AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA INFORMATYKI STOSOWANEJ

Praca dyplomowa magisterska

Analiza i porównanie wybranych algorytmów dla gry karcianej
Analysis and comparison of selected algorithms for the card game

Autor: *Damian Malarczyk*
Kierunek studiów: *Informatyka*
Opiekun pracy: *dr inż. Edyta Kucharska*

Kraków, 2015

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.): „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej «sądem koleżeńskim».", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Podziękowania

Spis treści

1. Wprowadzenie

1.1. Przedmiot pracy

Od kilku lat coraz większą popularnością cieszą się wszelkiego rodzaju gry planszowe i karciane. Przyciągają nie tylko coraz lepszą oprawą graficzną, lecz również ciekawą mechaniką pozwalającą na stosowanie różnych taktyk. Z tego powodu stanowią szerokie pole do testowania algorytmów optymalizujących dostępne ruchy tak, by zapewnić zwycięstwo.

Są również gry, w których kluczową rolę odgrywa tak zwana 'intuicja'. Obliczenie całego drzewa dostępnych ruchów jest zbyt skomplikowane i decyzja musi zostać podjęta na podstawie niepełnych danych. Przykładem takiej gry jest Łove Letter", gra niezbyt skomplikowana, jednak zawierająca dużo interakcji i możliwych ścieżek rozwoju sytuacji.

Inspirując się wyżej wymienioną grą, w poniższej pracy porównuję trzy algorytmy podejmowania decyzji w grze:

– probabilistyczny - który będzie podejmował decyzję w sposób losowy na podstawie prawdopodobieństwa wystąpień kart,

– zachłanny - który będzie wybierał zawsze najbardziej prawdopodobny scenariusz,

– Monte Carlo Tree Search - algorytm heurystyczny, który podejmowane decyzje opiera na symulacjach.

1.2. Cel pracy

1.3. Zawartość pracy

Rozdział 1 jest wprowadzeniem definiującym przedmiot pracy. W rozdziale 2 opisana została gra karciana wraz z problemem który przedstawia. Rozdział 3 przedstawia trzy proponowane algorytmy rozwiązujące problem. W rozdziale 4 zebrane są poszczególne etapy tworzenia aplikacji:

– analiza wymagań,

– koncepcja wykonania,

– wykorzystane technologie,

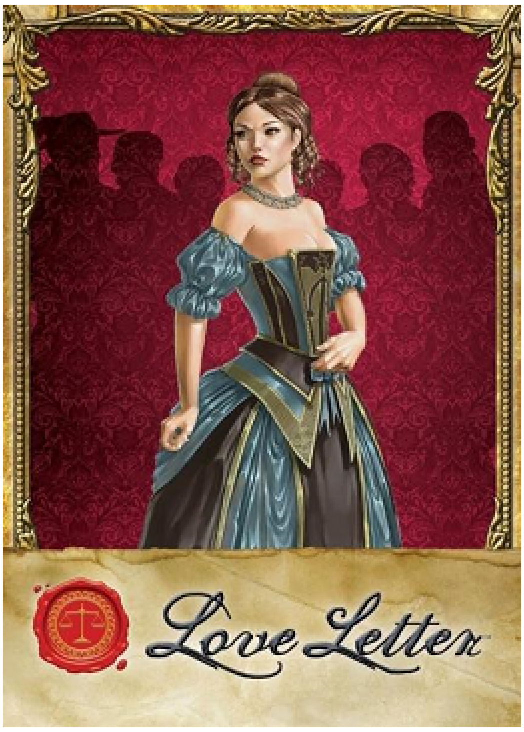
– diagramy,

– prezentacja systemu.

W rozdziale 5 zebrane są statystyki związane z implementacją algortytmów. Rozdział 6 stanowi podsumowanie pracy.

2. Gra karciana Love Letter

W tym rozdziale opisuję kontekst oraz zasady gry Love Letter. Tłumaczę działanie każdej karty oraz przedstawiam główny cel gry - wygranie określonej ilości rund. Następnie formułuję zagadnienie w sposób matematyczny. Wszystkie załączone zdjęcia oraz instrukcja zaczerpnięte są z [1] oraz [2].



