



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

specjalność: Inżynieria oprogramowania

Praca dyplomowa - magisterska

OCENA EFEKTYWNOŚCI ALGORYTMÓW

auth

słowa kluczowe:
efektywność algorytmu
genetyczny

krótkie streszczenie:

Bardzo krótkie streszczenie w którym powinno się znaleźć omówienie tematu pracy i poruszanych terminów. Tekst ten nie może być zbyt długi.

Promotor:
	<i>imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Wrocław 2010

Niniejszy dokument został złożony w systemie L^AT_EX.

Contents

Wstęp	1
Rozdział 1 Inżynieria transportowa	3
Rozdział 3 Algorytmy klasyczne	5
3.1 Algorytm Dijkstry	5
3.2 Różne klasy złożoności algorytmu Dijkstry	6
3.3 Implementacja algorytmu Dijkstry	6
3.4 Dyskusja wyników	6
Rozdział 4 Algorytm klasyczny z heurystyką	7
4.1 Algorytm A	7
4.2 Heurystyka Manhattan	7
4.3 Implementacja algorytmu Astar	7
4.4 Dyskusja wyników	7
Appendix A. Coś dodatkowego	9
List of Figures	11
List of Tables	13
Bibliography	15

Abstrakt

Abstract

Abstrakt

Abstract

Wstep

Rozdział 1 Inżynieria transportowa

Rozdział 3 Algorytmy klasyczne

3.1 Algorytm Dijkstry

Jak wcześniej wspomniano algorytm Dijkstry służy przede wszystkim do rozwiązywania problemu najkrótszych ścieżek z jednym źródłem w ważonym grafie skierowanym $G = (V, E)$ w przypadku, gdy wagi wszystkich krawędzi są nieujemne¹. Należy więc założyć za [1], że

$$w(u, v) \geq 0$$

dla każdej krawędzi

$$(u, v) \in E$$

Zasada działania algorytmu Dijkstry jest dość prosta, można ją krótko opisać poprzez zachłanne rozprzestrzenianie się zbioru wierzchołków już przejrzanych na te, które jeszcze nie zostały rozpatrzone, przy czym o kolejności rozpatrywania decyduje waga wyrażona przez odległość od punktu startowego.

W algorytmie Dijkstry jest pamiętany zbiór S zawierający te wierzchołki, dla których wagi najkrótszych ścieżek ze źródła s zostały już obliczone. To znaczy, że dla każdego wierzchołka $c \in S$ mamy $d[c] = \beta(s, c)$. Algorytm Dijkstry polega na wielokrotnym powtarzaniu następujących operacji: wybrania wierzchołka $u \in V - S$ o najmniejszym oszacowaniu wagi najkrótszej ścieżki, dodania wierzchołka u do S , wykonania relaksacji krawędzi opuszczających wierzchołek u . [1]

W algorytmie Dijkstry jest pamiętany zbiór S zawierający te wierzchołki, dla których wagi najkrótszych ścieżek ze źródła s zostały już obliczone. To znaczy, że dla każdego wierzchołka $c \in S$ mamy $d[c] = \beta(s, c)$. Algorytm Dijkstry polega na wielokrotnym powtarzaniu następujących operacji: wybrania wierzchołka $u \in V - S$ o najmniejszym oszacowaniu wagi najkrótszej ścieżki, dodania wierzchołka u do S , wykonania relaksacji krawędzi opuszczających wierzchołek u . [1]

W algorytmie Dijkstry jest pamiętany zbiór S zawierający te wierzchołki, dla których wagi najkrótszych ścieżek ze źródła s zostały już obliczone. To znaczy, że dla każdego wierzchołka $c \in S$ mamy $d[c] = \beta(s, c)$. Algorytm Dijkstry polega na wielokrotnym

1. W odróżnieniu od algorytmu Bellmana-Forda gdzie graf wejściowy może zawierać ujemne wagi krawędzi.

```

DIJKSTRA( $G, w, s$ )
1  INITIALIZE-SINGLE-SOURCE( $G, s$ )
2   $S \leftarrow \emptyset$ 
3   $Q \leftarrow V[G]$ 
4  while  $Q \neq \emptyset$ 
5      do  $u \leftarrow \text{EXTRACT-MIN}(Q)$ 
6           $S \leftarrow S \cup \{u\}$ 
7          for każdy wierzchołek  $v \in \text{Adj}[u]$ 
8              do RELAX( $u, v, w$ )

```

Figure 0.1. Algorytm Dijkstry

```

INITIALIZE-SINGLE-SOURCE( $G, s$ )
1  for każdy wierzchołek  $v \in V[G]$ 
2      do  $d[v] \leftarrow \infty$ 
3           $\pi[v] \leftarrow \text{NIL}$ 
4   $d[s] \leftarrow 0$ 

```

Figure 0.2. A picture of a tucan

powtarzaniu następujących operacji: wybrania wierzchołka $u \in V - S$ o najmniejszym oszacowaniu wagi najkrótszej ścieżki, dodania wierzchołka u do S , wykonania relaksacji krawędzi opuszczających wierzchołek u . [1]

W algorytmie Dijkstry jest pamiętany zbiór S zawierający te wierzchołki, dla których wagi najkrótszych ścieżek ze źródła s zostały już obliczone. To znaczy, że dla każdego wierzchołka $c \in S$ mamy $d[v] = \beta(s, v)$. Algorytm Dijkstry polega na wielokrotnym powtarzaniu następujących operacji: wybrania wierzchołka $u \in V - S$ o najmniejszym oszacowaniu wagi najkrótszej ścieżki, dodania wierzchołka u do S , wykonania relaksacji krawędzi opuszczających wierzchołek u . [1]

3.2 Różne klasy złożoności algorytmu Dijkstry

3.3 Implementacja algorytmu Dijkstry

3.4 Dyskusja wyników

Rozdział 4 Algorytm klasyczny z heurystyką

4.1 Algorytm A

4.2 Heurystyka Manhattan

4.3 Implementacja algorytmu Astar

4.4 Dyskusja wyników

Appendix A

Coś dodatkowego

List of Figures

0.1	Algorytm Dijkstry	6
0.2	A picture of a tucan	6

List of Tables

Bibliography

- [1] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. *Wprowadzenie do algorytmów, Wydanie szóste zmienione i rozszerzone*. Wydawnictwa Naukowo Techniczne, Warszawa, 2004.