tutte le possibili partite non è temporalmente possibile con gli strumenti a disposizione, questo perché supponendo una media di 70ms a partita (media calcolata sui dati raccolti) ci vorrebbero $137.281.098.240 * 70 = 9.609.676.876.800$ ms, equivalenti a 304.7 anni su un sistema monoprocessore; avendo avuto a disposizione un server con 40 processori, il tempo di calcolo necessario sarebbe stato $304/40 \tilde{=} 7.5$ anni.

Per trovare un compromesso temporale sulla raccolta dati, sul totale delle partite teoriche, ho preso un campione di 40.000.320 configurazioni iniziali per fare in modo che ogni possibile set fosse rappresentato, e testato l'algoritmo su queste partite raccogliendone i dati.

Il processo è stato parallelizzato dividendo il file in sotto-file da 1 milione di righe.

3.1 Statistiche Globali

Alcune statistiche globali raccolte:

Durata media della partita	71.56 ms
Numero medio di foglie	1023.36
Percentuale di vittorie del giocatore iniziale	100.00%
Percentuale di partite in cui nessun giocatore esauriva la mano	48.13%

La partita con il maggior numero di foglie in assoluto, ovvero 32.172, è quella in cui i giocatori hanno le seguenti mani:

Mano giocatore 1: $0|2, 1|2, 1|5, 2|2, 2|6, 4|4, 5|6$

Mano giocatore 2: $0|4, 1|4, 1|6, 2|4, 2|5, 4|5, 4|6$

Un'altra statistica interessante è che ci sono 356.142 partite con una sola foglia, equivalenti al 9%, questo accade perché un giocatore ha una doppia tessera con cui partire ma poi né lui né l'avversario hanno una tessera che si può attaccare a quella inizialmente giocata.

Il numero di foglie è indice della complessità della partita, con una sola foglia la partita è banale e non necessita che vengano prese decisioni, al contrario se si hanno migliaia di foglie la partita permette molta libertà d'azione e l'ordine in cui vengono giocate le tessere assume molta più importanza.

Un esempio banale di partita ad una sola foglia è dato da un giocatore che parte con la tessera doppio-6 e nessuno dei due giocatori ha altre tessere in cui compaia un 6, la partita quindi termina immediatamente procedendo al calcolo del punteggio.

Non è comunque garantito che un alto numero di foglie indichi una partita complessa, spesso si presentano dinamiche per cui l'ordine in cui vengono giocate le tessere è ininfluente ai fini del risultato finale; questo tipo di partite hanno solo una complessità apparente, ma di fatto sono banali quanto quelle con poche foglie.

3.2 Statistiche di set specifici

Ho investigato dei set specifici di tessere alla ricerca di mani più o meno interessanti o vantaggiose in termini di partite vinte.

Questa ricerca è stata condotta lanciando dei test su tutte le possibili partite in cui un giocatore ha la mano specificata e analizzandone l'albero di gioco risultante.

Mano: $0|0, 0|1, 0|2, 0|3, 0|4, 0|5, 0|6$

Match considerati	116280
Durata media delle partite	47.87 ms
Numero medio di foglie	2390.60
Percentuale di vittorie del giocatore iniziale	100.00%
Percentuale di partite in cui nessun giocatore esauriva la mano	24.54%

Tabella 3.1: Statistiche per la mano $0|0, 0|1, 0|2, 0|3, 0|4, 0|5, 0|6$

Questa configurazione di mano iniziale è chiaramente vantaggiosa dato che sono state vinte tutte le partite affrontate.

Soffre di uno svantaggio iniziale, ovvero la difficoltà a partire per primi, visto che qualsiasi tessera doppia garantirebbe la precedenza su questa mano; superato questo difetto, si ha una mano che riesce ad agganciarsi ad ogni tessera giocata visto che si hanno tutti i numeri da 0 a 6, inoltre è molto facile connettere le tessere tra loro usando il numero 0 come "aggancio"; un ulteriore vantaggio è quello di chiudere le possibilità all'avversario, visto che non potrà avere tessere con il numero 0 in quanto le ha tutte il giocatore con questa mano, ne segue che una delle due estremità del tavolo è spesso bloccata dal numero 0, e anche la seconda estremità è facilmente bloccabile agganciando un qualsiasi numero.

Mano: $0|0, 1|1, 2|2, 3|3, 4|4, 5|5, 6|6$

Ho scelto di investigare questa mano per la particolarità di avere tutte le doppie, i risultati non sono stati buoni visto che il rateo di vittoria è di 1 su 3.

