

Jak nám heuristiky usnadňují řešení problémů?

aneb Cesta z Prahy do Brna
přes Liberec

- **Yahor Herashchanka** -
yherashchanka@gmail.com,
Gymnázium Turnov
- **Josef Lezna** -
josef.lezna@gymzn.cz,
Gymnázium doktora Karla
Polesného Znojmo
- **Matouš Richter** -
2018-richter-matous@gymtan.c
z, Gymnázium Tanvald

21. 6. 2023

Týden vědy na Jaderce, 2023



Motivace

- **Příklad**

- máme deset místností, které jsou buď špinavé, nebo čisté
- vysavač je buď zapnutý, vypnutý, či spí
- vysavač má pět stupňů výkonnosti
- senzor znečištění on/off
- $k = 10 \cdot 3 \cdot 5 \cdot 2 \cdot 2^{10}$
- v tomto případě máme zhruba 300 000 možných stavů

- heuristiky nám umožní efektivnější prohledávání těchto stavů

Stavový prostor

- *stavový prostor* = množina stavů daného problému
- lze ho formalizovat jako graf
- *cesta* = posloupnost přechodů mezi propojenými stavy
- snažíme se nalézt cestu mezi počátečním a koncovém stavem (problémem a řešením)
- používáme *algoritmy pro hledání cest v grafu*

Hledání v stavovém prostoru & heuristiky

- vyhledávací algoritmy sledují podle určitých kritérií cesty vedoucí z počátečního stavu
- některé cesty povedou ke koncovému stavu s nižšími náklady
- chceme se zabývat primárně jimi → zavedeme si funkci (heuristiku), která bude odhadovat zbývající cenu do cíle

Popis řešení

- zavádíme množinu vrcholů V
- její součástí je “s” a “t” - vrcholy start a cíl
- $V = E \cup F \cup U$
 - E - prozkoumaná políčka
 - F - hranice
 - U - neprozkoumaná políčka

Popis řešení

$E = \text{set} ()$ # množina prozkoumaných vrcholů je prázdná

$F = \{ s \}$ # v hranici leží pouze výchozí vrchol

while (True)

 if (! F) : return None # nemá řešení

 v = vyberZHranice (F)

 if (v == t) : return vytvořReseni(v) # našli jsme cestu

 E = E.add(v) # vrchol je přidán k prozkoumaným

 F = aktualizujHranici(F, v) # doplní sousedy v do hranice, označí u nich,

 # že jsme do nich vstoupili z v a spočítá pro ně cenu

Popis řešení - zjednodušení zadání

- diskretizace mapy - rozdělení na čtverečky
- použití manhattanské vzdálenosti
- převedení reálných hodnot na celočíselné

Výběrové řešení - UCS

- zavádí funkci g - vybírá nejlevnější cestu
- cesty si ukládá → vysoká paměťová složitost
- optimalizuje

Výběrové řešení - GBFS

- funkce h - odhad vzdálenosti do cíle
- = "hladový algoritmus"

