

Seminar aus maschinellem Lernen

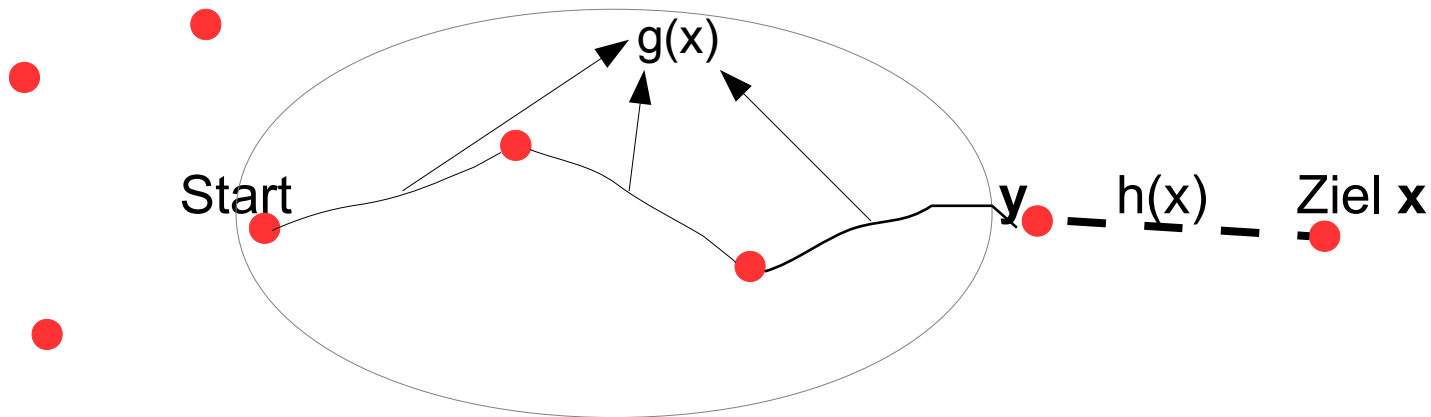
Single Player MCTS



TECHNISCHE
UNIVERSITÄT
DARMSTADT

A* und IDA*

- Traditionelle Herangehensweisen an Single Player spielen
- A* best-first suche mit $f(x) = g(x) + h(x)$
- $h(x)$ ist eine Heuristikfunktion (ohne $h(x)$ ist A* gleich Dijkstra)
- In Sokoban und Eight Puzzle ist $h(x)$ die Manhattan Distance



- IDA* ist eine depth-first variante von A* mit Iterativer Tiefen suche

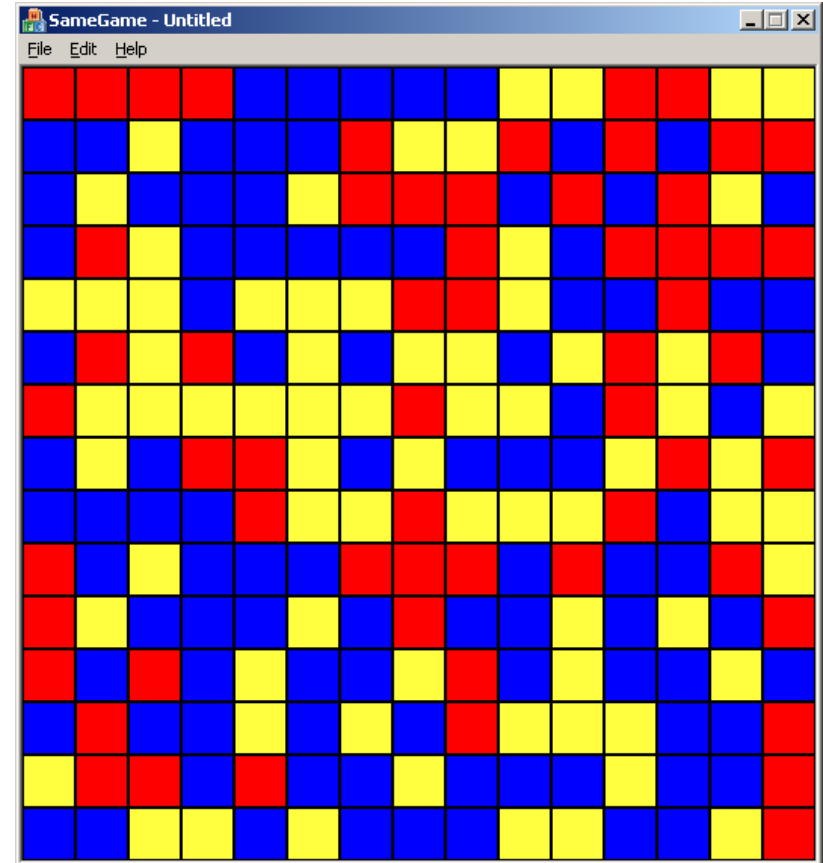
Motivation

Same Game

- Keine gute Heuristik funktion gefunden für A* und IDA*
- Spielfeld 15*15 mit 5 farben
- Bei einer Gruppe von n Feldern bekommt der Spieler: $(n-2)^2$
- $n > 1$
- Bonuspunkte beim zerstören von allen felderern

