3. Heuristické metódy

Táto kapitola sa zaoberá metódami, ktoré nezaručujú nájdenie optimálneho riešenia a dokonca ani prípustného riešenia. Nazývame ich **heuristickými metódami** alebo jednoducho **heuristikami**.

3.1. Zdôvodnenie potreby heuristík

Pri úlohách diskrétnej optimalizácie sa heuristické metódy najčastejšie používajú na riešenie tých **kombinatorických** úloh, ktoré sú rozsiahle a NP-ťažké [31]. Medzi kombinatorické úlohy zaraďujeme úlohy, kde v konečnej množine riešení hľadáme optimálne riešenie, pričom jedným z možných spôsobov nájdenia optima je aj preskúmanie každého riešenia, ktorých počet je možné vyjadriť kombinatorickým výpočtom. Preskúmaním riešenia myslíme jeho porovnanie s ostatnými riešeniami z hľadiska kritéria optimality. V mnohých prípadoch však preskúmanie každého riešenia zadanej množiny je veľmi zdĺhavé a vyčerpávajúce.

Náročnosť hľadania optimálneho riešenia niektorých kombinatorických úloh si ukážeme na nasledujúcom prípade.

Do jednej predajne je potrebné prepraviť televízory z troch skladov. Televízory sa navzájom líšia hmotnosťou a cenami. S ohľadom na úložný priestor sa do malého pick-upu, ktorý poslúži pri preprave, zmestí najviac 6 škatúľ s televízormi, pričom nie je dovolené prekročiť nosnosť automobilu. Cieľom je naložiť na pick-up najdrahšiu možnú zostavu najviac šiestich televízorov z jedného skladu. V prvom sklade je 10 druhov televízorov, v druhom 50 a v treťom sklade je 100 druhov televízorov. Z každého druhu televízora je v skladoch práve jeden televízor.

Kvôli hladkému priebehu nakládky tovaru musí zodpovedný pracovník vopred určiť optimálnu zostavu televízorov. Z každého skladu má k dispozícii zoznamy televízorov spolu s ich cenami a hmotnosťami. Pracovník si vytvoril jednoduchý počítačový program, ktorý pre každú možnú zostavu najviac šiestich televízorov spočíta ich celkovú hmotnosť, potom pre tie zostavy, ktorých celková hmotnosť neprekročila kapacitu pick-upu, určí ich celkovú *cenu a nakoniec program určí zostavu s najväčšou celkovou cenou. Pri predbežnom testovaní programu pracovník zistil, že spracovanie jednej zostavy televízorov netrvá dlhšie ako 1 milisekundu. Optimálnu zostavu televízorov v prvom sklade určil program za necelú sekundu. Tento program sa pracovník chystá použiť pre výpočet optimálnej zostavy televízorov v druhom a treťom sklade. Ako dlho bude trvať výpočet optimálnej zostavy televízorov v týchto skladoch?*

Keďže celková hmotnosť niektorých zostáv televízorov môže prekročiť nosnosť automobilu, je nutné určiť, ktoré zostavy neprichádzajú pre prepravu do úvahy a zo zvyšných určiť tú s najväčšou celkovou cenou. Z uvedeného dôvodu je preto potrebné preskúmať celkové hmotnosti a celkové ceny každej zostavy televízorov. Zostavou televízorov pre prepravu pritom rozumieme samostatný televízor, alebo dvojicu, trojicu, štvoricu, päticu alebo šesticu televízorov.

V prvom sklade s desiatimi rôznymi televízormi môžeme vytvoriť celkom 847 zostáv, čo je súčet všetkých zostáv samostatných televízorov (10), všetkých dvojíc televízorov z desiatich možných (45), všetkých trojíc (120), všetkých štvoríc (210), všetkých pätíc (252) a všetkých zostáv šestíc televízorov vytvorených z desiatich televízorov (210). Pri dobe spracovania jednej zostavy, čo nie je viac ako 1 milisekunda, trval výpočet optimálneho riešenia uvedeným programom zanedbateľný čas. V druhom sklade, kde je 50 rôznych televízorov, je počet všetkých možných zostáv vytvorených z maximálne šiestich televízorov rovný číslu 18 260 635. Ak predpokladáme, že počítačové spracovanie jednej možnosti trvá okolo 1 milisekundy, potom program určí optimálnu zostavu približne za 5 hodín. Pri treťom sklade so 100 rôznymi druhmi televízorov je počet všetkým možných zostáv rovný číslu 1 271 427 895 a program by určil víťaznú zostavu za 14-15 dní.

Ak by sme podobným spôsobom hľadali najdrahšiu zostavu televízorov pre prepravu väčším automobilom, tak iba napríklad pre výpočet všetkých zostáv pozostávajúcich z desiatich televízorov, by sme museli na výsledok z počítača čakať v prípade druhého skladu skoro 4 mesiace a v prípade tretieho skladu takmer 550 rokov.

Tento spôsob preskúmania každého objektu (tu je objektom zostava televízorov), je vhodný iba pre veľmi malé úlohy. V praxi sa však stretávame s podobnými úlohami oveľa väčších rozmerov, a preto tento „násilný“ spôsob hľadania optimálneho riešenia nepoužívame, resp. používame iba pre rozmerovo veľmi malé úlohy.

Kombinatorické úlohy môžeme riešiť všeobecnými exaktnými algoritmami určenými pre riešenie úloh celočíselného lineárneho programovania alebo exaktnými algoritmami špeciálne vyvinutými pre konkrétne typy úloh, ktoré nájdu optimálne riešenie v priebehu niekoľkých sekúnd. Školské verzie a voľne šíriteľné verzie exaktných optimalizačných algoritmov však pri riešení kombinatorických úloh často zlyhávajú už pri menších úlohách. Buď je výpočet úlohy neúmerne dlhý alebo algoritmus zlyhá pre nedostatok pamäte počítača. Kvalitný profesionálny softvér síce dokáže vyriešiť veľké úlohy z praxe, ale obyčajne býva finančne veľmi náročný a vyžaduje aj kvalitnejšiu výpočtovú techniku, čo tiež zvyšuje náklady na riešenie úloh.