Match considerati	116280
Durata media delle partite	22.51 ms
Numero medio di foglie	66.74
Percentuale di vittorie del giocatore iniziale	28.91%
Percentuale di partite in cui nessun giocatore esauriva la mano	64.41%

Tabella 3.2: Statistiche per la mano $0|0, 1|1, 2|2, 3|3, 4|4, 5|5, 6|6$

Il motivo per cui questa mano non funziona bene potrebbe essere dato dal fatto che di fatto non ha possibilità di prendere decisioni significative; ogni volta che collega una tessera, il tavolo rimane invariato, di conseguenza è l'avversario che tramite delle tessere diversificate può gestire la partita in maniera autonoma, il giocatore con questa mano non può fare altro che attaccare le tessere che ha senza mai influire sulle estremità del tavolo.

Presenta solo due vantaggi, chiaramente insufficienti: garanzia di iniziare per primo e avere tutti i numeri in mano, quindi potersi attaccare facilmente ad ogni situazione, con lo svantaggio di poterlo fare una volta sola per numero.

Mano: $0|0, 0|1, 0|2, 0|3, 1|1, 1|2, 1|3$

Match considerati	116280
Durata media delle partite	40.97 ms
Numero medio di foglie	1609.93
Percentuale di vittorie del giocatore iniziale	60.97%
Percentuale di partite in cui nessun giocatore esauriva la mano	41.50%

Tabella 3.3: Statistiche per la mano $0|0, 0|1, 0|2, 0|3, 1|1, 1|2, 1|3$

Questa mano è caratterizzata da una distribuzione di tessere dai valori bassi, i risultati sono relativamente scarsi, appena al di sopra del tiro di una moneta in quanto a probabilità di vittoria. Il potenziale di questa mano era quello di avere tessere ben connesse e coese tra loro, cercando di tagliare le possibilità all'avversario andando a creare un "circuito chiuso". Un grosso svantaggio è quello di avere poche probabilità di iniziare per primi, andando a limitare fin dall'inizio la possibilità di creare un tavolo formato da tessere basse.

Mano: $3|4, 4|4, 4|5, 5|5, 4|6, 5|6, 6|6$

Match considerati	116280
Durata media delle partite	34.62 ms
Numero medio di foglie	801.53
Percentuale di vittorie del giocatore iniziale	17.12%
Percentuale di partite in cui nessun giocatore esauriva la mano	50.13%

Tabella 3.4: Statistiche per la mano $3|4, 4|4, 4|5, 5|5, 4|6, 5|6, 6|6$

Mano: $3|3, 3|4, 4|4, 2|2, 2|3, 2|4, 4|5$

Match considerati	116280
Durata media delle partite	37.43 ms
Numero medio di foglie	1013.82
Percentuale di vittorie del giocatore iniziale	12.17%
Percentuale di partite in cui nessun giocatore esauriva la mano	49.17%

Tabella 3.5: Statistiche per la mano $3|3, 3|4, 4|4, 2|2, 2|3, 2|4, 4|5$

Mano: $0|1, 1|1, 1|2, 1|3, 1|4, 1|5, 1|6$

Match considerati	116280
Durata media delle partite	52.85 ms
Numero medio di foglie	2580.79
Percentuale di vittorie del giocatore iniziale	100.00%
Percentuale di partite in cui nessun giocatore esauriva la mano	23.38%

Tabella 3.6: Statistiche per la mano $0|1, 1|1, 1|2, 1|3, 1|4, 1|5, 1|6$

Mano: $0|2, 1|2, 2|2, 2|3, 2|4, 2|5, 2|6$

Match considerati	116280
Durata media delle partite	54.74 ms
Numero medio di foglie	2843.77
Percentuale di vittorie del giocatore iniziale	100.00%
Percentuale di partite in cui nessun giocatore esauriva la mano	22.15%

Tabella 3.7: Statistiche per la mano $0|2, 1|2, 2|2, 2|3, 2|4, 2|5, 2|6$

Mano: $0|3, 1|3, 2|3, 3|3, 3|4, 3|5, 3|6$

Match considerati	116280
Durata media delle partite	57.32 ms
Numero medio di foglie	3199.19
Percentuale di vittorie del giocatore iniziale	99.99%
Percentuale di partite in cui nessun giocatore esauriva la mano	20.97%