1	B	C			
2	D	A	C		
3	C	B	B	A	C
4	C	C	B	A	B
5	D	C	C	D	B
	A	B	C	D	E

1	B				
2	D	C			
3	C	A		A	C
4	C	C	C	A	B
5	D	C	C	D	B
	A	B	C	D	E



http://www.cprogramming.com/tutorial/game_programming/samegame_tiles.png

Single Player MCTS

- Basierend auf MCTS wurde SP MCTS erstellt.
- Single-Player (Puzzle) Spiele müssen nicht mit der Unsicherheit des nächsten Zuges des Gegenspielers leben
- Sie können Spielzüge optimieren ohne das der Gegenspieler einen behindert
- In Puzzle Single-Player Spielen gibt es keine “unentschieden”
- In den Papes wurde SP MCTS in vier schritte aufgeteilt.
 - 1) Selection Step
 - 2) Play out Step
 - 3) Expansion Step
 - 4) Backpropagation Step

Selection Step

- Auswahl eines Kindes von einem gegebenem Knoten p .
- Es kontrolliert den Ausgleich zwischen Exploitation (Ausbeutung) und Exploration (Erkundung).
- Exploitation ist die Aufgabe sich auf Züge zu konzentrieren die das beste Ergebnis liefern.
- Exploration kümmert sich um Zügen die noch Unbekannt sind

Selection Step

- SP-MCTS erweitert UCT-Formel (Upper Confidence Bounds applied to Tees)
- Aufgabe vom Selection Step ist die Formel (1) zu Maximiere

$$\underbrace{v_i + C \times \sqrt{\frac{\ln n_p}{n_i}}}_{\text{UCT-Formel}} + \underbrace{\sqrt{\frac{\sum r^2 - n_i \times v_i^2 + D}{n_i}}}_{\text{SP-MCTS Erweiterung}} \quad (1)$$

p: Vaterknoten, i: Kindknoten

np: Stehen für die Anzahl der besuchere (n_i gleich)

v_i : Durchschnitt Belohnung von Knoten i

C: ist für das verhältnis von Exploration und Exploitation zuständig

Selection Step

- Für SP-MCTS Addieren einen dritten term zu der Formel (1), der eine mögliche Abweichung des Kindknotens i darstellt

$$+ \sqrt{\frac{\sum r^2 - n_i \times v_i^2 + D}{n_i}}$$

SP-MCTS Erweiterung

$\sum r^2$: Summe der quadrierten Ergebnisse der Kind Knoten

$n_i \times v_i^2$: Voraussichtliche Ergebnis

D : Konstante die dazu Adiirt wird um dafür zu sorgen das Knoten die selten erweitert wurden als unsicher betrachtet werden

- Play out Step
 - Beginnt dann wenn einen Knoten erreichte der noch nicht teil des Baumes
 - Dann werden Zufällige Spielzüge gewählt bis zum Spielende
 - In Same Game erhält man mehr punkte wenn man eine Große Gruppe der selben Farbe/Elemente
 - Um das zu erreichen wurden zwei Quasi-Zufallsmethoden benutzt:
 - 1) TabuRandom: Am start eine Farbe zufällige wählen und bis zum ende versuchen diese farbe zu vermeiden
 - 2) TabuColorRandom: Am Start Farbe wählen die am häufigsten auftritt und bis zum Ende versuchen zu vermeiden

- Expansion Step
 - Entscheidet welche Knoten abgespeichert werden
 - Erste Position die noch nicht im Baum ist wird gespeichert.
 - Es wird immer ein Kind erweitert
- Backpropagation Step
 - Das Ergebnis des Play Out Step wird von einem Blatt zu der Wurzel zurückgereicht
 - Dadurch muss die Summe aus der Formel (1) aktualisiert werden
 - Update Top Scores (r) und (v_i) Durchschnitt Belohnung von Knoten i

- Final Move Selection
 - Die vier Schitte werden wiederholt bis die Zeit abgelaufen ist
 - Zeit abgelaufen dann für den Final Move entscheiden
 - In Puzzel ist es nicht erforderlich auf die Reaktion des Gegenspieler zu warten
 - Somit ist es möglich eine große suche zu starten die mit dem Startzustand des Spiels beginnt
 - Und dann alle Spielzüge auf einmal zu spielen
 - Mit dieser Herangehensweise werden alle möglichen Spielzüge vom Startzustand berücksichtigt bis die Zeit für SP-MCTS abläuft