Je potrebné zvážiť, čo je výhodnejšie. Zakúpiť finančne náročný exaktný algoritmus, ktorý dokáže v krátkom čase optimálne vyriešiť aj veľmi rozsiahle úlohy, alebo sa zmieriť s jednoduchým finančne nenáročným heuristickým programom, ktorý v síce v krátkom čase, ale na úkor optimality nájde iba prípustné riešenie.

Heuristické algoritmy používame vtedy, ak neexistujú algoritmy, resp. nedisponujeme takými algoritmami, ktoré nájdu optimálne riešenie alebo aspoň nájdu pre nás vhodné riešenie, o ktorom môžeme s istotou povedať, že sa k optimálnemu riešeniu približuje, resp. kde je možné odchýlku od optimálneho riešenia aj číselne vyjadriť (aproximatívne algoritmy).

3.2. Princípy prostých minimalizačných heuristík

Ako sme už spomenuli, niektoré kombinatorické úlohy sú optimálne riešiteľné len vtedy, ak sú malého rozsahu. Pri úlohách, kde výpočet optimálneho riešenia rastie exponenciálne s rozsahom úlohy, sa v praxi musíme uspokojiť s heuristickými metódami, ktoré nezaručujú nájdenie optimálneho riešenia a pri ktorých mnohokrát ani nevieme odhadnúť, o koľko sa nájdené riešenie líši od optimálneho. Využijúc vlastností kritéria optimality, platných v práve spracovávanej časti priestoru riešení (lokálne kritérium), dokážu však heuristiky vo veľmi krátkom čase nájsť dobré riešenie danej úlohy [3], [33].

Vo všeobecnosti pracujú heuristiky dvomi spôsobmi. Pri prvom spôsobe sa prípustné východiskové riešenie postupne vylepšuje z hľadiska lokálneho kritéria optimality, pričom sa dbá na to, aby sme pri prechode k lepšiemu riešeniu neporušili prípustnosť riešenia. Pri druhom spôsobe riešenia heuristiky nájdu východiskové neprípustné riešenie s lepšou hodnotou účelovej funkcie ako by mohlo mať optimálne riešenie. Heuristika toto neprípustné riešenie postupnými krokmi upravuje tak, aby sa zmenšila miera neprípustnosti riešenia a aby sa pritom čo najmenej zhoršilo riešenie z hľadiska kritéria optimality. Heuristiku, využívajúcu prvý spôsob hľadania optima, nazývame **primárnou heuristikou (incrementation heuristic)**. **Duálna heuristika (construction heuristic)** pracuje tým druhým spôsobom.

Primárna heuristika skončí, ak získame prípustné riešenie, ktoré nie je možné zlepšiť z hľadiska lokálneho kritéria optimality bez toho, aby sme porušili prípustnosť riešenia. Práca duálneho algoritmu je ukončená, ak bolo dosiahnuté prípustné riešenie, alebo v prípade, že uvažovanými zmenami nie je možné zmenšiť mieru neprípustnosti.

Aby sme sa čo najviac priblížili k optimálnemu riešeniu, môžeme heuristiku použiť opakovane, vždy s iným východiskovým riešením, a zo získaných riešení vybrať to najlepšie. Môžeme tiež kombinovať primárnu s duálnou heuristikou a to tak, že riešenie získané duálnou heuristikou bude východiskovým riešením pre primárnu heuristiku, ktorá môže (ale nemusí) toto riešenie zlepšiť. Ak sa nám však nepodarí primárnou heuristikou prípustné riešenie vylepšiť, neznamená to, že toto riešenie je optimálnym riešením.

Úlohy diskrétnej optimalizácie sú charakteristické tým, že v **diskrétnej množine prípustných riešení**, t. j. v diskrétnej množine všetkých riešení, ktoré vyhovujú všetkým podmienkam zadaných úlohou, hľadajú také riešenie, ktoré má určitú, úlohou požadovanú vlastnosť optimálnu a to buď minimálnu alebo maximálnu. V matematických modeloch je táto vlastnosť vyjadrená **účelovou funkciou**, nazývanou aj **cieľovou funkciou**, **kriteriálnou funkciou** alebo tiež **kritériom optimality.** Účelová funkcia môže vyjadrovať cenu, zisky, náklady, dĺžku a pod. Spomedzi všetkých riešení potom hľadáme riešenie s maximálnou cenou, s maximálnym ziskom, s minimálnymi nákladmi, s minimálnou dĺžkou a pod.

O **globálnom kritériu** hovoríme v súvislosti s pôvodným kritériom, vyjadrujúcom hodnotu účelovej funkcie. **Lokálne kritérium** vyjadruje iba výhodnosť vykonania nejakej operácie. Pri heuristických metódach hľadáme riešenie splňujúce globálne kritérium pomocou lokálneho kritéria.

Lokálne kritérium nemusí byť zhodné s globálnym kritériom. Napríklad vo vyššie uvedenej úlohe, kde sa do auta snažíme naložiť televízory, je globálnym kritériom „čo najvyššia celková cena televízorov naložených do auta". Heuristika, ktorá takúto úlohu rieši, postupne preberá jednotlivé predmety (televízory) a o každom zvlášť rozhoduje, či bude alebo nebude naložený do auta. Pri výbere predmetu, o ktorom chce rozhodnúť, si heuristika pomáha lokálnym kritériom, ktoré môžeme zadefinovať aj takto: „Z množiny predmetov, ktoré nie sú doposiaľ vložené do auta, vyber ten, ktorý má najnižšiu hmotnosť". V tomto lokálnom kritériu sa nespomína cena predmetov ani napriek tomu, že hľadáme riešenie s najvyššou možnou celkovou cenou. Navyše v globálnom kritériu sa vyskytuje slovo najvyšší a v lokálnom kritériu jeho opozitum – slovo najnižší. Lokálne kritérium sme zadefinovali takýmto spôsobom na základe predpokladu, že ak na pick-up budeme prednostne nakladať predmety s malou hmotnosťou, tak sa naň zmestí viac predmetov a že čím viac bude v aute predmetov, tým vyššia môže byť ich celková cena. To však ale nemusí byť pravda.