Tabella 3.8: Statistiche per la mano $0|3, 1|3, 2|3, 3|3, 3|4, 3|5, 3|6$

Mano: $0|4, 1|4, 2|4, 3|4, 4|4, 4|5, 4|6$

Match considerati	116280
Durata media delle partite	60.73 ms
Numero medio di foglie	3670.20
Percentuale di vittorie del giocatore iniziale	99.93%
Percentuale di partite in cui nessun giocatore esauriva la mano	19.93%

Tabella 3.9: Statistiche per la mano $0|4, 1|4, 2|4, 3|4, 4|4, 4|5, 4|6$

Mano: $0|5, 1|5, 2|5, 3|5, 4|5, 5|5, 5|6$

Match considerati	116280
Durata media delle partite	65.31 ms
Numero medio di foglie	4283.70
Percentuale di vittorie del giocatore iniziale	99.82%
Percentuale di partite in cui nessun giocatore esauriva la mano	19.11%

Tabella 3.10: Statistiche per la mano $0|5, 1|5, 2|5, 3|5, 4|5, 5|5, 5|6$

Mano: $0|6, 1|6, 2|6, 3|6, 4|6, 5|6, 6|6$

Match considerati	116280
Durata media delle partite	71.24 ms
Numero medio di foglie	5070.79
Percentuale di vittorie del giocatore iniziale	99.89%
Percentuale di partite in cui nessun giocatore esauriva la mano	18.58%

Tabella 3.11: Statistiche per la mano $0|6, 1|6, 2|6, 3|6, 4|6, 5|6, 6|6$

Ho approfondito le configurazioni del tipo $x|0$ ... $x|6$ perché dopo aver notato la % di vittorie del set $0|0$ ... $0|6$ era ipotizzabile che anche gli altri set "mononumerici" potessero ottenere risultati simili e così è stato.

Il motivo per cui questa tipologia di set è così prestante risiede nella sua efficacia di bloccare l'avversario, ad esempio: se possiedo tutte le tessere con il 6 l'avversario ovviamente non ne avrà, aggiungendo il fatto che oltre ad avere tutti i 6 sull'altra faccia ho tutti i numeri, sono pronto a rispondere a qualsiasi mossa venga giocata, e nel rispondere è implicito che chiuderò una delle due estremità all'avversario, che

ora non potrà attaccare nulla al 6 che ho giocato; ripetendo questa cosa anche per l'altra estremità è banale chiudere entrambe le estremità all'avversario, facendogli saltare il turno.

Queste configurazioni sono dunque le più vittoriose offrendo certezza di vittoria nel caso in cui le mosse giocate siano sempre quelle ottimali; in un contesto reale, quindi a informazione imperfetta, non è comunque detto che la vittoria sia necessariamente garantita usando uno di questi set: una mossa sbagliata potrebbe portare a un ramo dell'albero perdente, ovvero in cui il punteggio è negativo.

Degli esempi di partite in cui il set 0|6...6|6 perde sono:

- 0|6 1|6 2|6 3|6 4|6 5|6 6|6, 0|0 0|2 1|1 2|2 2|5 3|3 4|4
- 0|6 1|6 2|6 3|6 4|6 5|6 6|6, 0|0 0|1 0|2 1|1 1|2 3|3 4|4
- 0|6 1|6 2|6 3|6 4|6 5|6 6|6, 0|1 1|1 1|2 1|3 1|4 1|5 3|3
- 0|6 1|6 2|6 3|6 4|6 5|6 6|6, 0|0 0|1 0|2 0|3 0|4 0|5 1|1
- 0|6 1|6 2|6 3|6 4|6 5|6 6|6, 0|4 1|1 1|4 2|4 3|4 4|4 4|5

3.3 Partite con una sola tessera giocata

Tra tutte le possibili partite ce ne sono alcune in cui non si ha alcuna possibilità di scelta, questo perché nessuno ha tessere da attaccare alla testa o alla coda dopo la prima mossa oppure perché ad ogni turno c'é solo una mossa disponibile.

Le partite in cui viene giocata una sola tessera seguono uno schema preciso: la partita inizia giocando una tessera doppia, successivamente nessuno dei due giocatori possiede delle tessere che raffigurino il numero della tessera giocata alla prima mossa, difatto terminando immediatamente la partita.

