

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI



Systemy inteligentne

Sprawozdanie z projektu

AUTORZY

Przemysław Barcicki, 260324

Dominik Lenio, 260274

kierunek: **Inżynieria systemów**

30 maja 2023

Streszczenie

W ramach projektu przeprowadzono porównanie skuteczności algorytmów min-max oraz min-max z żalem w celu znalezienia najkrótszej ścieżki między dwoma wierzchołkami w grafie nieskierowanym, gdzie długość krawędzi podana jest jako możliwy zakres w postaci rozkładu równomiernego. Algorytm podczas szukania najlepszej ścieżki ma wgląd tylko w informację o minimum i maksimum z rozkładu.

1 Wstęp – opis problemu

Optymalizacja jest aktualnie dziedziną na której polega cały świat. Pozwala ona na przykładowo minimalizację kosztów prowadzenia działalności poprzez lepszy dobór dostawców czy też zmniejszanie czasu potrzebnego na wykonanie skomplikowanych procesów poprzez ich lepsze kolejkovanie. Znalezienie dobrego sposobu na optymalizację danego procesu, w szczególności w zmiennych warunkach, może mówić o istnieniu danego podmiotu na rynku.

Optymalizacja procesów w przypadku gdzie dany proces ma z góry określone zmienne nie jest żadnym problemem. Istnieje bardzo dużo różnych algorytmów dla różnych problemów które bardzo dobrze sobie radzą z optymalizacją takich zadań. Problem pojawia się jednak gdy w optymalizacji mamy do czynienia ze zmienną, która może przyjąć pewien zakres wartości. Gotowe algorytmy mogą radzić sobie z szukaniem takich rozwiązań, ale w żaden sposób nie korzystamy z informacji o szerokości rozkładu, co może przekładać się na gorsze wyniki.

Celem sprawozdania jest przeprowadzenie analizy oraz opisanie wyników działania algorytmów min-max oraz min-max regret dla problemu szukania ścieżki pomiędzy dwoma wierzchołkami w nieskierowanym grafie, gdzie długości krawędzi dane są w postaci możliwych zakresów dodatnich wartości przedstawiających rozkład równomierny.

2 Definicja problemu

Znając aktualny scenariusz $s \in S$ określony za pomocą informacji o nieskierowanym grafie wejściowym $G = (V, E)$ oraz zakresach odległości

$$s = E \rightarrow \{e \in (o^-, o^+) : 0 < o^- \leq o^+ < \infty\},$$

szukamy takiego rozwiązania $x = (e_1, e_2, \dots, e_n)$, $x \in X$ które będzie minimalizować długość trasy od źródła do celu, v_{start} do v_{koniec} , którą możemy zdefiniować jako:

$$\min_{x \in X} \sum_{i=1}^n s_{x_i} \quad (1)$$

Korzystając z tych definicji możemy określić dwa różne podejścia optymalizacyjne, jedno klasyczne korzystające z samej informacji o aktualnej trasie oraz drugie, korzystające z funkcji żalu do określenia najlepszej trasy.

2.1 Min-max

Dla danego rozwiązania $x \in X$ i danego scenariusza $s \in S$ możemy określić pewną funkcję $val(x, s)$ oceniającą jak dobre jest dane rozwiązanie. Funkcja ta będzie sumą pewnej długości opisującej daną

ścieżkę. Dla danej trasy możemy rozpatrywać trzy różne wartości, są to: wartość oczekiwana danej trasy, teoretycznie minimalna i teoretycznie maksymalna długość trasy.

$$val^{\sim}(x, s) = \sum_{e \in x} \frac{1}{2} (s_e^+ + s_e^-), \quad (2)$$

$$val^+(x, s) = \sum_{e \in x} s_e^+, \quad (3)$$

$$val^-(x, s) = \sum_{e \in x} s_e^- \quad (4)$$

$$\min_{x \in X} \max_{s \in S} val(x, s) \quad (5)$$

2.2 Min-max regret

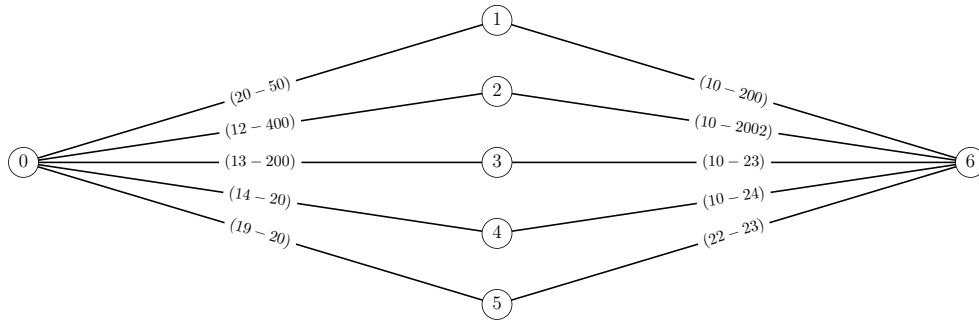
Metoda min-max z żalem jest bardzo podobna do tej metody bez. Różnica to wprowadzenie pewnego dodatkowego elementu określanego jako żal val_s^* , który zamienia nam wartość liczonej funkcji z samej odległości między wierzchołkami na odchylenie od najlepszego możliwego rozwiązania dla danego scenariusza, definiowanej jako optymalne rozwiązanie w scenariuszu s .