Všeobecné princípy heuristík, ktoré teraz uvedieme, je nutné špeciálne prispôsobiť pre každý typ úlohy. Pre potreby tejto učebnice heuristiky môžeme rozdeliť na dva základné typy a to: **vkladacie heuristiky (insertion heuristics)** a **výmenné heuristiky (exchange heuristics)**.

Vkladacie heuristiky sú založené na vkladaní objektov do riešenia alebo na odstránení objektov riešenia.

Princíp primárnej vkladacej heuristiky

1. Vytvoríme počiatočné (východiskové) prípustné riešenie, najčastejšie to býva prázdna množina objektov. Objekty, o ktorých môže byť v heuristike rozhodované, budú tvoriť množinu doposiaľ nespracovaných objektov.
2. Za pomoci lokálneho kritéria vyberieme objekt z množiny doposiaľ nespracovaných objektov a v prípade, že sa tým nenaruší prípustnosť riešenia, upravíme ho pridaním tohto objektu do riešenia.
3. Bod 2 opakujeme, pokiaľ sa v množine nespracovaných objektov ešte nachádzajú nejaké objekty.

Princíp duálnej vkladacej heuristiky

1. Vytvoríme počiatočné (východiskové) neprípustné riešenie, najčastejšie to býva celá množina uvažovaných objektov.
2. Za pomoci lokálneho kritéria určíme objekt z aktuálneho riešenia a odstránime ho z neho.
3. Bod 2 opakujeme dovtedy, kým sa aktuálne riešenie nestane prípustným.

Výmennou heuristikou môžeme vylepšiť prípustné riešenie, ale tiež nemusíme a to ani v prípade, že existuje prípustné riešenie s lepšou hodnotou globálneho optimalizačného kritéria, ako má to, ktoré chceme vylepšiť. Prostá výmenná heuristika je primárna, čo znamená, že počas celého procesu vylepšovania pracujeme iba s prípustnými riešeniami.

Vo všeobecnosti je výmenná heuristika, ako je to aj zo samotného pomenovania heuristiky zrejmé, založená na výmene nejakých objektov. Spravidla vymieňame objekty, ktoré sú zaradené do riešenia, s tými, ktoré do riešenia zaradené nie sú. Ak touto výmenou objektov neporušíme prípustnosť riešenia, potom takúto výmenu nazveme **prípustnou výmenou.** V prípade, že výmenou uvažovaných predmetov vylepšíme hodnotu účelovej funkcie, potom takúto výmenu nazveme **vhodnou výmenou.** Aby výmena objektov bola realizovaná, musí byť prípustná a vhodná.

Z mnohých stratégií, ktorými sa riadia heuristiky, spomenieme dve, ktoré sa najvýraznejšie líšia. Sú to **stratégia „prvý vhodný“ (first**

admissible) a **stratégia „najlepší vhodný“ (best admissible).** Pri stratégii „prvý vhodný“ uskutočníme výmenu objektov pri nájdení prvej prípustnej a zároveň vhodnej výmeny a výmennú heuristiku zopakujeme pre nové riešenie. Pri stratégii „najlepší vhodný“ nájdeme najprv všetky prípustné a zároveň vhodné výmeny a z nich vyberieme tú výmenu objektov, ktorá najviac zlepší hodnotu účelovej funkcie. Výmenu uskutočníme a heuristiku zopakujeme pre nové riešenie.

Princíp výmennej heuristiky so stratégiou „prvý vhodný“

1. Z objektov prípustného riešenia vytvoríme množinu objektov zaradených do riešenia a z ostatných objektov úlohy vytvoríme množinu objektov nezaradených do riešenia.
2. Vytvoríme množinu usporiadaných dvojíc objektov, kde prvý člen dvojice tvorí jeden alebo viacero objektov z množiny zaradených objektov a druhý člen dvojice tvorí jeden alebo viacero objektov z množiny nezaradených objektov
3. Postupne preberáme dvojice. Zistíme, či by sa zmenila prípustnosť riešenia, ak by sme prvý prvok z dvojice preradili do množiny nezaradených predmetov a druhý prvok z dvojice by sme preradili do množiny zaradených predmetov. Akonáhle nájdeme dvojicu, ktorá výmenou neporuší prípustnosť riešenia a ktorá je zároveň vhodnou výmenou (zlepší hodnotu účelovej funkcie), výmenu uskutočníme a z novovzniknutého riešenia aktualizujeme množinu usporiadaných dvojíc, t. j. prejdeme na bod 2. Ak skúmaná dvojica netvorí prípustnú alebo vhodnú výmenu, odstránime ju z množiny usporiadaných dvojíc.
4. Bod 3 opakujeme, pokiaľ množina usporiadaných dvojíc ešte obsahuje nejaké dvojice.

Princíp výmennej heuristiky so stratégiou „najlepší vhodný“

1. Z objektov prípustného riešenia vytvoríme množinu objektov zaradených do riešenia a z ostatných objektov úlohy vytvoríme množinu objektov nezaradených do riešenia.

2. Vytvoríme množinu usporiadaných dvojíc objektov, kde prvý člen dvojice tvorí jeden alebo viacero objektov z množiny zaradených objektov a druhý člen dvojice tvorí jeden alebo viacero objektov z množiny nezaradených objektov
3. Postupne preberáme všetky prvky z množiny usporiadaných dvojíc a nájdeme všetky dvojice, pre ktoré je výmena prípustná a zároveň vhodná. Z nich určíme tú dvojicu, ktorá po výmene objektov najviac zlepší hodnotu účelovej funkcie, výmenu uskutočníme a získame tak nové riešenie. V prípade, že sme nenašli žiadnu dvojicu, pre ktorú by bola výmena prípustná a vhodná, tak prejdeme na bod 4, inak pokračujeme bodom 2.
4. Koniec heuristiky.

3.3. Algoritmizácia heuristických metód

V tejto časti navrhneme niekoľko heuristík pre riešenie úlohy o batohu, umiestňovaciu úlohu a úlohu obchodného cestujúceho a heuristiky pre nájdenie východiskového prípustného riešenia dopravnej úlohy.