Rysunek 2.1. Love Letter - okładka

2.1. Opis zasad gry

W trakcie gry wcielamy się w rolę jednego z adoratorów księżniczki starającego się o zdobycie jej serca. W tym celu przygotowaliśmy list miłosny, który chcemy jej dostarczyć. Niestety, księżniczka pogrążona jest obecnie w żałobie i nie przyjmuje do siebie nikogo obcego, w związku z czym musimy znaleźć inny sposób na przekazanie jej naszego listu. Oprócz księżniczki, na dworze znajdują się inne postacie, z których każda ma mniejszy lub większy dostęp do komnat naszej wybranki i może oddać jej list. Przekazujemy więc naszą przesyłkę swojemu tajnemu posłańcowi, a na koniec gry księżniczka jako pierwszy przeczyta ten list, który został przekazany przez najbardziej zaufaną postać. Serce wybranki zdobywa gracz, który jako pierwszy przekaże w ten sposób od 4 do 7 listów, w zależności od liczby graczy.

Cel i ustawienie początkowe

Love Letter rozgrywa się jako serię rund. Grę wygrywa gracz o następującej ilości wygranych rund:

– 7 w grze na 2 graczy,

– 5 w grze na 3 graczy,

– 4 w grze na 4 graczy.

Ustawienie początkowe każdej rundy wygląda następująco:

– przetasuj karty

– odrzuć 1 wierzchnią kartę nie odkrywając jej (nie bierze udziału w rundzie),

– jeśli gra tylko 2 graczy, odrzuć 3 wierzchnie karty, odkryte,

– rozdaj po 1 karcie wszystkim graczom,

– jeśli jest to pierwsza runda, grę zaczyna gracz, który jako ostatni był na randce, w przeciwnym wypadku zwycięzca poprzedniej rundy.

Tura gracza i opis kart

Podczas swojej tury gracz dociąga jedna kartę ze stosu. Następnie wybiera jedną z dwóch kart, które posiada już na ręce, kładzie ją przed sobą tak, by była widoczna dla wszystkich i zastosowuje opisany na niej efekt - nawet jeśli jest negatywny. Zagrana karta pozostaje odkryta przez całą rundę, a druga pozostaje na ręce. Następnie tura przechodzi na osobę po lewej stronie aktywnego gracza.

W grze znajduje się 16 kart, w 8 typach. Są to kolejno: 4 karty Strażniczki, po 2 karty Kapłana, Barona, Pokojówki i Księcia, oraz po jednej karcie Króla, Hrabiny i Księżniczki. Ich szczegółowy opis wraz z wyglądem znajduje się poniżej:

Rysunek 2.2. Strażniczka

Na rysunku 2.2 przedstawiona jest karta typu Strażniczka. Zagrywając tę kartę należy wskazać jednego z pozostałych graczy i odgadnąć kartę którą posiada. Jeśli karta została prawidłowo odgadnięta, wskazany gracz odrzuca ją i przegrywa rundę.

Rysunek 2.3. Kapłan

Rysunek 2.3 przedstawia kartę typu Kapłan. Zagrywając tę kartę należy podglądnąć kartę wybranego gracza.

Rysunek 2.4. Baron - 2 karty

Na rysunku 2.4 przedstawiona jest karta typu Baron. Po zagraniu tej karty należy w ukryciu porównać drugą posiadaną kartą z wybranym graczem. Następnie ten gracz, który ma kartę o mniejszej wartości odrzuca swoją kartę i przegrywa rundę. W przypadku remisu nic się nie dzieje.

Rysunek 2.5. Pokojówka - 2 karty

Rysunek 2.5 przedstawia kartę typu Pokojówka. Zagranie tej karty sprawia, że gracz jest niewrażliwy na efekt pozostałych kart do czasu swojej następnej tury.

Rysunek 2.6. Książe - 2 karty

Na rysunku 2.6 przedstawiona jest karta typu Książę. Zagranie pozwala wybrać dowolnego gracza (w tym siebie), zmusić go do odrzucenia posiadanej karty i pociągnięcia następnej.

Rysunek 2.7. Król - 1 karta

Rysunek 2.7 przedstawia kartę typu Król. Po jej zagraniu należy wymienić się pozostałą kartą z innym graczem.

Rysunek 2.8. Hrabina - 1 karta

Rysunek 2.8 przedstawia kartę typu Hrabina. Ta karta ma działanie pasywne. Nie wywiera efektu po zagraniu, natomiast zmusza gracza do jej zagrania jeśli równocześnie posiada na ręce kartę typu Książę lub Król.

Rysunek 2.9. Księżniczka - 1 karta

Na rysunku 2.9 przedstawiona jest karta typu Księżniczka. Zagranie tej karty oznacza natychmiastową przegraną w rundzie. Ta zasada działa również, gdy gracz został zmuszony do zagrania tej karty, np. przez efekt karty Książe.