Single Player MCTS

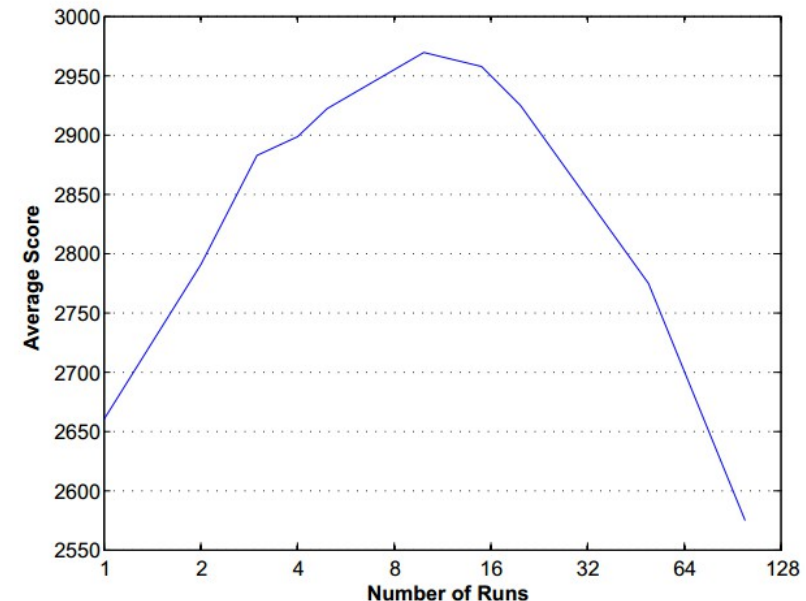
- Simulation Strategy
 - Um die unterschiedlichen simulation strategien TabuRandome und TabuColorRandome auszuwerten wurden 250 Zufällige Start Felder von SameGame erstellt.
 - SP MCTS lief so lange bis 10 million Knoten gespeichert wuden.
 - Diese Durchläufe liefen immer 5 bis 6 Minuten
 - C: 0.5 und D: 10000

Table 3.1: Effectiveness of the simulation strategies.

	Random	TabuRandom	TabuColorRandom
Average Score	2,069	2,737	3,038
StdDev	322	445	479

Single Player MCTS

- Randomized Restarts
 - Um Lokalen Maxima zu vermeiden werden Randomized Restarts mit unterschiedlichen Seeds gewählt (start parametern)
 - Es hat sich herausgestellt das in Same Game tiefe Bäume zu besseren Erlebnissen führen
 - Knoten: 500000
 - C: 0,1 D: 32
 - Bei zwei statt einer suche eine Verbesserung von 140 punkten
 - Das Maximum liegt bei 10 zufälligen Restarts mit 2950 punkten
 - Nach 10 Neustarts ist die Performance wieder stark gesunken



Single Player MCTS

- Manual Parameter Tuning von (C;D)

Table 3.2: Results of SP-MCTS for different settings.

	Exploitation (0.1; 32)	Balanced (0.5; 10,000)	Exploration (1; 20,000)
10^5 nodes (~ 20 seconds)			
Average Score	<u>2,552</u>	2,388	2,197
Standard Deviation	572	501	450
Average Depth	25	7	3
Average Deepest Node	63	19	8
10^6 nodes (~ 200 seconds)	(0.1; 32)	(0.5; 10,000)	(1; 20,000)
Average Score	2,674	<u>2,858</u>	2,579
Standard Deviation	607	560	492
Average Depth	36	14	6
Average Deepest Node	71	<u>37</u>	15
5×10^6 nodes ($\sim 1,000$ seconds)	(0.1; 32)	(0.5; 10,000)	(1; 20,000)
Average Score	2,806	<u>3,008</u>	2,901
Standard Deviation	576	524	518
Average Depth	40	18	9
Average Deepest Node	69	<u>59</u>	20

- M. P. D. Schadd, “Selective Search in Games of Different Complexity,” Ph.D. dissertation, Maastricht Univ., Netherlands, 2011.
- M. P. D. Schadd, M. H. M. Winands, H. J. van den Herik, G. M. J.-B. Chaslot, and J. W. H. M. Uiterwijk, “Single-Player Monte-Carlo Tree Search,” in Proc. Comput. and Games, LNCS 5131, Beijing, China, 2008
- Y. Björnsson and H. Finnsson, “CadiaPlayer: A Simulation-Based General Game Player,” IEEE Trans. Comp. Intell. AI Games, vol. 1, no. 1, 2009