Pre jednotlivé typy úloh uvedieme na začiatku každej podkapitoly konkrétne zadanie úlohy, na ktorom ukážeme ako postupujú navrhnuté heuristiky pri hľadaní **suboptimálneho** riešenia, t. j. takého riešenia, ktoré sa k optimálnemu riešeniu približuje, ale je od neho horšie. Veríme, že tieto postupy riešenia pomôžu čitateľom rýchlejšie pochopiť jednotlivé heuristické algoritmy a že budú tiež pomôckou pri ladení programov.

3.3.1 Heuristické metódy pre úlohu o batohu

Prvým typom úloh, pre ktoré navrhneme heuristiky je **úloha o batohu** (Knapsack problem).

Z piatich kontajnerov, pre ktoré poznáme ich hmotnosť a cenu obsahu, je potrebné vybrať a naložiť na nákladný automobil o nosnosti 15 ton tie, ktorých celková hmotnosť nepresiahne kapacitu automobilu a ktorých celková cena bude čo najväčšia. Ceny a hmotnosti kontajnerov sú uvedené v tabuľke 3.01.

Tabuľka 3.01

Ceny c_j a hmotnosti m_j kontajnerov pre úlohu o batohu

Kontajner	1	2	3	4	5
Cena [peňažná jednotka]	24	35	27	15	30
Hmotnosť [t]	6	7	3	5	6

Pre úlohu o batohu zvolíme bivalentné (0-1) rozhodovacie premenné x_j, ktoré pre indexy j = 1, ..., 5 budú modelovať rozhodnutie, či kontajner j bude naložený na automobil (x_j=1) alebo nebude naložený na automobil (x_j=0). V závislosti od variantu heuristiky budeme riešenia v jednotlivých krokoch heuristiky uvádzať ako vektory $\boldsymbol{x} = <x_1, x_2, x_3, x_4, x_5>$ alebo ako množinu indexov tých kontajnerov, o ktorých bolo rozhodnuté, že budú naložené na automobil. Matematický model úlohy o batohu s konkrétnymi údajmi, uvedenými v tabuľke 3.01, je nasledovný:

$$\textit{maximalizujte } f(\boldsymbol{x}) = 24x_1 + 35x_2 + 27x_3 + 15x_4 + 30x_5 \qquad (3.01)$$

$$\textit{za podmienok} \quad 6x_1 + 7x_2 + 3x_3 + 5x_4 + 6x_5 \le 15 \qquad (3.02)$$

$$x_j \in \{0,1\}, \text{ pre } j = 1, \dots, 5 \qquad (3.03)$$

Účelová funkcia (3.01) vyjadruje celkovú cenu kontajnerov, naložených na automobil, ktorú je potrebné maximalizovať. Ľavá strana podmienky (3.02) vyjadruje celkovú hmotnosť kontajnerov naložených na automobil a samotná podmienka (3.02) zabezpečuje, aby tá hmotnosť neprekročila 15 ton.

Heuristiky popíšeme všeobecne pre n predmetov (kontajnerov), kde j-ty predmet má cenu c_j a hmotnosť m_j pre j = 1, ..., n a kde kapacita batoha (nákladného automobilu) je K a f je cena súčasného riešenia $\boldsymbol{x} = <x_1, ..., x_n>$. Každé riešenie úlohy, ktoré neprekročí kapacitu batohu nazveme **prípustným riešením**. Aj riešenie $\boldsymbol{x} = <0, ..., 0>$, t. j. prázdny batoh, je prípustným riešením úlohy o batohu.

Aby malo zmysel riešiť úlohu o batohu predpokladáme, že suma hmotností všetkých predmetov je väčšia ako kapacita batohu (inak niet čo optimalizovať) a že hmotnosť každého predmetu je menšia ako kapacita batohu. Ak by totiž každý predmet, o ktorom chceme rozhodnúť, či ho vložíme do batoha alebo nie, mal hmotnosť väčšiu ako je kapacita batohu, nemohli by sme doň vložiť žiaden predmet.

V nasledujúcich heuristikách použijeme označenie *VK* pre **voľnú kapacitu** batohu a označenie *PV* pre **previs**. Voľnou kapacitou batohu rozumieme rozdiel kapacity batohu a súčtu hmotností predmetov, ktoré sa už v batohu nachádzajú. Previsom označuje celkovú hmotnosť, o ktorú je prekročená kapacita batohu.

Pre určenie konečnej množiny predmetov, ktorá bude tvoriť obsah batohu, musíme o každom predmete zvlášť rozhodnúť, či bude alebo nebude vložený do batoha. Pre toto rozhodovanie potrebujeme lokálne kritérium. Jednotlivé heuristiky sa budú líšiť v tom, ako toto lokálne kritérium zadefinujeme.

K návrhu algoritmu môžeme pristupovať rôznymi spôsobmi a to môže ovplyvňovať aj návrh údajových štruktúr pri počítačovej implementácii algoritmu. Pri nasledujúcich vkladacích heuristikách môžeme zabezpečiť použitie lokálneho kritéria dvomi spôsobmi. Pri prvom spôsobe pripravíme vstupné údaje do tvaru splňujúceho lokálne kritérium ešte pred samotným algoritmom. Ak napríklad lokálnym kritériom v primárnej vkladacej heuristike bude „čo najvyššia cena predmetu“, potom usporiadame predmety tak, aby sme mohli postupne preskúmavať predmety od tých s najväčšou cenou po tie s najmenšou cenou. Pri takomto variante heuristiky musíme použiť triediaci algoritmus na začiatku, najčastejšie v prvom kroku heuristiky.

Pri ďalšom spôsobe aplikácie lokálneho kritéria údaje pre heuristiku nepripravíme vopred, ale vždy, keď chceme preskúmať ďalší predmet, vyhľadáme ho na základe zadefinovaného kritéria z množiny doposiaľ nespracovaných predmetov. V tomto prípade potrebujeme vyhľadávací algoritmus, ktorý musíme použiť zakaždým, keď chceme preskúmať ďalší predmet.

Primárna vkladacia heuristika pre úlohu o batohu s prednostným výberom najcennejšieho predmetu

Pre túto heuristiku zadefinujeme lokálne kritérium takto: „*Rozhodujeme o predmete s najvyššou cenou z tých predmetov, o ktorých sme ešte* *nerozhodovali*." V navrhnutej heuristike sú s ohľadom na lokálne kritérium usporiadané predmety podľa klesajúcich cien hneď v úvode heuristiky a riešenie je uložené v 0-1-tkovom vektore $\boldsymbol{x}$.