2.2. Analiza złożoności problemu

Z wyżej przedstawionych zasad wynika, że gra cechuje się wysokim stopniem losowości i jest niedeterministyczna. W związku z tym powstaje pytanie, czy istnieje strategia mieszana[4] optymalizująca podejmowanie decyzji w taki sposób, by zwiększać szansę wygrania gry. Dla uproszczenia założyłem, że gra będzie rozgrywana przez dwóch graczy. Dodatkowo, by ustandaryzować pewne pojęcia, wprowadziłem następujące definicje:

– *Zagranie* - jest to typ karty wraz ze sposobem jej wykorzystania. Przykładowo: Zagranie karty typu Strażniczka z wyborem karty typu Król, lub zagranie karty typu Książę z wyborem na gracza przeciwnego.

– *Decyzja* - to zagranie użyte jako ruch w grze. Podjęcie decyzji to inaczej zwrócenie zagrania przez algorytm.

– *Strategia* - inaczej algorytm, który podejmuje decyzje.

– *Scenariusz* - chronologiczny spis decyzji podjętych przez obu graczy, od początku do końca rundy.

By móc opracować najlepszą strategię, musiałbym znać wszystkie możliwe scenariusze gry, na podstawie których można by ustalić statystycznie które zagranie w danym momencie jest najbardziej opłacalne. Jako scenariusz należy rozumieć wszystkie podjęte przez graczy ruchy w danej rundzie. Wpływ na to mają dwa czynniki:

– Kolejność kart w talii na początku rundy

– Sposób zagrania karty

Dokonałem więc oszacowania, ile takich scenariuszy istnieje.

W każdej rundzie bierze udział wszystkie 16 kart. Zakładając, że każda z nich jest unikalna, to liczba wszystkich możliwych kolejności kart to permutacja, którą obliczam wzorem podanym w [3]:

$$P_n = n! \text{ , gdzie } n \in N^+$$

Dla n = 16, $n! = 20922789888000$. Część kart się powtarza, więc tę liczbę należy jeszcze podzielić przez permutacje powtarzających się kart Strażniczki, Kapłana, Barona, Pokojówki oraz Księcia. Razem jest to $4! * 2! * 2! * 2! * 2! = 384$. Ostatecznie wynika, że liczba unikalnych kolejności kart wynosi:

$$20922789888000/32 = 54486432000 \text{ - 54mld, 864mln i 432 tys.}$$

Nie jest to jednak liczba wszystkich dostępnych scenariuszy. Ponieważ w każdej turze gracz zagrywa jedną z dwóch dostępnych kart, a ponadto w przypadku części z nich może podjąć różne decyzje, to kolejność w której gracze dociągną karty z dostępnej talii będzie się wielokrotnie zmieniać w trakcie rozgrywki. Dobrym przykładem jest Książę, który zmusza przeciwnika do odrzucenia posiadanej karty i pociągnięcia następnej.

By wyznaczyć w liczbę scenariuszy na zadanej kolejności kart, posłużyłem się następującym przybliżeniem:

– zgodnie z zasadami gry dla dwóch graczy, odrzucam łącznie 4 pierwsze karty (1 zakryta, 3 odkryte).

– rozdaję po 1 karcie obu graczom. Pozostaje 10 kart w talii.

– zakładając, że żadna decyzja nie spowoduje przerwania rundy, gracze łącznie 10 razy pociągną kartę, więc podejmą 10 decyzji.

– w uproszczeniu, każdą decyzję można przedstawić jako 0 (zagranie posiadanej karty) lub 1 (zagranie pociągniętej karty).

Na podstawie powyższego można oszacować, że możliwych scenariuszy dla danej kolejności kart jest $2^{10} = 1024$. Łącząc tę liczbę z ilością możliwych kolejności kart, utrzymujemy przybliżoną liczbę scenariuszy:

$$54486432000 * 1024 = 55794106368000 \approx 5.8 * 10^{12}$$

Z uwagi na rząd wielkości stworzenie strategii na podstawie analizy statystycznej wszystkich dostępnych scenariuszy jest problemem *NP-trudnym*. Z tego powodu, zamiast odpowiadać na pytanie "Jaka jest najlepsza strategia podejmowania decyzji?", dużo łatwiej będzie odpowiedzieć na pytanie "Która z podanych strategii jest najlepsza?", gdyż jest to charakterystyczna cecha problemów klasy *NP*. Kierując się tą zasadą, w następnym rozdziale opisałem wybrane strategie, których skuteczność sprawdzę implementując je w napisanej przeze mnie aplikacji.