Bardzo często zdarza się sytuacja, że samo określenie teoretycznie najlepszej ścieżki jest bardzo trudne, więc w praktyce często korzysta się w tym miejscu z pewnego oszacowania, najczęściej z relaksacji. W przypadku małych grafów $|E| \ll 50$ w celu znalezienia najlepszej wartości można wykonać przegląd zupełny.

$$\min_{x \in X} \max_{s \in S} val(x, s) - val_s^* \quad (6)$$

2.3 Przykład

Rozpatrzmy prosty przykład aby zobrazować różnice między każdym z podejść. Do tego celu skorzystajmy z prostego grafu znajdującego się na rysunku (1), gdzie na każdej z krawędzi zaznaczono możliwe rozkłady.



Rysunek 1: Przykładowy graf gdzie wierzchołek 0 to v_{start} , a wierzchołek 6 to v_{koniec}

Podejście bez żalu, w zależności od rozpatrywanej metody w przypadku tego grafu będzie zachowywało się różnie. Ścieżki wygenerowane przez każdą z metod znajdują się w tabeli (1)

Tabela 1: Tabela zawierająca ścieżki, wartości funkcji $val(x, s)$, minimum oraz maksimum dla każdej z metod

f	Ścieżka	$f(x)$	min	max
val^{\sim}	(0, 4, 6)	34	24	44
val^{-}	(0, 2, 6)	22	22	2402
val^{+}	(0, 5, 6)	43	41	43
val_s^{*}	(0, 4, 6)	14	24	44

Nawet na tak prostym przykładzie widać pewne różnice, każdego z tych podejść. Szukanie po wartości minimalnej, może wpasć w takie rozwiązanie, w którym może i minimalna wartość ścieżki jest bardzo niska, ale wartość w najgorszym wypadku może mieć bardzo słabe konsekwencje. Tak samo dla przypadku z szukaniem po najgorszym możliwym przypadku. Przyjmowanie każdego rozwiązania, pod warunkiem że wartość maksymalna lekko spada może sprawić, że wartość minimalna może być bardzo wysoka co również może skutkować słabym rzeczywistym wynikiem.

W tym przypadku, optymalizowanie po wartości oczekiwanej ścieżki jak i po wartości pochodzącej z funkcji żalu daje takie same wyniki, widać pewną różnicę w ich zachowaniu po samej wartości funkcji, co może nam sugerować, że dla większych przypadków rozwiązania na które te podejścia wskazują, mogą być zgoła inne.

3 Badania

Analizę przeprowadzono na dwóch obszarach, w pierwszym z nich skorzystano z przeglądu zupełnego dla małych grafów $|E| < 40$, w drugim $|E| < 1000$. W drugim pojeździe skorzystano z pewnej heurystyki, w której przeglądano tylko część wszystkich możliwych rozwiązań w celu optymalizacji, które zostały wcześniej wybrane jako quasioptymalne. W dużych grafach tylko takie podejście jest możliwe, ze względu na wysoką złożoność czasową algorytmów potrzebnych do policzenia każdej możliwej ścieżki w każdym możliwym scenariuszu.

Każdy wygenerowany graf stworzony jest na zasadzie pierścienia, gdzie każdy wierzchołek połączony jest z K sąsiadami po jego "prawej" i "lewej" stronie ($|E| = K \times |V|$), a początek i koniec trasy, znajdują się po wzajemnej przekątnej tak aby wszystkie najkrótsze trasy w ilościach krawędzi były sobie równe, ale zawsze była ewentualna możliwość przeskoku do następnego wierzchołka inną trasą. Wygenerowane długości pomiędzy każdym z wierzchołków pochodzą z rozkładu równomiernego $((o^-, o^+); 0 < o^- < o^+ < P)$ i dla każdego nowego grafu są generowane od nowa. Każdy z grafów można opisać tymi trzema wartościami $G \leftarrow (K, |V|, P)$.

3.1 Przegląd zupełny

W tabeli (2) pokazano wyniki przeglądu zupełnego dla grafów z 10, 12, 14, 16, 18 i 20 wierzchołkami dla grafów $(2, x, 1)$.