Heuristika

1. Vytvor zoznam indexov predmetov $I = < i(1), ..., i(n) >$ usporiadaný v poradí klesajúcich cien predmetov. ($i(1)$ obsahuje index predmetu s najvyššou cenou a $i(n)$ obsahuje index najlacnejšieho predmetu).
2. Urči východiskové prípustné riešenie ako $\boldsymbol{x} = < 0, ..., 0 >$, polož $VK = K$ a $f = 0$. Nastav počítadlo predmetov $k = 1$.
3. Ak $k > n$, prejdi na krok 5. V opačnom prípade vyber k-ty prvok zoznamu I. Ak je $(VK - m_{i(k)}) < 0$, polož $k = k+1$ a opakuj krok 3.
4. Polož $x_{i(k)} = 1$, $VK = VK - m_{i(k)}$, $f = f + c_{i(k)}$, $k = k+1$. Ak je $VK > 0$, prejdi na krok 3, inak prejdi na krok 5.
5. Koniec algoritmu. Vektor $\boldsymbol{x}$ je prípustným riešením úlohy o batohu, f obsahuje cenu prípustného riešenia a VK obsahuje veľkosť nevyužitej časti kapacity batohu.

Podľa uvedeného variantu heuristiky dospejeme k riešeniu konkrétnej úlohy o piatich kontajneroch ($n = 5$) takto:

1. $I = <2, 5, 3, 1, 4>$. (Krok 1)
2. $\boldsymbol{x} = <0, 0, 0, 0, 0>$, $VK = 15$, $f = 0$, $k = 1$. (Krok 2)
3. $i(1) = 2$, $VK - m_{i(1)} = VK - m_2 = 15 - 7 = 8 > 0$. (Krok 3)
4. $\boldsymbol{x} = <0, 1, 0, 0, 0>$, $VK = 8$, $f = 35$, $k = 2$. (Krok 4)
5. $i(2) = 5$, $VK - m_5 = 8 - 6 = 2 > 0$. (Krok 3)
6. $\boldsymbol{x} = <0, 1, 0, 0, 1>$, $VK = 2$, $f = 65$, $k = 3$. (Krok 4)
7. $i(3) = 3$, $VK - m_3 = 2 - 3 = -1 < 0$, $k = 4$. (Krok 3)
8. $i(4) = 1$, $VK - m_1 = 2 - 6 = -4 < 0$, $k = 5$. (Krok 3)
9. $i(5) = 4$, $VK - m_4 = 2 - 5 = -3 < 0$, $k = 6$. (Krok 3)
10. $k > 5$. (Krok 3)
11. Koniec. $\boldsymbol{x} = <0, 1, 0, 0, 1>$, $VK = 2$, $f = 65$. (Krok 5)

Riešenie heuristiky je teda také, že na automobil budú naložené druhý a piaty kontajner s celkovou cenou 65 peňažných jednotiek, pričom nebude využitá celá kapacita automobilu.

Cvičenie 3.01

Navrhnite variant vkladacej heuristiky pre úlohu o batohu s prednostným výberom najcennejšieho predmetu, v ktorej bude lokálne kritérium používané až v momente, keď sa bude rozhodovať o predmete, ktorý sa má spracovať v nasledujúcom kroku, t. j. na začiatku heuristiky sa nepripraví poradie spracovávaných predmetov. Riešenie heuristiky bude v tvare zoznamu indexov predmetov zaradených do riešenia, nie v tvare 0-1 vektora $\boldsymbol{x}$.

Heuristika pre úlohu o batohu s lokálnym kritériom najvyššej ceny musí preskúmať každý predmet v prípade, že počas celého výpočtu je voľná kapacita väčšia ako nula. Nemáme totiž istotu, či sa medzi doposiaľ nepreskúmanými predmetmi nenachádza predmet s hmotnosťou menšou alebo rovnou voľnej kapacite. Iba v prípade, že nulová voľná kapacita batohu bude dosiahnutá skôr, akoby sme preskúmali všetky predmety, nemusí algoritmus rozhodovať o každom predmete zvlášť.

Primárna vkladacia heuristika pre úlohu o batohu s prednostným výberom predmetu s najmenšou hmotnosťou

V tomto prípade zadefinujeme lokálne kritérium takto: „*Rozhodujeme o predmete s najmenšou hmotnosťou z tých predmetov, o ktorých sme ešte nerozhodovali*.“

Pre toto lokálne kritérium je výhodné usporiadať predmety podľa stúpajúcich hmotností. To spôsobí, že heuristika nemusí preskúmať každý predmet. Pretože predmety budú usporiadané podľa ich stúpajúcej hmotnosti, algoritmus skončí v momente, ak by vloženie skúmaného predmetu nastavilo voľnú kapacitu na zápornú hodnotu alebo na nulu. Všetky ďalšie nepreskúmané predmety musia mať totiž hmotnosť vyššiu, ako je súčasná voľná kapacita batohu.

Cvičenie 3.02

Navrhnite jednotlivé kroky heuristiky pre úlohu o batohu s lokálnym kritériom, ktoré uprednostňuje výber predmetu s najmenšou hmotnosťou z tých predmetov, o ktorých sa ešte nerozhodovalo! Vyriešte touto heuristikou úlohu o batohu zadanú tabuľkou 3.01!

