

Assignment 1: Kvalitatívne heuristiky (greedy algoritmy a ich vylepšenia)

*Filip Fylyp
Vadym Mospan
Oleksandr Boyko
Daniil Melnychuk
FEI, INTS, HI_HOP*

18/11/2025

Content

1. Uvod do greedy algoritmov
2. Activity Selection Problem
3. Coin Change Problem
4. Experimenty
5. Vysledky
6. Záver

1. Uvod do greedy algoritmov

Kvalitatívne heuristiky, reprezentované predovšetkým greedy algoritmami (žiadnymi algoritmami), predstavujú efektívne optimalizačné metódy založené na princípe vykonávania lokálne optimálnych rozhodnutí v každom kroku s cieľom nájsť globálne riešenie. Tieto algoritmy sa vyznačujú jednoduchosťou implementácie, rýchlosťou výpočtovou dobu a nízkou pamäťovou náročnosťou, čo ich robí vhodnými pre riešenie rozsiahlych problémov v reálnom čase.

Greedy algoritmy predstavujú základnú techniku riešenia problémov, ktorá funguje tak, že v každom kroku sa rozhoduje pre lokálne optimálnu možnosť s nádejou nájsť globálne optimálne riešenie. Táto stratégia je obzvlášť užitočná v optimalizačných problémoch, ako sú plánovanie, grafové algoritmy a výber intervalov.

1.1 Základné vlastnosti

Greedy algoritmy sa vyznačujú dvoma fundamentálnymi vlastnosťami, ktoré určujú ich správnosť a použiteľnosť:

- *Greedy Choice Property* (vlastnosť žiadostivého výberu): Táto vlastnosť znamená, že problém je taký, že riešenie (globálne optimum) môže byť dosiahnuté vykonaním žiadostivých volieb v každom kroku (lokálne optimálne voľby). Inými slovami, lokálne optimálny výber vedie k globálnemu riešeniu
- *Optimal Substructure* (optimálna podštruktúra): Znamená, že optimálne riešenie problému obsahuje optimálne riešenia subproblémov. Táto vlastnosť umožňuje rozdeliť problém na menšie, súvisiace subproblémy, pričom optimálne riešenie týchto subproblémov prispieva k nájdeniu optimálneho riešenia pôvodného problému.

1.2 Výpočtová zložitosť:

Greedy algoritmy vyhľadávajú tak z hľadiska časovej ($O(n)$ - $O(n \log n)$), ako aj priestorovej ($O(1)$ - $O(n)$) zložitosti v porovnaní so zložitejšími optimalizačnými metódami. Časová zložitosť bežných greedy algoritmov je väčšinou $O(n \log n)$ alebo $O(n)$, čo ich robí ideálnou voľbou pre úlohy, kde je dôležitá rýchlosť a efektívnosť využívania zdrojov.

1.3 Inšpirácia

Inšpirácia greedy algoritmov vychádza z prirodzeného ľudského rozhodovania, kde sa často volí najslubnejšia možnosť v danom okamihu bez analýzy dlhodobých dôsledkov. Tento prístup odráža spôsob, akým ľudia často prijímajú rozhodnutia v reálnom živote - výberom možnosti, ktorá sa v danej chvíli javí ako najlepšia.

Je dôležité poznamenať, že podobne ako v ľudskom rozhodovaní, aj greedy algoritmy nemusia vždy viesť k absolútne najlepšiemu riešeniu, ale často poskytujú rozumné a efektívne riešenia v praktickom časovom rámci.

1.4 Vylepšené verzie greedy algoritmov

Napriek tomu, že žiadne algoritmy nemusia vždy garantovať absolutne optimálne riešenie, vylepšenia ako randomizácia, iteratívne prístupy alebo kombinácia s metaheuristikami výrazne zvyšujú kvalitu a robustnosť riešení. Hybridizácia metaheuristik je v súčasnosti jedným z najúspešnejších trendov v optimalizácii.

2. Activity Selection Problem

2.1 Definícia problému

Activity Selection Problem (problém výberu aktivít) je klasický optimalizačný problém, kde máme danú množinu n aktivít, pričom každá aktivita i má čas začiatku s_i a čas ukončenia f_i . Cieľom je vybrať maximálny počet navzájom nezávislých (neprekryvajúcich sa) aktivít.

