

Optymalizacja struktury sieci drogowej

Michał Siatkowski

Promotor: dr hab. inż. Aneta Poniszewska - Marańda

Kopromotor: mgr inż. Łukasz Chomątek

Politechnika Łódzka

Łódź, FTIMS, Informatyka 2014/2015

Problematyka i zakres pracy

Niniejsza praca obejmuje zagadnienia z zakresu inżynierii oprogramowania i sztucznej inteligencji. Głównym jej celem jest stworzenie aplikacji optymalizującej strukturę sieci drogowej.

Cele pracy

Celami pracy są:

- 1 Zdefiniowanie problematyki optymalizacji struktury sieci drogowej.
- 2 Stworzenie aplikacji optymalizującej tę strukturę.
- 3 Analiza i ocena efektywności zastosowanych rozwiązań.

Metoda badawcza

Prototypowanie

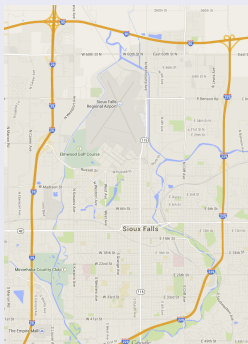
jest to proces budowy modelu matematycznego i obserwacja czy jego zachowanie może pomóc inżynierom w odkryciu ukrytych wad ich projektu. Z założenia prototypy nie wchodzą w skład ostatecznego systemu.

Przegląd literatury w dziedzinie

- ① Wataru Nanya, Hiroshi Kitada, Azusa Hara, Yukiko Wakita, Tatsuhiro Tamaki, and Eisuke Kita
Road Network Optimization for Increasing Traffic Flow.
International Conference on Simulation Technology, JSST
2013.

Podstawowe definicje

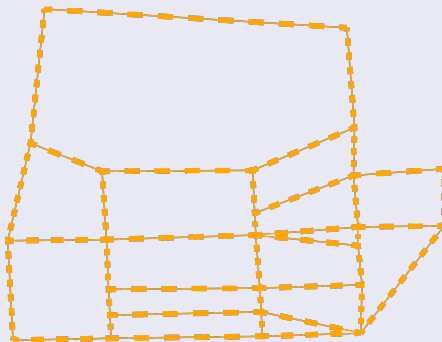
Sieć drogowa



Rysunek 1 : Fragment sieci drogowej w Sioux Falls, Południowa Dakota.

Podstawowe definicje

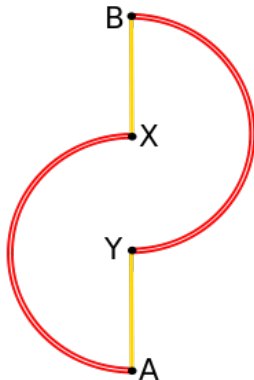
Sieć drogowa w postaci grafu



Rysunek 2 : Sieć drogowa miasta Sioux Falls w postaci grafu.

Michał Siatkowski

Paradoks Braessa[13]



Rysunek 4 : Wyjściowy układ drogowy

Autostrady:

$$AX, t_{AX}(p) = 50 + p \text{ min}$$

$$BY, t_{YB}(p) = 50 + p \text{ min}$$

Drogi lokalne:

$$BX, t_{XB}(p) = 10p \text{ min}$$

$$AY, t_{AY}(p) = 10p \text{ min}$$

Aut jest 6000 i wszystkie mają za zadanie przejechać trasę z A do B.

Równowaga Nasha[13]

Równowaga Nasha to taka sytuacja, w której każdy z samochodów spowoduje wydłużenie swojego czasu jazdy, zmieniając decyzję co do wyboru trasy przy niezmienionych decyzjach pozostałych aut.

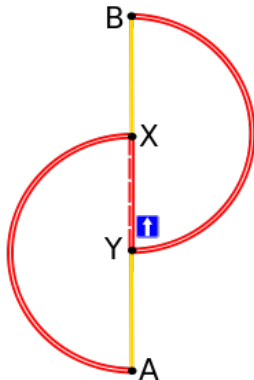
Jeśli p i q to liczby aut w tysiącach pokonujących odpowiednio trasy AXB i AYB, otrzymujemy równania:

$$\begin{aligned}p + q &= 6 \\t_{AX}(p) + t_{BX}(p) &= t_{AY}(q) + t_{BY}(q) \\50 + p + 10p &= 10q + 50 + q\end{aligned}$$

rozwiązaniem jest $p = q = 3$.

Przy tej gęstości ruchu pokonanie obu dostępnych tras zabiera $50 + 3 + 30 = 83$ minuty.

Uzupełniony układ drogowy [13]



Do wyjściowego układu drogowego dodana zostaje autostrada:

$$YX, t_{YX}(p) = 10 + p \text{ min}$$

Aut jest nadal 6000 i wszystkie mają za zadanie przejechać trasę z A do B.

Rysunek 5 : Uzupełniony układ drogowy

Równowaga Nasha dla uzupełnionego układu[13]

Jeśli p , q i r to liczby aut w tysiącach pokonujących odpowiednio trasy AXB, AYB i AYXB, otrzymujemy równania:

$$\begin{aligned} p + q + r &= 6 \\ t_{AX}(p) + t_{BX}(p + r) &= t_{AY}(q + r) + t_{BY}(q) = \\ &= t_{AY}(q + r) + t_{YX}(r) + t_{XB}(p + r) \end{aligned}$$

$$50 + p + 10(p + r) = 10(q + r) + 50 + q = 10(q + r) + 10 + r + 10(p + r)$$

rozwiązaniem jest $p = q = r = 2$.