Primárna vkladacia heuristika pre úlohu o batohu s prednostným výberom predmetu s najvyšším koeficientom výhodnosti

Lokálne kritériá oboch predchádzajúcich heuristík pri určovaní poradia predmetov pre ich preskúmanie zohľadňovali iba jednu ich charakteristiku. Buď boli pri preskúmavaní uprednostňované predmety s vyššou cenou ako mali ostatné predmety alebo boli uprednostňované ľahšie predmety pred ostatnými predmetmi. Pre výslednú, čo najväčšiu cenu batohu však majú vplyv obe charakteristiky. Pomer koeficientov c_j / m_j pre predmet $j = 1, ..., n$ pomôže pri výbere poradia zohľadniť súčasne aj cenu predmetu j aj jeho hmotnosť. Keďže celková cena batoha má byť čo najväčšia je vhodné, aby boli do batoha postupne vkladané predmety s vyššou cenou, ale zároveň s menšou hmotnosťou, aby sa do batoha zmestilo, čo najviac predmetov, čo tiež môže napomôcť k výslednej cene batoha. Čím je teda pomer c_j / m_j pre j-ty predmet vyšší, tým je výhodnejšie uvažovať o jeho vložení do batoha. Ani heuristika s koeficientmi výhodnosti však nezaručuje nájdenie optimálneho riešenia a dokonca nezaručuje ani nájdenie lepšieho riešenia, ako by mohli nájsť iné heuristiky.

V tabuľke 3.02 sú vypočítané koeficienty výhodnosti pre úlohu o kontajneroch zadanú tabuľkou 3.01 je v nej uvedené aj poradie výhodnosti.

Tabuľka 3.02

Ukážka výpočtu koeficientov výhodnosti pre úlohu o batohu

Kontajner	1	2	3	4	5
Cena [peňažná jednotka] - c_j	24	35	27	15	30
Hmotnosť [t] - m_j	6	7	3	5	6
Koeficienty výhodnosti c_j / m_j	4	5	9	3	5
Poradie výhodnosti	4	2	1	5	3

Lokálne kritérium **pre primárnu heuristiku s koeficientmi výhodnosti** pre úlohu o batohu zadefinujeme nasledovne: „*Rozhodujeme o predmete* *s najväčším pomerom c_j / m_j z tých predmetov, o ktorých sme ešte nerozhodovali.*“ Pre toto lokálne kritérium je výhodné usporiadať predmety podľa klesajúcich pomerov c_j / m_j pre $j = 1, ..., n$. Jednotlivé kroky heuristiky budú zhodné s krokmi primárnej vkladacej heuristiky pre úlohu o batohu s prednostným výberom najcennejšieho predmetu až na krok 1, kde poradie spracovania predmetov bude určené podľa klesajúcich koeficientov výhodnosti.

Cvičenie 3.03

Aplikujte lokálne kritérium s koeficientmi výhodnosti na heuristiku o batohu s prednostným výberom najcennejšieho predmetu a vyriešte ňou úlohu o kontajneroch zadanú tabuľkou 3.01!

Cvičenie 3.04

Prekročili sme nosnosť výťahu, ktorým chceme vyviesť balíky na vyššie poschodie. Výťah má bezpečnostnú poistku a nepohne sa, pokiaľ hmotnosť nákladu nebude menšia aspoň o 15 hmotnostných jednotiek (hj). Vo výťahu je veľa dôležitých balíkov, ale je medzi nimi 5 balíkov, ktoré by mohli byť prepravené výťahom inokedy, nemusíme ich však odstrániť z výťahu všetky. Tieto balíky majú postupne 6, 7, 3, 5 a 6 hj a sú poistené pred stratou na 24, 35, 27, 15 a 30 peňažných jednotiek. Je potrebné určiť, ktoré z týchto piatich balíkov musíme odstrániť z výťahu, aby celková hmotnosť odstránených balíkov bola aspoň 15 hj a aby celková poistná suma (pre prípade straty) bola čo najnižšia.

Vytvorte matematický model tejto „obrátenej“ úlohy o batohu a pre jej riešenie navrhnite primárnu vkladaciu heuristiku. Nezabudnite definovať aj lokálne kritérium pre odstránenie balíka z riešenia. Navrhnutou heuristikou zadanú úlohu vyriešte!

Duálna vkladacia heuristika (insertion heuristic) pre úlohu o batohu

Ako sme už vyššie spomínali, duálna heuristika štartuje v neprípustnom riešení, ktoré je z hľadiska globálneho kritéria optimality veľmi dobré. Postupnými krokmi znižuje mieru neprípustnosti na úkor kvality riešenia, pričom zhoršenie riešenia má byť najmenšie možné z hľadiska lokálneho kritéria. Pre úlohu o batohu tvorí východiskové neprípustné riešenie všetkých n predmetov riešenej úlohy, t. j. všetky predmety sú vybraté do batoha bez ohľadu na jeho kapacitu. Samozrejme toto riešenie je neprípustné iba v prípade, že suma hmotností všetkých predmetov je väčšia ako kapacita batohu. Ak táto podmienka nie je splnená, potom východiskové riešenie duálnej úlohy je automaticky optimálne.

Riešenie úlohy o batohu je neprípustné, ak hodnota previsu PV je kladná, teda ak je prekročená kapacita batohu. Znižovanie miery neprípustnosti je tu realizované postupným „vyhadzovaním" predmetov z batohu, čo zakaždým zníži hodnotu previsu. Heuristika sa ukončí vo chvíli, keď po odstránení predmetu z batoha hodnota previsu bude menšia alebo rovná nule.

Podobne, ako pri primárnych heuristikách pre úlohu o batohu, aj pri duálnych heuristikách musíme navrhnúť lokálne kritérium pre odstránenie predmetu z batoha. Z charakteru úlohy vyplývajú tri základné lokálne kritéria:

1. **minimálna cena** predmetu z tých, ktoré tvoria aktuálne neprípustné riešenie (predmety usporiadané podľa ich rastúcich cien),
2. **maximálna hmotnosť** predmetu z tých, ktoré tvoria aktuálne neprípustné riešenie (predmety usporiadané podľa ich klesajúcich hmotností),
3. **minimálny koeficient výhodnosti** predmetu z tých, ktoré tvoria aktuálne neprípustné riešenie (predmety usporiadané podľa ich rastúcich koeficientov výhodnosti).

Pre ktorékoľvek z uvedených lokálnych kritérií, končí duálna vkladacia heuristika pre úlohu o batohu rovnako a to v momente, keď je previs $PV \leq 0$. Ak je hmotnosť každého predmetu menšia ako kapacita batohu, zostane nakoniec v batohu najmenej jeden predmet.