Formálna definícia:

- Vstup: Množina $S = \{a_1, a_2, \dots, a_n\}$ n aktivít
- Každá aktivita a_i má čas začiatku s_i a čas ukončenia f_i (kde $0 \leq s_i < f_i < \infty$)
- Dve aktivity A_i a A_j sú kompatibilné, ak $s_i \geq f_j$ alebo $s_j \geq f_i$
- Výstup: Maximálna množina navzájom kompatibilných aktivít

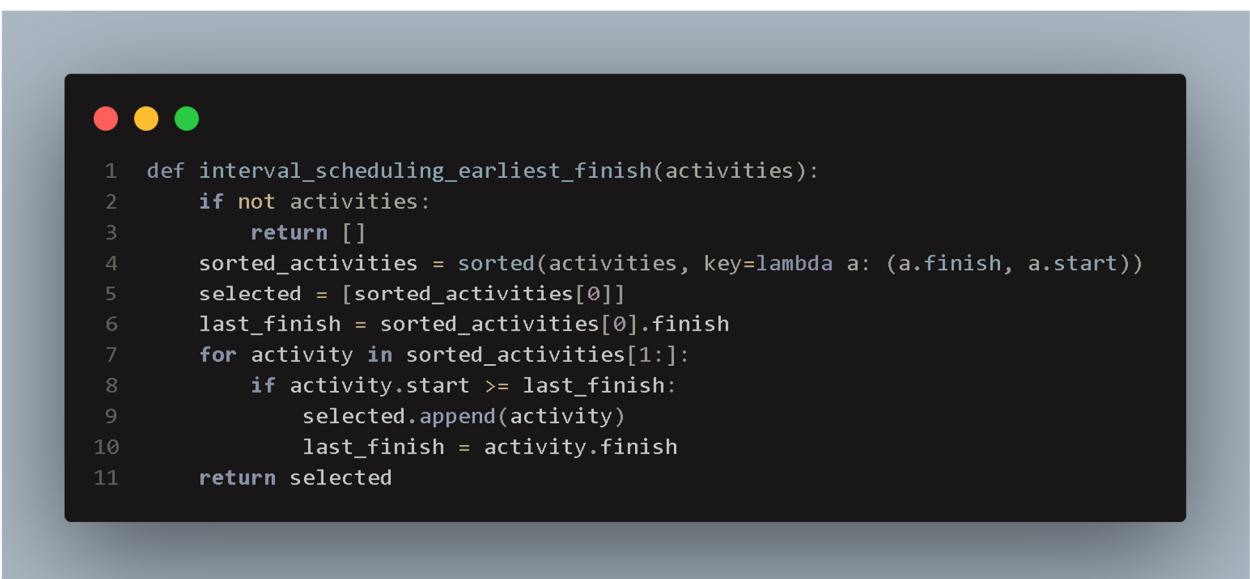
2.2 Greedy stratégia

Greedy algoritmus pre activity selection funguje nasledovne:

1. Zoradiť aktivity podľa času ukončenia f_i v rastúcom poradí
2. Vybrať prvú aktivitu (aktivitu s najskorším časom ukončenia)
3. Pre každú zostávajúcu aktivitu: ak jej čas začiatku je väčší alebo rovný času ukončenia predchádzajúcej vybratej aktivity, vybrať ju

Tento prístup je optimálny, pretože:

- Výberom aktivity, ktorá skončí najskôr, zanecháme čo najviac času pre zostávajúce aktivity
- Problém vykazuje optimálnu podštruktúru
- Platí greedy choice property (lokálne optimálne voľby vedú k globálne optimálnemu riešeniu)



```
● ● ●
1 def interval_scheduling_earliest_finish(activities):
2     if not activities:
3         return []
4     sorted_activities = sorted(activities, key=lambda a: (a.finish, a.start))
5     selected = [sorted_activities[0]]
6     last_finish = sorted_activities[0].finish
7     for activity in sorted_activities[1:]:
8         if activity.start >= last_finish:
9             selected.append(activity)
10            last_finish = activity.finish
11    return selected
```

2.3 Kritérium optimality

Cieľ: Maximalizovať počet vybratých aktivít, pričom žiadne dve aktivity sa nesmú časovo prekrývať.