Czas przejazdu każdej z tych dróg wynosi wówczas

$$50 + 2 + 10(2 + 2) = 92 \text{ minuty.}$$

Słabe punkty istniejących rozwiązań

Paradoks Braessa został sformułowany w roku 1970, a od roku 1996 zaczęły pojawiać się prace negujące lub podważające paradoks[2]. Wiele miast jednak brało i bierze pod uwagę paradoks Braessa podczas projektowania swojej przestrzeni:

- Korea, Seul, likwidacja m.in. estakad Cheonggyecheon,
- Niemcy, Stuttgart, likwidacja dróg zbudowanych w latach 60,
- USA, Nowy Jork, czasowe zamknięcie ulicy 42,
- USA, Winnipeg.[14]

Symulator transportu



Rysunek 6 : Logo symulatora transportu MATSim [5]

Przestrzeń poszukiwań

Najlepszego rozwiązania będę poszukiwał wykorzystując algorytm genetyczny.

Technologie i metodologie programistyczne



Rysunek 7 : Logo języka Java[7]



Rysunek 9 : Logo języka Python[9]



Rysunek 8 : Logo aplikacji IDE Eclipse[8]



Rysunek 10 : Logo rozszerzenia PyDev[10]

Analiza wymagań

- Użytkownik może korzystać z konta "tymczasowego" dla niezalogowanej sesji
- Użytkownik może się zarejestrować na portalu
- Użytkownik może się zalogować na portalu
- Użytkownik może komentować treść portalu, oceniać ją, co wpływa na treść wyświetlaną dla danego użytkownika
- Użytkownik może dodawać do znajomych innych użytkowników portalu
- Użytkownik może filtrować treść portalu względem gustów innych użytkowników

Projekt bazy danych

Informacje o użytkownikach są przechowywane w bazie danych

- Tabela użytkowników
- Tabela komentarzy
- Tabela artykułów
- Tabela ocen

Implementacja: punkty kluczowe

Filtrowanie uwspólnione

Każdy użytkownik jest traktowany jako n -wymiarowy wektor pozycji dostępnych na portalu. Każdy element wektora jest oceną danej pozycji, np. w skali 0 (brak oceny) - 10. Dla przeciętnego użytkownika wektor ten będzie wektorem rzadkim, tzn. będzie występować dużo wartości zerowych. Algorytm, generuje rekomendację w oparciu o grupę użytkowników najbardziej podobnych do rozważanego użytkownika porównując miarę podobieństwa przy użyciu wzoru na cosinusową miarę podobieństwa wektorów:^[1]

$$\text{podobiestwo}(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{a_1 b_1 + a_2 b_2 + \dots + a_n b_n}{\sqrt{a_1^2 + a_2^2 + \dots + a_n^2} \cdot \sqrt{b_1^2 + b_2^2 + \dots + b_n^2}}$$

Wdrożenie

Przy pomocy Media Wiki API treść haseł Wikipedii będzie wyświetlana jako główna część strony. Oprócz tego z treści artykułu zostaną wyodrębnione jego kategorie oraz artykuły, do których zawiera bezpośrednie połączenia (linki). Na podstawie tych danych stworzona zostanie siatka powiązań pomiędzy hasłami a oceniając dany artykuł będziemy modyfikować wielkości odpowiadających elementów wektora ocen.

Przewidywane problemy

Zimny start (Cold start)

Problem typowy dla systemów rekomendacyjnych. Przejawia się brakiem treści do analizy dla systemu, a co za tym idzie, niemożność przedstawienia wyników. [1]

Rozwiązanie:

Integracja portalu z Facebookiem, korzystanie z profilu danego użytkownika.

Dyskusja wyników

Celem pracy jest uzyskanie działającej aplikacji sieciowej, dynamicznie pobierającej treść z portalu Wikipedia.org oraz działający system oceny tej treści.

Drugim celem jest zbadanie efektywności systemu rekomendacyjnego przy pomocy pierwiastkowego błędu średniokwadratowego. ^[1]

$$RMSE = \sqrt{\frac{1}{|\tau|} \sum_{u,i \in \tau} (\hat{r}_{ui} - r_{ui})^2}$$

Błąd obliczany jest dla przygotowanych wcześniej, pewnych, zestawów danych.

Perspektywy dalszych badań w dziedzinie

- Wdrożenie hybrydowego systemu rekomendowania treści; uwspólnionego z kontekstowym.
- Zastosowanie innych funkcji matematycznych do wybierania treści podobnych
- Porównanie i ocena wybranych rozwiązań

Bibliografia I



Leslie Arthur Keith Bloy,

An investigation into Braess' paradox, 02/2007



Rric Pas and Shari Principio

Braess' paradox: Some new insights, April 1996



Wataru Nanya, Hiroshi Kitada, Azusa Hara, Yukiko Wakita,
Tatsuhiro Tamaki, and Eisuke Kita

Road Network Optimization for Increasing Traffic Flow
Conference on Simulation Technology, JSST 2013.



Ana L. C. Bazzan and Franziska Klügl

*Reducing the Effects of the Braess Paradox with Information
Manipulation*

Bibliografia II

-  <http://matsim.org>
-  <http://jgap.sourceforge.net>
-  <http://www.java.com/pl/>
-  <https://eclipse.org>
-  <http://pl.python.org>
-  <http://pydev.org>

Bibliografia III

-  M. Rieser, C. Dobler, T. Dubernet, D. Grether, A. Horni, G. Lammel, R. Waraich, M. Zilske, Kay W. Axhausen, Kai Nagel
MATSim User Guide
updated September 12, 2014
-  A. Chakirov
Enriched Sioux Falls Scenario with Dynamic Demand
MATSim User Meeting, Zurich/Singapore, June 2013.
-  http://pl.wikipedia.org/wiki/Paradoks_Braessa
-  <http://urbnews.pl/paradoks-braessa/>