Heuristika

1. Vytvor zoznam indexov predmetov $I = < i(1), ..., i(n) >$ usporiadaný podľa zadaného lokálneho kritéria.
2. Urči východiskové prípustné riešenie ako $\boldsymbol{x} = < 1, ..., 1 >$, polož

$PV = \sum_{j=1}^{n} m_j - K$, ďalej polož $f = \sum_{j=1}^{n} c_j$ a nastav počítadlo $k = 1$.

3. Ak je $PV \leq 0$, prejdi na krok 4. V opačnom prípade vyber k-ty prvok zo zoznamu I a polož $x_{i(k)} = 0$, $PV = PV - m_{i(k)}$, $f = f - c_{i(k)}$, $k = k+1$. Opakuj krok 3.
4. Koniec algoritmu. Vektor $\boldsymbol{x}$ je prípustným riešením úlohy o batohu, f obsahuje cenu prípustného riešenia a absolútna hodnota $|PV|$ obsahuje veľkosť nevyužitej časti kapacity batohu.

Riešenie heuristiky môžeme uchovávať aj v tvare zoznamu Z indexov predmetov zaradených do riešenia, t. j. vybraných do batoha.

V tabuľke 3.03 sú riešenia duálnej vkladacej heuristiky pre úlohu o kontajneroch zadanú tabuľkou 3.01, resp. tabuľkou 3.02, v ktorej sú uvedené aj koeficientmi výhodnosti tejto úlohy. Postupy prechodov od neprípustných východiskových riešení ku konečným prípustným riešeniam pre jednotlivé lokálne kritéria sú uvedené v stĺpcoch tabuľky.

Tabuľka 3.03

Postup riešenia duálnej vkladacej heuristiky pre úlohu o kontajneroch

	Minimálna cena predmetu	**Maximálna hmotnosť predmetu**	**Minimálny koeficient výhodnosti predmetu**
1.	$\boldsymbol{x}$ = <1,1,1,1,1> Z = {1,2,3,4,5} PV = 12, f = 131	$\boldsymbol{x}$ = <1,1,1,1,1> Z = {1,2,3,4,5} PV = 12, f = 131	$\boldsymbol{x}$ = <1,1,1,1,1> Z = {1,2,3,4,5} PV = 12, f = 131
2.	$\boldsymbol{x}$ = <1,1,1,0,1> Z = {1,2,3,5} PV = 7, f = 72	$\boldsymbol{x}$ = <1,0,1,1,1> Z = {1,3,4,5} PV = 5, f = 96	$\boldsymbol{x}$ = <1,1,1,0,1> Z = {1,2,3,5} PV = 7, f = 116
3.	$\boldsymbol{x}$ = <0,1,1,0,1> Z = {2,3,5} PV = 1, f = 92	$\boldsymbol{x}$ = <0,0,1,1,1> Z = {3,4,5} PV = -1, f = 72	$\boldsymbol{x}$ = <0,1,1,0,1> Z = {2,3,5} PV = 1, f = 92
4.	$\boldsymbol{x}$ = <0,1,0,0,1> Z = {2,5} PV = -2, f = 65		$\boldsymbol{x}$ = <0,0,1,0,1> Z = {3,5} PV = -6, f = 57

Cvičenie 3.05

Pre riešenie „obrátenej“ úlohy o batohu navrhnite duálnu vkladaciu heuristiku a vyriešte ňou konkrétnu úlohu, zadanú v cvičení 3.04!

Výmenná heuristika pre úlohu o batohu

V úlohe o batohu navzájom vymieňame jeden predmet, ktorý je zaradený do batohu, s iným predmetom, ktorý v batohu zaradený nie je. Ako východiskové riešenie môžeme použiť niektoré z riešení, ktoré sme získali primárnou alebo duálnou heuristikou.

Primárna výmenná heuristika pre úlohu o batohu so stratégiou „prvý vhodný“

1: Z prípustného riešenia úlohy o batohu s aktuálnou hodnotou účelovej funkcie f a voľnou kapacitou VK vytvor množinu indexov Z predmetov zaradených do batoha a množinu indexov predmetov N nezaradených do batoha. Z množín Z a N vytvor zoznam S usporiadaných dvojíc (z, n) takých, že $z \in Z$ a $n \in N$.

2: Ak je zoznam S prázdny, prejdi na krok 5. V opačnom prípade vezmi dvojicu $(z, n) \in S$ a zisti, či by prípadná výmena z a n bola prípustná.
Ak je $VK + m_z - m_n \geq 0$ (prípustná výmena), prejdi na bod 3. V opačnom prípade odstráň dvojicu (z, n) zo zoznamu S a zopakuj bod 2.

3: Ak je $c_n \leq c_z$ (nevýhodná výmena), odstráň dvojicu (z, n) zo zoznamu S a pokračuj krokom 2.
V opačnom prípade, t. j. ak je $c_n > c_z$ (vhodná výmena), polož $f=f-c_z+c_n$, $VK = VK + m_z - m_n$ a aktualizuj riešenie výmenou predmetov: $Z = (Z - \{z\}) \cup \{n\}$ a $N = (N - \{n\}) \cup \{z\}$ (t. j. $x_z = 0$ a $x_n = 1$).

4: Z množín Z a N vytvor **nový** zoznam S usporiadaných dvojíc (z, n) takých, že $z \in Z$ a $n \in N$ a pokračuj krokom 2.

5: Koniec algoritmu. Vektor $\boldsymbol{x}$ je prípustným riešením úlohy o batohu, resp. zoznam Z obsahuje indexy tých predmetov, ktoré sú zaradené do batoha, hodnota f je cena prípustného riešenia a VK obsahuje veľkosť nevyužitej časti kapacity batohu.