Vyhodnotenie kvality riešenia:

- Počet vybratých aktivít (čím viac, tým lepšie)
- Časová zložitosť: $O(n \log n)$ kvôli triedeniu; samotný výber je $O(n)$

3. Coin Change Problem

3.1 Definícia problému

Coin Change Problem je klasický optimalizačný problém, kde máme množinu n nominálnych hodnôt mincí a sumu S . Cieľom je nájsť minimálny počet mincí, ktoré dokážu rozmeniť sumu S , pričom každú mincu môžeme použiť ľubovoľne veľakrát.

Formálna definícia:

- Vstup: Množina nominálnych hodnôt mincí $\mathbf{C} = \{c_1, c_2, \dots, c_n\}$ a cieľová suma S
- Každá minca môže byť použitá neobmedzene veľakrát
- Výstup: Minimálny počet mincí k , kde $\sum c_i = S$, alebo informácia, že riešenie neexistuje

3.2 Greedy stratégia

Greedy algoritmus pre Coin Change Problem funguje nasledovne:

1. Zoradiť mince podľa hodnoty zostupne
2. Iteratívne vyberať najväčšiu mincu, ktorá neprevyšuje zostávajúcu sumu
3. Pokračovať, dokým nedosiahneme S alebo zistíme, že riešenie neexistuje

3.4 Kritérium optimality a zložitosť

Vyhodnotenie kvality riešenia:

- Počet použitých mincí (čím menej, tým lepšie)
- Časová zložitosť $O(n)$, Priestorová zložitosť $O(1)$

Dôležité pozorovanie:

- Greedy nie je vždy optimálne

4. Experimenty

4.1 Experimentálny dizajn pre Activity Selection Problem

Cieľ experimentu: Porovnať rôzne greedy stratégie pre Activity Selection Problem z hľadiska kvality riešenia a výpočtového času.

Testované stratégie:

1. Greedy-Finish (EFT): Výber podľa najskoršieho času ukončenia
2. Greedy-Start (EST): Výber podľa najskoršieho času začiatku
3. Shortest-Duration (SDF): Výber podľa najkratšej dĺžky aktivity
4. Randomized: Žiadostivý výber s náhodným prvkom

Parametre experimentu:

- Veľkosti vstupných dát (n): 10, 20, 30, 50, 100 aktivít
- Pre každú veľkosť bolo vygenerovaných niekoľko náhodných inštancií
- Merané metriky:
 - Počet vybratých aktivít (kvalita riešenia)
 - Výpočtový čas v milisekundách (efektivita)
 - Štandardná odchýlka času vykonávania

Metodológia generovania dát:

- Aktivity boli generované náhodne s časmi začiatku a ukončenia
- Zabezpečená rozmanitosť prekrývaní aktivít
- Všetky stratégie testované na rovnakých inštanciach pre objektívne porovnanie

4.2 Experimentálny dizajn pre Coin Change Problem

Cieľ experimentu: Porovnať greedy a dynamic programming prístupy pre Coin Change Problem, s osobitným zameraním na kanonické a nekanonické systémy mincí.

Testované prístupy:

1. Greedy Algorithm: Vždy vyberá najväčšiu možnú mincu
2. Dynamic Programming: Garantuje optimálne riešenie

Merané metriky:

- Počet použitých mincí (kvalita riešenia)
- Zhoda/nesúlad medzi greedy a DP riešením
- Status optimality greedy riešenia

Metodológia:

- Pre kanonické systémy sa očakáva, že greedy = DP
- Pre nekanonické systémy sa očakáva, že greedy > DP (viac mincí)

5. Výsledky

5.1 Výsledky pre Activity Selection Problem

Analýza výsledkov:

1. Kvalita riešenia (počet vybraných aktivít):

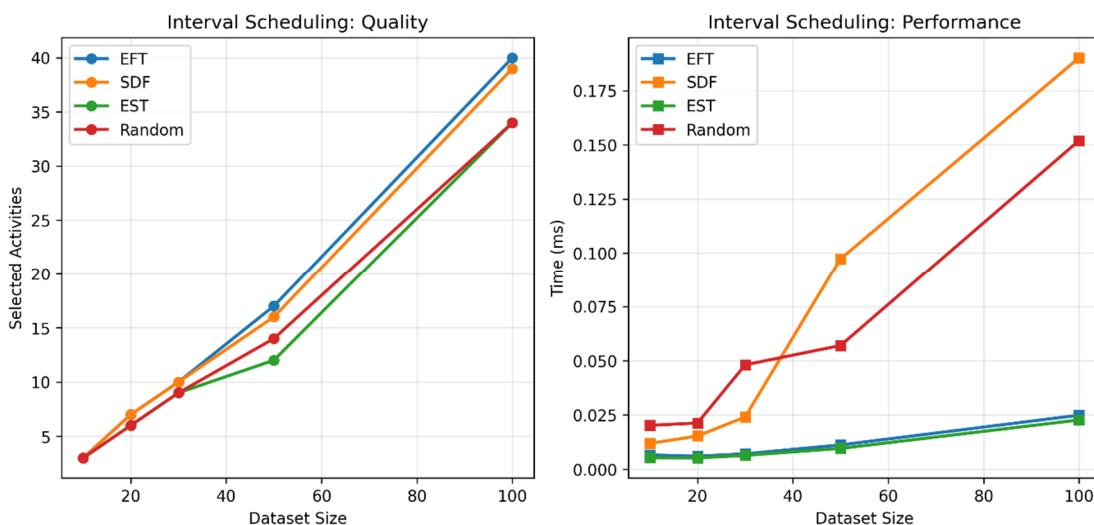
- Earliest Finish Time (EFT) konzistentne dosahuje najvyšší počet vybraných aktivít napriek všetkými veľkosťami datasetu
- Pre $n=100$: EFT vybralo 40 aktivít, čo je najlepší výsledok
- Earliest Start Time (EST) a Randomized stratégia dosahujú suboptimálne výsledky (34 aktivít pre $n=100$)
- Shortest Duration (SDF) dosahuje takmer optimálne výsledky (39 aktivít pre $n=100$)

Experiment Results: Interval Scheduling

n	Greedy-Finish (ms) ±0 / avg count	Greedy-Start (ms) ±0 / avg count	Shortest-Duration (ms ±0) / avg count	Randomized (ms ±0) / avg count
10	4 ± 0.014 / 4	4 ± 0.007 / 4	4 ± 0.023 / 4	4 ± 0.033 / 4
20	7 ± 0.010 / 7	7 ± 0.007 / 7	7 ± 0.031 / 7	4 ± 0.040 / 4
30	14 ± 0.015 / 14	9 ± 0.011 / 9	14 ± 0.057 / 14	10 ± 0.059 / 10
50	17 ± 0.022 / 17	15 ± 0.017 / 15	16 ± 0.102 / 16	16 ± 0.147 / 16
100	40 ± 0.043 / 40	34 ± 0.036 / 34	39 ± 0.367 / 39	34 ± 0.276 / 34

2. Výpočtový výkon (čas vykonávania):

- Všetky stratégie vykazujú lineárny rast času s veľkosťou vstupu (po triedení)
- EFT má najlepší pomer kvality a času: 40 aktivít za 0.043 ms pri $n=100$
- EST je najrýchlejšia, ale s nižšou kvalitou riešenia
- SDF je najpomalšia (0.367 ms pri $n=100$) kvôli nutnosti výpočtu dĺžky aktivít
- Randomized vykazuje najväčšiu variabilitu výpočtového času (najvyššia štandardná odchýlka)



3. Škálovateľnosť:

- Všetky stratégie dobre škálujú s rastúcim počtom aktivít
- Časová zložitosť zodpovedá teoretickému $O(n \log n)$ pre triedenie + $O(n)$ pre výber
- Pre $n=100$ je čas stále v rámci zlomkov sekundy, čo potvrdzuje efektivitu greedy prístupu

Závery pre Activity Selection:

- Earliest Finish Time (EFT) je najlepšia stratégia z hľadiska kvality riešenia aj výpočtového času
- Greedy algoritmus založený na EFT je optimálny a garantuje maximálny počet vybraných aktivít
- Alternatívne stratégie (EST, SDF, Randomized) neposkytujú optimálne riešenia a preto nie sú vhodné pre tento problém