Il numero di partite in cui viene giocata una sola tessera è calcolabile, di seguito spiegati i passaggi dell'esempio in cui viene giocato solamente il 6|6:

1. le tessere utilizzabili sono 22, questo perché oltre al 6|6 le altre sei tessere che rappresentano il numero 6 non possono essere usate nella distrubuzione delle mani dei gocatori o si andrebbe a negare il fatto che verrà giocata una sola tessera

2. di queste 22 tessere una è il 6|6, che diamo in mano al giocatore 1

3. ci sono $\binom{21}{6} = 54.264$ modi di distribuire le restanti 6 tessere al giocatore 1

4. e $\binom{15}{7} = 6.435$ modi di distruibuire la mano al giocatore 2

5. il totale di partite in cui un giocatore ha un 6|6 e nessun altro dei due ha un'altra tessera con un 6 sono: $54.264 \cdot 6.435 = 349.188.840$

6. questo numero va raddoppiato in quanto il procedimento appena eseguito va applicato anche al giocatore 2, potrebbe essere lui ad avere il 6|6 e non il giocatore 1, quindi $349.188.840 \cdot 2 = 698.377.680$

Lo stesso procedimento va fatto per i numeri da 0 a 5, facendo un'ulteriore sottrazione: nel calcolo del 5|5 ad esempio vanno tolti tutti i set che contengono la tessera 6|6, mentre nel calcolo del 4|4 vanno tolti sia quelli con il 6|6 che quelli con il 5|5, e così via; di seguito i calcoli.

Caso in cui venga giocato solo il 5|5:

1. le tessere utilizzabili sono 21 perché devo rimuovere la tessera 6—6 e tutte le tessere che riportano il numero 5 (ad eccezione della 5|5
2. di queste 21 tessere una è la 5|5, che do in mano al giocatore 1
3. ci sono $\binom{20}{6} = 38.760$ modi di distribuire le restanti 6 tessere al giocatore 1
4. e $\binom{14}{7} = 3.432$ modi di distruibuire la mano al giocatore 2
5. il totale di partite in cui un giocatore ha un 5|5, nessuno dei due ha un'altra tessera con un 5 e non è mai presente la 6|6 sono: $38.760 \cdot 3.432 = 133.024.320$

Caso con il 4|4:

1. 20 tessere utilizzabili, rimosse la 6|6, 5|5 e quelle con il 4
2. tessera 4|4 in mano al giocatore 1
3. $\binom{19}{6} = 27.132$ modi di distribuire le restanti 6 tessere al giocatore 1
4. e $\binom{13}{7} = 1.716$ modi di distruibuire la mano al giocatore 2
5. totale: $27.132 \cdot 1.716 = 46.558.512$

Caso con il 3|3:

1. 19 tessere utilizzabili, rimosse la 6|6, 5|5, 4|4 e quelle con il 3
2. tessera 3|3 in mano al giocatore 1
3. $\binom{18}{6} = 18.564$ modi di distribuire le restanti 6 tessere al giocatore 1
4. e $\binom{12}{7} = 792$ modi di distruibuire la mano al giocatore 2
5. totale: $18.564 \cdot 792 = 14.702.688$

Caso con il 2|2:

1. 18 tessere utilizzabili, rimosse la 6|6, 5|5, 4|4, 3|3 e quelle con il 2
2. tessera 2|2 in mano al giocatore 1
3. $\binom{17}{6} = 12.376$ modi di distribuire le restanti 6 tessere al giocatore 1
4. e $\binom{11}{7} = 330$ modi di distruibuire la mano al giocatore 2
5. totale: $12.376 \cdot 330 = 4.084.080$

Caso con il 1|1:

1. 17 tessere utilizzabili, rimosse la 6|6, 5|5, 4|4, 3|3, 2|2 e quelle con l'1
2. tessera 1|1 in mano al giocatore 1
3. $\binom{16}{6} = 8.008$ modi di distribuire le restanti 6 tessere al giocatore 1
4. e $\binom{10}{7} = 120$ modi di distruibuire la mano al giocatore 2
5. totale: $8.008 \cdot 120 = 960.960$

Caso con il 0|0:

1. 16 tessere utilizzabili, rimosse la 6|6, 5|5, 4|4, 3|3, 2|2, 1|1 e quelle con lo 0
2. tessera 0|0 in mano al giocatore 1
3. $\binom{15}{6} = 5.005$ modi di distribuire le restanti 6 tessere al giocatore 1
4. e $\binom{9}{7} = 36$ modi di distruibuire la mano al giocatore 2
5. totale: $5.005 \cdot 36 = 180.180$

Concludendo, le partite in cui viene giocata una sola tessera sono $698.377.680 + 133.024.320 + 46.558.512 + 14.702.688 + 4.084.080 + 960.960 + 180.180 = 897.999.420$ che corrispondono allo 0.65% del totale delle partite totali.