3. Przegląd wybranych algorytmów i innych rozwiązań

W tym rozdziale przedstawiam kilka wybranych algorytmów, które zaimplementuję w następnym rozdziale. Dla każdej z nich opisuję ich użycie w odniesieniu do gry 'Love Letter'. By ustandaryzować pewne pojęcia, wprowadziłem następujące definicje:

– *Zagranie* - jest to typ karty wraz ze sposobem jej wykorzystania. Przykładowo: Zagranie karty typu Strażniczka z wyborem karty typu Król, lub zagranie karty typu Książę z wyborem na gracza przeciwnego.

– *Decyzja* - to Zagranie użyte jako ruch w grze. Podjęcie decyzji to inaczej zwrócenie *Zagrania* przez dany algorytm.

– *Strategia* - inaczej algorytm, który podejmuje decyzje.

3.1. Algorytm losowy

3.1.1. Opis

Jest to najprostszy algorytm podejmowania decyzji. Na wejściu otrzymuje listę dostępnych zagrań, z których w sposób całkowicie losowy podejmuje jedną. Każde z dostępnych zagrań ma takie samo prawdopodobieństwo bycia wybraną przez ten algorytm.

3.1.2. Sposób wykorzystania

Z uwagi na prostotę tego algorytmu, używam go jako punktu odniesienia dla wszystkich pozostałych strategii. Dla każdej z nich algorytm losowy spełnia rolę drugiego gracza i w ten sposób można łatwo porównać je ze sobą. Dla tego algorytmu zastosowałem jedną drobną modyfikację: nigdy nie podejmie decyzji o zagraniu Księżniczki - oznacza to natychmiastową przegraną niezależnie od momentu gry, co mogłoby mocno wypaczyć wyniki innych algorytmów.

3.2. Algorytm zachłanny

3.2.1. Opis

Algorytm zachłanny polega na wyborze najlepszego możliwego zagrania dostępnego w danej chwili, nie analizując jego konsekwencji w przyszłości. Pomimo, że takie podejmowanie decyzji krótkowzroczne, jest łatwe w implementacji i daje atrakcyjne wyniki w niektórych problemach, np. przy szukaniu minimalnego drzewa rozpinającego[5].

3.2.2. Sposób wykorzystania

W kontekście gry 'Love Letter', implementacja algorytmu zachłannego wymaga pewnego doprecyzowania. Najważniejszą częścią jest funkcja oceny zagrania, która w niektórych przypadkach musi być oparta o probabilistykę wystąpienia kart u przeciwnika.

Rozważmy scenariusz, w którym algorytm musi podjąć decyzję o zagraniu karty Strażniczki, lub karty Barona. Jest to pierwszy ruch gracza, a w widocznych kartach odrzuconych na starcie są odrzucone karty Króla, Księcia i Pokojówki. Oznacza to, że w talii pozostało 9 kart, a jedna z odrzuconych jest niewidoczna, niemniej jednak też ją trzeba brać pod uwagę. Wyliczenie prawdopodobieństwa P jaką kartę ma przeciwnik jest tym momencie proste, jednak trzeba jeszcze wziąć pod uwagę drobny szczegół - czy do liczenia P wliczać karty posiadane w ręce. Z jednej strony wydaje się to nielogiczne i może prowadzić do wybierania nieoptymalnych decyzji (co przeczyłoby idei algorytmu zachłannego), z drugiej strony można to potraktować jako element blefu, który jest nieodłączną częścią każdej gry towarzyskiej. Z tego powodu zaimplementowałem dwie wersje, nazwane odpowiednio klasyczną i blefującą.

Przyjmując do obliczeń wersję klasyczną, prawdopodobieństwo wystąpienia kart u przeciwnika rozkłada się następująco:

Tutaj będzie tabelka zamiast na następnej stronie.

Tablica 3.1. Przykład rozkładu prawdopodobieństwa

Karta	$P_{(karta)}$
Strażniczka	30%
Kapłan	20%
Baron	10%
Strażniczka	10%
Książę	10%
Hrabina	10%
Księżniczka	10%

Zagranie karty Baron oznacza porównanie drugiej karty z kartą przeciwnika. Łatwo policzyć, że w 70% przypadków skończyłoby to się porażką, a w 30% remisem. Druga dostępna decyzja to zagranie karty Strażniczki, a w jej przypadku najlepszym wyborem jest wytypowanie Kapłana, co daje 20% szans na zwycięstwo i 80% szans na remis. Zauważmy, że gdybyśmy wliczali posiadane karty do obliczenia prawdopodobieństwa, wystąpienie Barona i Kapłana byłoby tak samo możliwe. W takich przypadkach algorytm powinien celować zawsze w kartę z wyższym numerem. By formalnie stwierdzić która decyzja jest lepsza, musimy wybrać przyjmijmy, że funkcja oceny wygląda następująco:

$$F_{(karta)} = 1 + prawdopodobienstwo\_wygranej - prawdopodobienstwo\_przegranej$$