Výběrové řešení - DFS

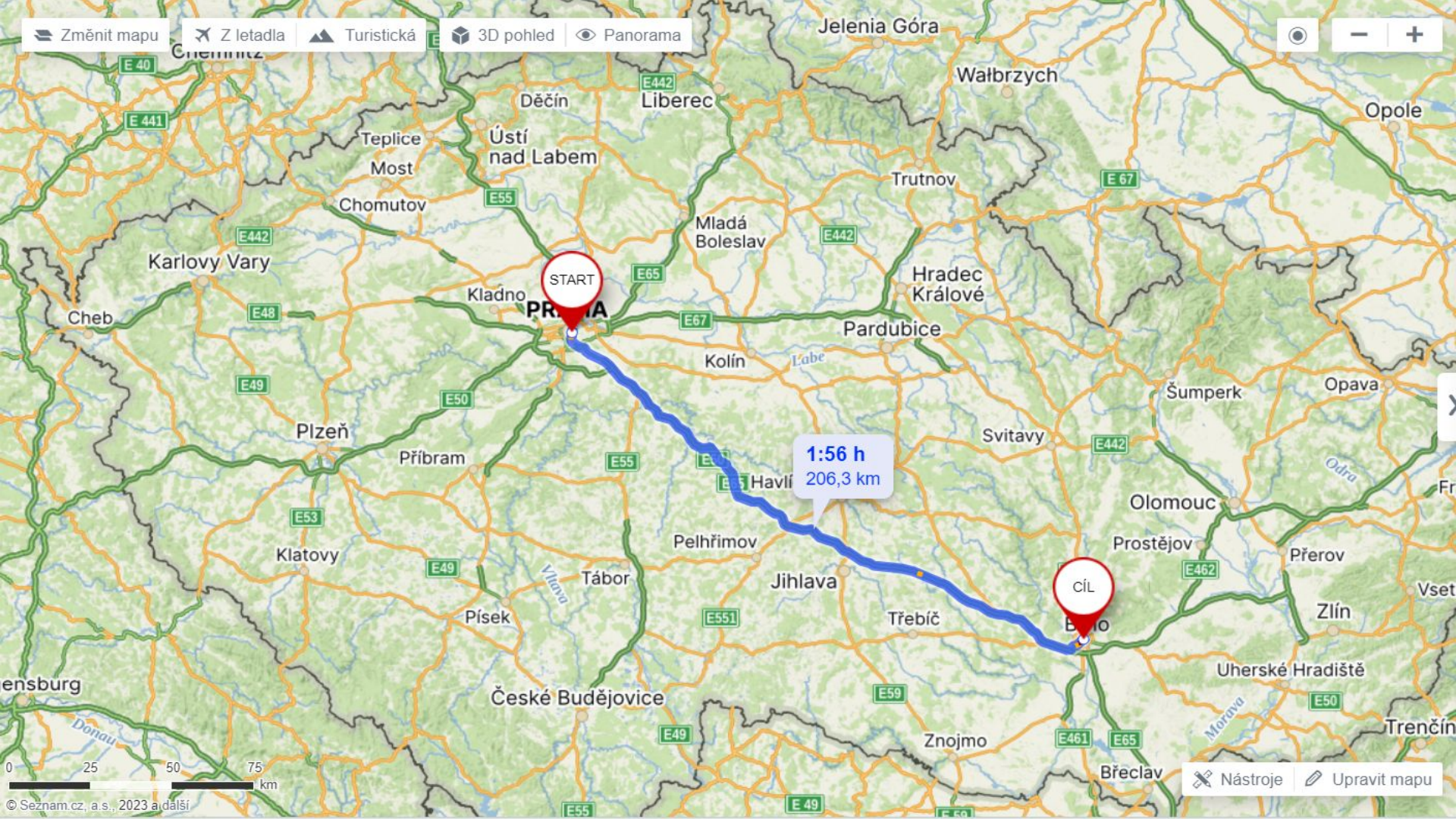
- = prohledávání do hloubky
- vybere jednu cestu, kterou dále prozkoumává, následně se vrací
- neoptimalizuje, nejlépe použitelný v praxi

Diskuze

- ve hrách se vyskytuje terén s odlišnými vlastnostmi (kopec, bažina, led...) → rozdílné hodnoty g
- nepřátelé → také vyšší g , ale pohybují se
- záleží na algoritmu, kudy řešení povede
 - UCS by se snažil horu obejít
 - GBFS by horu přešel
 - A^* zhodnotí, co je výhodnější

Závěr

- heuristiky dokáží řádově zrychlit řešení problémů
- zužují možnost výběru
- při cestě z Prahy do Brna nemusím brát do úvahy Liberec
- urychluje proces
- A* je obecně nejefektivnější algoritmus
- existují případy, kdy není tak rychlý
- v případě chybného návrhu heuristiky nemusí být řešení optimální (ne nutně nemusí být na škodu)



Změnit mapu

Z letadla

Turistická

3D pohled

Panorama



START

PRŮVA

1:56 h

206,3 km

CÍL

0 25 50 75 km

© Seznam.cz, a.s., 2023 a další

Nástroje

Upravit mapu

Poděkování a reference

- Děkujeme našemu garantovi Vladimíru Jarému za vedení našeho miniprojektu.

Reference:

[1] S. RUSSEL; P. NORVIG. *Artificial Intelligence: A Modern Approach*. 2. vyd. New Jersey, USA: Prentice Hall, 2003. ISBN 0-13-790395-2.

[2] M. TUREČEK. *Demonstrace metod prohledávání stavového prostoru*. Brno, 2010. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce František Zbořil.

[3] N. SWIFT. *Easy A* (star) Pathfinding*. [online] 2017 [cit. 20. 6. 2023]. Dostupné z <https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2>