3.4 Partite con tutte le foglie a segno uguale

Un'altra tipologia di partite interessanti sono quelle con tutte le foglie con segno uguale: tutte positive, tutte negative o tutte uguali a zero.

Questo tipo di partite è interessante perché per quante foglie possa avere l'albero, la partita è in verità banale: se tutte le foglie sono positive significa che qualsiasi mossa venga fatta, sarà il giocatore 1 a vincere; al limite si può massimizzare o minimizzare il punteggio, ma la vittoria è innegabile.

Ragionamento speculare si può fare se sono tutte negative o se sono tutte uguali a zero, caso particolare in cui il pareggio è inevitabile.

Partite con sole foglie negative: 309.634
Esempio:
Mano Giocatore 1: $0|1, 1|2, 1|3, 1|4, 1|6, 4|6, 5|5$
Mano Giocatore 2: $0|0, 0|2, 0|3, 0|4, 0|5, 0|6, 1|1$

Partite con sole foglie positive: 890.965
Esempio:
Mano Giocatore 1: $0|1, 0|2, 0|3, 0|6, 1|2, 1|5, 6|6$
Mano Giocatore 2: $0|0, 0|4, 0|5, 1|1, 1|3, 1|4, 2|2$

Partite con sole foglie = 0: 14.051
Esempio:
Mano Giocatore 1: $2|3, 2|4, 2|5, 3|3, 3|5, 3|6, 6|6$
Mano Giocatore 2: $0|3, 0|4, 1|1, 1|2, 3|4, 4|5, 5|5$

Sommando queste partite si ottiene $309.634 + 890.965 + 14.051 = 1.214.650$, che rapportato ai 40.000.320 di partite testate corrisponde a un 3.03% del totale delle partite testate, un numero non indifferente, indice del fatto che quasi 1 partita su 30 è banale.

3.5 Partite a foglia singola (deterministiche)

Un ultimo caso di partite particolari è quello in cui l'albero di gioco ha una foglia singola.

Se l'albero di gioco ha una foglia sola è perché ad ogni turno la mossa è obbligata: non avere mai più di un opzione nel proprio turno implica che l'albero non si dirama mai in più di una direzione, portando l'albero ad avere una foglia soltanto.

Questa tipologia di partite è estremamente banale in quanto non ci sono decisioni da prendere.

Partite con una sola foglia: 356.142
Esempio:
Mano Giocatore 1: $0|5, 2|3, 3|4, 4|5, 4|6, 5|5, 6|6$
Mano Giocatore 2: $0|0, 0|4, 1|1, 1|2, 2|2, 2|5, 3|3$

Di seguito sono rappresentati dei grafici, il primo dei quali è quello generale, gli altri sono ingrandimenti di specifici intervalli.

Sull'asse x è rappresentato il numero di foglie e sull'asse y il numero di partite.

Si nota subito il picco sul valore 0, indicando che ci sono 356.142 partite con una sola foglia, corrispondenti approssimativamente a un 9% del totale delle partite testate.

Curiosa l'alternanza dei valori, probabilmente solo una coincidenza che non porta a nulla di significativo.