Zatem $Max(F_{(Strazniczka)}, F_{(Baron)}) = 1.2$. Najlepszą decyzją w tym wypadku jest zagranie strażniczki. Jak jednak na podstawie powyższego wzoru określić działanie Kapłana, Pokojówki lub Króla? Każda z tych kart wymaga unikalnej funkcji oceny. Mając na uwadze, że decyzja zawsze powinna być optymalna w ujęciu danego momentu, ustaliłem następujące warunki oceny zagrania karty:

– Strażniczka - zagranie premiowane gdy pozwala wyeliminować przeciwnika.

– Kapłan - jest to karta neutralna, jednak jej przydatność wzrasta jeśli nasza druga karta to Strazniczka lub Ksiaze.

– Baron - premiowane gdy mamy drugą kartę silniejszą niż może mieć przeciwnik

– Pokojówka - im silniejszą posiadamy drugą kartę, tym bardzo warto ją zagrać

– Książę - ocena wzrasta z prawdopodobieństwem Księżniczki u przeciwnika, lub gdy druga karta ma mniejszą siłę niż może mieć przeciwnik (wtedy zagrywana na siebie)

– Król - premiowane gdy przeciwnik ma Księżniczkę lub Hrabinę

– Hrabina - karta neutralna, jednak musimy ją zagrać w określonych sytuacjach

– Księżniczka - nie powinna być nigdy wyrzucona.

3.2.3. Zapis formalny

Strategia zachłanna:
Wejście - (obecny stan gry, karta dociągnięta)
Wyjście - podjęta decyzja

1. Stwórz tablicę prawdopodobieństwa.
2. Dla obu posiadanych kart oblicz $F_{(Karta)}$.
3. Zwróć decyzję o najwyższej ocenie. Jeśli obie karty mają taką samą ocenę, zagraj tę o niższym numerze.

Funkcja oceny:
Skróty: DK - druga karta, $S_{(n)}$ - siła karty

- $F_{(Strazniczka)} = 1 + Max_{(P_{(Zbior\_kart - \{Strazniczka\})})}$
- $F_{(Kaplan)} = 1 + 0.25 * (DK == Ksiaze \vee DK == Strazniczka)$
- $F_{(Baron)} = 1 + \sum P_{Zbior\_kart\_o\_mniejszej\_sile} - \sum P_{Zbior\_kart\_o\_wiekszej\_sile}$
- $F_{(Pokojowka)} = 1 + 0.25 * (S_{(DK)} - S_{(Pokojowka)})$
- $F_{(Ksiaze)} = 1 + P_{(Ksiezniczka)} + 0.1 * (S_{DK} < 5)$
- $F_{(Krol)} = 1 + P_{(\{Ksiezniczka, Hrabina\})}$
- $F_{(Hrabina)} = \begin{cases} 10 & \text{if } DK = Ksiaze; \\ 10 & \text{if } DK = Krol; \\ 1 & \text{w innym wypadku.} \end{cases}$
- $F_{(Ksiezniczka)} = 0$

3.3. Algorytm Monte Carlo Tree Search

4. Implementacja

4.1. Analiza wymagań

4.2. Koncepcja wykonania

4.3. Wykorzystane technologie

4.4. Diagramy

4.5. Prezentacja systemu

4.6. Problemy napotkane w trakcie realizacji

5. Rezultaty

5.1. Czas działania na 1000 partii

5.2. Statystyki zwycięstw

5.3. Zbieżność algorytmów podejmowania decyzji

5.4. Wnioski

6. Podsumowanie

Bibliografia

[1] Alderac Entertainment Group, *Love Letter*, 2012.

[2] Alderac Entertainment Group, *Love Letter*, http://www.alderac.com/tempest/love-letter, 2016-02-08.

[3] Alicja Cewe, Halina Nahorska, Irena Pancer, *Tablice matematyczne*, Wydawnictwo Podkowa Gdańsk 2002, rozdział *Kombinatoryka*.

[4] Alicja Cewe, Halina Nahorska, Irena Pancer, *Wikipedia*, hasło *Strategia w teorii gier*. https://pl.wikipedia.org/wiki/Strategia_mieszana, 2016-02-21.

[5] Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, *Algorytmy*, Wydawnictwo Naukowe PWN, Warszawa 2012, rozdział *Algorytmy Zachłanne*, s. 133.