5.2 Výsledky pre Coin Change Problem

Analýza výsledkov:

1. Kanonické systémy (US coins, EU cents, Large, Small, Medium):

- Greedy algoritmus poskytuje optimálne riešenia identické s DP
- Pre US coins systém $\{1, 5, 10, 25\}$ je greedy vždy optimálne
- Pre EU cents systém $\{1, 2, 5, 10, 20, 50\}$ je greedy vždy optimálne
- Tento výsledok potvrdzuje, že takmer všetky reálne menové systémy sú kanonické

2. Nekanonické systémy (6,10 sys, 5,6,9 sys, 7,10 sys, 3,4 sys):

- Greedy algoritmus zlyháva a poskytuje suboptimálne riešenia
- Príklad $\{1, 6, 10\}$ pre sumu 12:
 - Greedy: $10 + 1 + 1 = 3$ mince
 - DP: $6 + 6 = 2$ mince (optimálne)
- Príklad $\{1, 7, 10\}$ pre sumu 14:
 - Greedy: $10 + 1 + 1 + 1 + 1 = 5$ mincí
 - DP: $7 + 7 = 2$ mince (optimálne)
- V niektorých prípadoch greedy používa až $2.5\times$ viac mincí než optimálne riešenie

Experiment Results: Coin Change Problem

Test Case	Coins	Amount	Greedy	DP	Status
US coins	$[1, 5, 10, 25]$	63	6	6	✓ Optimal
EU cents	$[1, 2, 5, 10, 20, 50]$	87	5	5	✓ Optimal
Large	$[1, 5, 10, 25]$	999	45	45	✓ Optimal
6,10 sys	$[1, 6, 10]$	12	3	2	✗ Not optimal
5,6,9 sys	$[1, 5, 6, 9]$	11	3	2	✗ Not optimal
7,10 sys	$[1, 7, 10]$	14	5	2	✗ Not optimal
3,4 sys	$[1, 3, 4]$	6	3	2	✗ Not optimal
Small	$[1, 5, 10]$	1	1	1	✓ Optimal
Medium	$[1, 5, 10, 25, 50]$	100	2	2	✓ Optimal

3. Porovnanie algoritmov:

- Dynamic Programming:
 - Vždy garantuje optimálne riešenie
 - Časová zložitosť $O(n \times S)$ - vyššia než greedy
 - Priestorová zložitosť $O(S)$ - vyššia než greedy
 - Vhodné pre všetky systémy mincí
- Greedy Algorithm:
 - Časová zložitosť $O(n)$ - nižšia než DP
 - Priestorová zložitosť $O(1)$ - nižšia než DP
 - Optimálne len pre kanonické systémy
 - Nie je vhodné pre nekanonické systémy

Závery pre Coin Change:

1. Dynamic Programming je univerzálné riešenie, ktoré vždy garantuje optimálne výsledky
2. Greedy algoritmus je optimálny len pre kanonické systémy mincí (napr. 1, 5, 10, 25)
3. Pre praktické menové systémy je greedy dostatočné a efektívnejšie
4. Pre teoretické alebo špeciálne systémy mincí je nutné použiť DP alebo overiť kanonickosť systému pred použitím greedy

6. Záver

Greedy algoritmy predstavujú fundamentálnu optimalizačnú techniku s výraznými silnými stránkami v oblasti efektivity a jednoduchosti implementácie. Experimenty potvrdili, že greedy prístup je vysoko efektívny pre problémy vykazujúce greedy choice property a optimal substructure, ako je Activity Selection Problem.

Kľúčové poznatky z analýzy:

- Earliest Finish Time stratégia pre Activity Selection dosahuje optimálne výsledky s časovou zložitosťou $O(n \log n)$
- Greedy prístup pre Coin Change je optimálny len pre kanonické systémy mincí, zatiaľ čo Dynamic Programming garantuje optimalitu univerzálne
- Vylepšenia greedy algoritmov ako GRASP, Iterated Greedy a hybridné metaheuristiké prístupy výrazne zvyšujú robustnosť a kvalitu riešení

Greedy algoritmy zostávajú neoddeliteľnou súčasťou moderného algoritmického dizajnu a poskytujú efektívne riešenia pre širokú škálu praktických optimalizačných problémov.