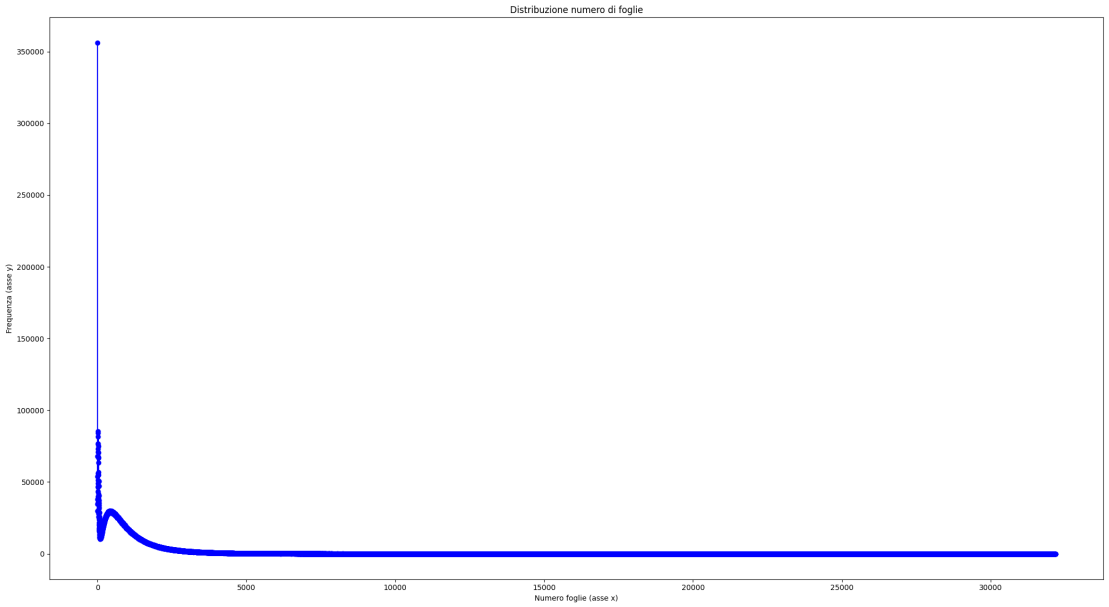

Figura 3.1: Grafico generale

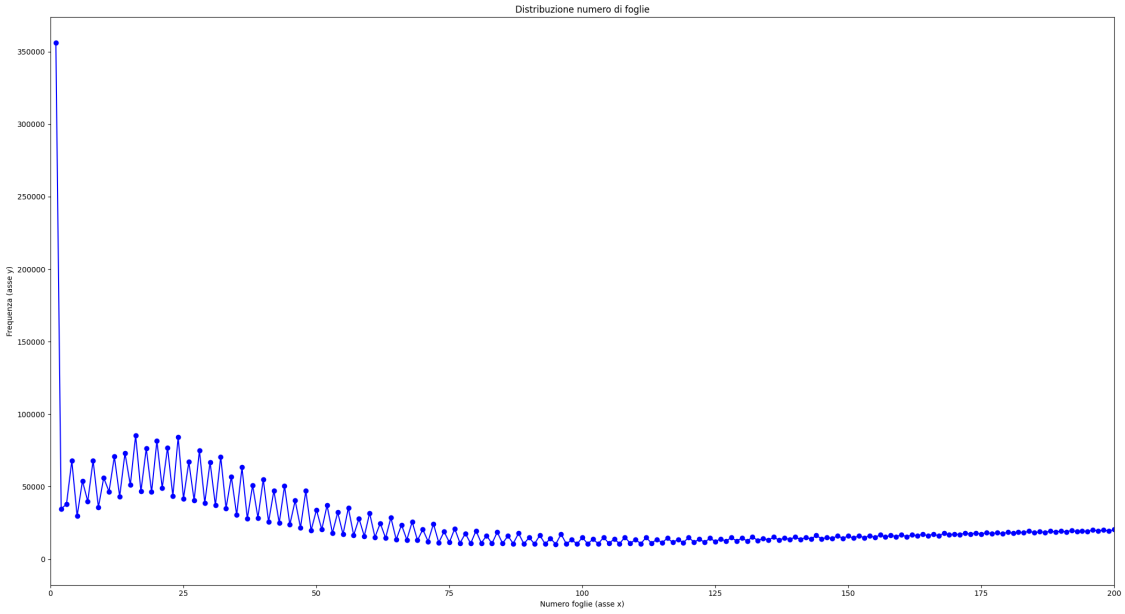

Figura 3.2: Ingrandimento dominio 0-200

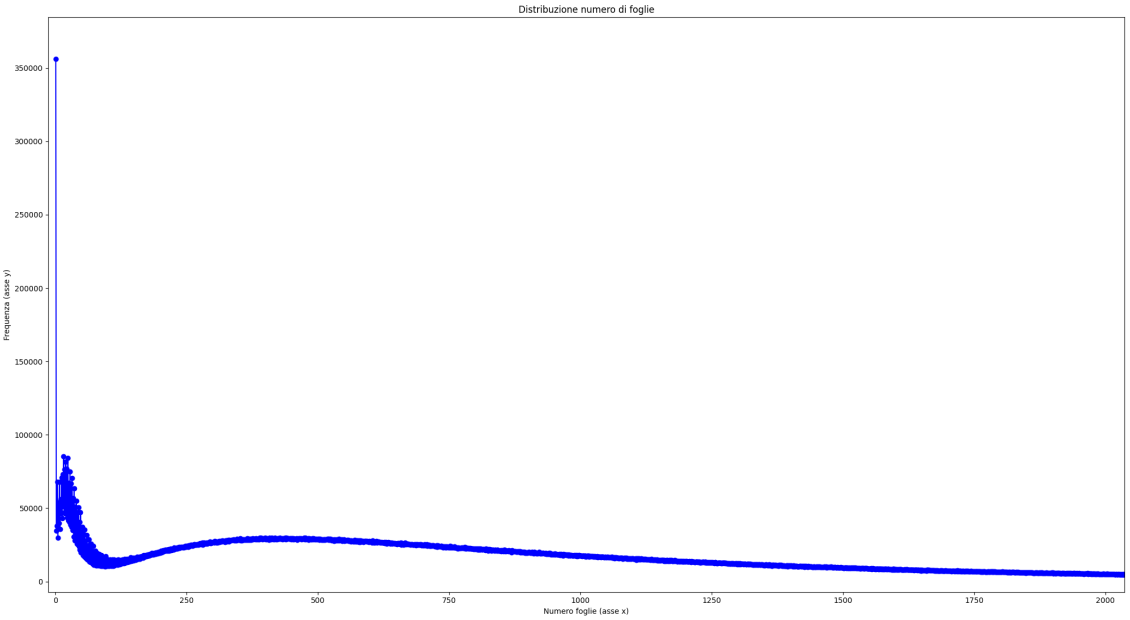

Figura 3.3: Ingrandimento dominio 0-2000

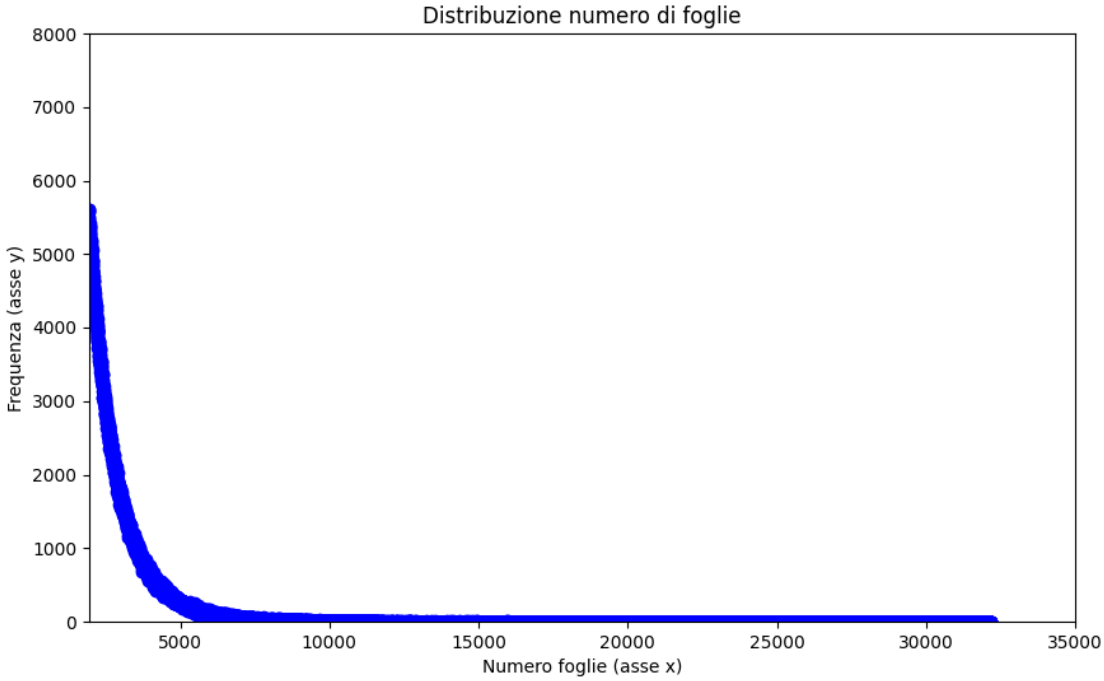

Figura 3.4: Ingrandimento dominio 2000-35000

Capitolo 4

Risultati partite

4.1 Modalità con mani selezionate

	Greedy	Random	Minimax	avg
Greedy	-	2.65	0.01	1.33
Random	0.48	-	-1.02	-0.27
Minimax	2.26	3.98	-	3.12
avg	1.37	3.31	-0.5	-

Tabella 4.1: Risultati della modalità con mani selezionate

In questa modalità sono state testate delle partite con mani iniziali selezionate: sono state scelte 1 milione di partite il cui albero avesse il maggior numero di foglie possibile.

Le mani selezionate per questo test sono di fatto le partite più complesse possibili tra quelle esplorate nei test precedenti.

Ogni milione di partite è stato affrontato due volte, mantenendo le stesse mani iniziali ma scambiando l'ordine dei giocatori.

Di seguito sono riportati i risultati ottenuti: ogni cella contiene il punteggio medio di X vs Y, con X sempre primo giocatore e X rappresentato dalla riga.