Tabela 2: Tabela zawierająca minimum oraz maksimum dla każdej z metod w danym grafie.

$G \rightarrow (2, 10, 1)$			$G \rightarrow (2, 12, 1)$			$G \rightarrow (2, 14, 1)$		
f	min	max	f	min	max	f	min	max
val_s^*	0.43	2.33	val_s^*	1.11	2.74	val_s^*	0.74	3.26
val^-	0.43	2.33	val^-	1.11	2.74	val^-	0.74	3.26
val^+	0.6	1.66	val^+	1.73	2.18	val^+	1.5	2.52
val^\sim	0.6	1.66	val^\sim	1.39	2.39	val^\sim	0.8	3.12
$G \rightarrow (2, 16, 1)$			$G \rightarrow (2, 18, 1)$			$G \rightarrow (2, 20, 1)$		
f	min	max	f	min	max	f	min	max
val_s^*	0.66	3.18	val_s^*	1.34	2.57	val_s^*	0.86	2.3
val^-	0.66	3.18	val^-	1.34	2.57	val^-	0.65	3.35
val^+	1.96	2.72	val^+	1.34	2.57	val^+	0.86	2.3
val^\sim	0.66	3.18	val^\sim	1.34	2.57	val^\sim	0.86	2.3

Zauważyć można, że w przypadku małych grafów rzadkich, gdzie jest mało najkrótszych ścieżek i nie jest wysoka różnica między najlepszym a najgorszym scenariuszem, optymalizacja za pomocą funkcji żalu daje takie same wyniki jak inne algorytmy.

3.2 Heurestyka

W metodzie heurystycznej wykorzystano dwa uproszczenia, które zdecydowanie skracają przeszukiwanie grafu, ale mogą nie znajdować optymalnego rozwiązania, lecz inne do niego zbliżone. Oba dotyczą się ograniczenia obszaru przeszukiwania i polegają na wcześniejszym wygenerowaniu mniejszej ilości scenariuszy do przeszukiwania i na ich podstawie wygenerowanie najkrótszych ścieżek, będzie się to objawiać szukaniem. W tabeli (3) zamieszczono wyniki przeszukiwania grafów z 50, 70, 90, 110, 130 i 150 wierzchołkami w kombinacji $(5, x, 5)$

Tabela 3: Tabela zawierająca minimum oraz maksimum dla każdej z metod w danym grafie.

$G \rightarrow (5, 50, 5)$			$G \rightarrow (5, 70, 5)$			$G \rightarrow (5, 90, 5)$		
f	min	max	f	min	max	f	min	max
val_s^*	6.26	11.6	val_s^*	5.46	14.24	val_s^*	6.41	20.45
val^-	2.64	15.46	val^-	3.73	18.56	val^-	4.51	22.51
val^+	6.26	11.6	val^+	6.83	13.2	val^+	5.25	19.61
val^\sim	4.08	13.59	val^\sim	4.56	14.83	val^\sim	5.25	19.61
$G \rightarrow (5, 110, 5)$			$G \rightarrow (5, 130, 5)$			$G \rightarrow (5, 150, 5)$		
f	min	max	f	min	max	f	min	max
val_s^*	8.03	27.52	val_s^*	9.81	35.27	val_s^*	11.78	36.32
val^-	4.68	41.38	val^-	6.23	66.08	val^-	7.54	67.79
val^+	13.4	24.31	val^+	12.32	29.32	val^+	13.77	31.9
val^\sim	8.92	26.24	val^\sim	11.0	30.52	val^\sim	10.52	33.3

4 Wnioski

Optymalizowanie za pomocą funkcji żalu pozwala nam minimalizować ewentualne złe skutki wybrania jednej metody względem drugiej, może być to potrzebne w przypadkach gdzie równie ważne są najgorsze i najlepsze możliwości tak jak np. w inwestowaniu czy bankowości gdzie każda możliwość jest niewiadomą. Zdarza się, że w pewnych warunkach znalezione rozwiązanie nie jest takie jakiego można się spodziewać i inne metody dorównują jakością, lecz nie mamy pewności w żadnej innej, że rozwiązanie które znajdziemy nie bierze pod uwagę różnicy między najlepszym a najgorszym możliwym scenariuszem.

A Kod źródłowy

Kody źródłowe umieszczone zostały w repozytorium GitHub:
<https://github.com/mlodybercik/nondeterministic-regret-path>.

Literatura

- [1] Hassene Aissi, Cristina Bazgan, Daniel Vanderpooten. Min-max and min-max regret versions of some combinatorial optimization problems: a survey. (2007, fhal-00158652)
- [2] Refael Hassin, Ariel Keinan Greedy Heuristics with Regret, with Application to the Cheapest Insertion Algorithm for the TSP