L'ultima riga e colonna riportano la media delle rispettive righe e colonne.

4.2 Modalità Torneo

La modalità torneo consiste nel far scontrare due giocatori attraverso una serie di partite con lo scopo di raggiungere o superare per primi i 100 punti.

Nel caso entrambi riuscissero a superare la soglia dei 100 punti, il vincitore sarà il giocatore con più punti.

Per ogni coppia di giocatori sono stati giocati 1000 tornei, in ogni partita di ogni torneo la distribuzione delle tessere iniziali è casuale rispettando la regola delle 7 tessere iniziali in mano e il 6 come numero massimo.

I risultati sono riassunti nella tabella che va letta nel seguente modo: il giocatore della riga ha l'X% di vittorie contro il giocatore della colonna.

	Greedy	Random	Minimax
Greedy	-	86.2%	19%
Random	13.8%	-	4%
Minimax	81%	96%	-

Tabella 4.2: Risultati della modalità torneo

Il Minimax è di gran lunga l'algoritmo più performante, seguito dal Greedy e il Random in ultima posizione.

Da notare che nonostante il Minimax utilizzi tecniche e strategie avanzate una mano iniziale può essere compromettente per il risultato finale; questo è stato osservato anche nella fase in cui venivano testati i 40 milioni di partite: alcune configurazioni di mani portavano necessariamente alla sconfitta, le foglie di quelle partite erano tutte negative.

Questo spiega perché anche l'algoritmo Random sia in grado di vincere qualche partita: una mano iniziale fortunata può garantire la vittoria esattamente come una mano sfortunata può garantire la sconfitta.

Il minimax è un algoritmo che al netto di una sconfitta punta comunque a minimizzarnee i danni giocando sempre le mosse migliori possibili.

E' da notare anche la differenza di risultato tra il Greedy e il Minimax contro il Random, per quanto il Random sia debole, il Minimax riesce comunque a costruire strategie migliori e ad avere uno scarto significativo in termini di esito. Questo è dimostrato anche dalla differenza di risultati nelle partite Greedy vs Minimax.

Capitolo 5

Bibliografia

- Osborne, M. J., & Rubinstein, A. (1994). A Course in Game Theory. MIT Press.
- von Neumann, J., & Morgenstern, O. (1944). Theory of Games and Economic Behavior. Princeton University Press.
- Wikipedia. (2025). "Domino". Disponibile su: `https://it.wikipedia.org/wiki/Domino` (Accesso il 24 Febbraio 2025).
- Wikipedia. (2025). "Teoria dei giochi". Disponibile su: `https://it.wikipedia.org/wiki/Teoria_dei_giochi` (Accesso il 24 Febbraio 2025).
- Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.
- Wikipedia, "Minimax" - Minimax
- "Dominorules". (2025). Regole e strategie del Domino. Disponibile su: `https://www.dominorules.com/the-basics` (Accesso il 24 Febbraio 2025).
- Maschler, M., Solan, E., & Zamir, S. (2013). Game Theory. Cambridge University Press.
- Guillermo Owen. (2013). Game Theory 4th edition. Emerald Group Publishing.

Capitolo 6

Conclusioni

Lo studio ha permesso di evidenziare alcuni aspetti strategici e computazionali, fornendo una visione più approfondita del comportamento del gioco e di alcune sue implicazioni.

Uno dei risultati più interessanti riguarda l'analisi delle mani iniziali più forti. Le configurazioni del tipo x|0 ... x|6 si sono rivelate vincenti nella totalità delle simulazioni testate, grazie alla loro capacità di controllare il tavolo e impedire all'avversario di giocare.

Un altro aspetto rilevante riguarda la quantità di partite banali, ovvero quelle in cui non esiste una possibilità di scelta per i giocatori. Lo 0.65% delle partite testate aveva una sola foglia nell'albero di gioco, rendendo la scelta delle mosse deterministica. Allo stesso modo, circa il 3,03% delle partite ha mostrato un esito predeterminato, con tutte le foglie dello stesso segno, indicando che il risultato finale era inevitabile indipendentemente dalle mosse giocate.

Dal punto di vista algoritmico, il confronto tra diverse strategie ha confermato la superiorità dell'algoritmo Minimax rispetto agli approcci Greedy e Random, con una percentuale di vittoria significativamente più alta nei tornei simulati. Tuttavia, la qualità della mano iniziale rimane un fattore determinante: anche Minimax, pur seguendo una strategia ottimale, può essere sconfitto se parte con una configurazione iniziale svantaggiosa.