Jedným z prípustných riešení úlohy o batohu, ktorá bola zadaná tabuľkou 3.01 je $\boldsymbol{x} = <1, 0, 1, 1, 0>$ s hodnotou účelovej funkcie $f = 66$ a voľnou kapacitou $VK = 1$. Výmenná heuristika so stratégiou „prvý vhodný“ postupovala pri hľadaní lepšieho riešenia takto:

1. $Z = \{1, 3, 4\}$, $N = \{2, 5\}$, $f = 66$, $VK = 1$,
   $S = \{(1, 2), (1, 5), (3, 2), (3, 5), (4, 2), (4, 5)\}$. (Krok 1)
2. Pre $(1, 2) \in S$ je $VK + m_1 - m_2 = 1 + 6 - 7 \geq 0$. (Krok 2)
3. $c_2 = 35 > c_1 = 24$, $\boldsymbol{f} = f - c_1 + c_2 = 66 - 24 + 35 = \mathbf{77}$, $\boldsymbol{VK} = VK + m_1 - m_2 =$ $1+6-7 = \mathbf{0}$, $Z = \{2, 3, 4\}$, $S = \{1, 5\}$, t. j. $\boldsymbol{x} = (0, 1, 1, 1, 0)$. (Krok 3)
4. $S = \{(2, 1), (2, 5), (3, 1), (3, 5), (4, 1), (4, 5)\}$. (Krok 4)
5. Pre $(2, 1) \in S$ je $VK + m_2 - m_1 = 0+7-6 \geq 0$ (prípustná výmena). (Krok 2)
6. $c_1 = 24 < c_2 = 35$, $S = \{(2, 5), (3, 1), (3, 5), (4, 1), (4, 5)\}$. (Krok 3)
7. Pre $(2, 5) \in S$ je $VK + m_2 - m_5 = 0+7-6 \geq 0$ (prípustná výmena). (Krok 2)
8. $c_5 = 30 < c_2 = 35$, $S = \{(3, 1), (3, 5), (4, 1), (4, 5)\}$. (Krok 3)
9. Pre $(3, 1) \in S$ je $VK + m_3 - m_1 = 0 + 3 - 6 \leq 0$ (neprípustná výmena), teda $S = \{(3, 5), (4, 1), (4, 5)\}$. (Krok 2)
10. Pre $(3, 5) \in S$ je $VK + m_3 - m_5 \leq 0$, teda $S = \{(4, 1), (4, 5)\}$. (Krok 2)
11. Pre $(4, 1) \in S$ je $VK + m_4 - m_1 \leq 0$, teda $S = \{(4, 5)\}$. (Krok 2)
12. Pre $(4, 5) \in S$ je $VK + m_4 - m_5 \leq 0$, teda $S = \emptyset$. (Krok 2)
13. Koniec. $\boldsymbol{x} = <0, 1, 1, 1, 0>$, $Z = \{2, 3, 4\}$, $f = 77$, $VK = 0$. (Krok 5)

Primárna výmenná heuristika pre úlohu o batohu so stratégiou „najlepší vhodný“

1: Z prípustného riešenia úlohy o batohu s aktuálnou hodnotou účelovej funkcie f a voľnou kapacitou VK vytvor množinu indexov Z predmetov zaradených do batoha a množinu indexov predmetov N nezaradených do batoha. Z množín Z a N vytvor zoznam S usporiadaných dvojíc (z, n) takých, že $z \in Z$ a $n \in N$. Polož $f^* = f$ a $Akt = 0$.

2: Ak je zoznam S prázdny, prejdi na krok 4. V opačnom prípade vezmi nasledujúcu dvojicu $(z, n) \in S$.
Ak je $VK + m_z - m_n \geq 0$ (prípustná výmena) a zároveň je $c_n > c_z$ (vhodná výmena) a zároveň ak je $f^* < f - c_z + c_n$, polož $f^* = f - c_z + c_n$, $z^* = z$, $n^* = n$ a $Akt = 1$.

3: Odstráň dvojicu (z, n) zo zoznamu S a pokračuj bodom 2.

4: Ak je $Akt = 0$, prejdi na bod 5. Inak aktualizuj riešenie: $f = f^*$, $VK = VK + m_{z^*} - m_{n^*}$, $Z = (Z - \{z^*\}) \cup \{n^*\}$, $N = (N - \{n^*\}) \cup \{z^*\}$ (t. j. $x_{z^*} = 0$ a $x_{n^*} = 1$), $Akt = 0$. Z množín Z a N vytvor **nový** zoznam S usporiadaných dvojíc (z, n) takých, že $z \in Z$ a $n \in N$ a pokračuj krokom 2.

5: Koniec algoritmu. Vektor $\boldsymbol{x}$ je prípustným riešením úlohy o batohu, resp. zoznam Z obsahuje indexy tých predmetov, ktoré sú zaradené do batoha, hodnota f je cena prípustného riešenia a VK obsahuje veľkosť nevyužitej časti kapacity batohu.

Výmenná heuristika so stratégiou „najlepší vhodný“ postupovala pri vylepšovaní toho istého východiskového riešenia, aké sme použili pri stratégii „prvý vhodný“, takto:

1. $Z = \{1, 3, 4\}$, $N = \{2, 5\}$, $f = 66$, $f^* = 66$, $VK = 1$, $Akt = 0$,
   $S = \{(1, 2), (1, 5), (3, 2), (3, 5), (4, 2), (4, 5)\}$. (Krok 1)
2. Pre $(1, 2) \in S$ je $VK + m_1 - m_2 = 1 + 6 - 7 \geq 0$, $c_2 = 35 > c_1 = 24$ a
   $f^* = 66 < f - c_1 + c_2 = 66 - 24 + 35$, preto:
   $\boldsymbol{f^* = 66 - 24 + 35 = 77}$, $z^* = 1$, $n^* = 2$ a $Akt = 1$. (Krok 2)
3. $S = \{(1, 5), (3, 2), (3, 5), (4, 2), (4, 5)\}$. (Krok 3)
4. Pre $(1, 5) \in S$ je $VK + m_1 - m_5 = 1 + 6 - 6 \geq 0$, $c_5 = 35 > c_1 = 24$ **ale**
   $f^* = 77 > f - c_1 + c_5 = 66 - 24 + 30$. (Krok 2)
5. $S = \{(3, 2), (3, 5), (4, 2), (4, 5)\}$. (Krok 3)
6. Pre $(3, 2) \in S$ je $VK + m_3 - m_2 = 1 + 3 - 7 < 0$. (Krok 2)
7. $S = \{(3, 5), (4, 2), (4, 5)\}$. (Krok 3)
8. Pre $(3, 5) \in S$ je $VK + m_3 - m_5 = 1 + 3 - 6 < 0$. (Krok 2)
9. $S = \{(4, 2), (4, 5)\}$. (Krok 3)
10. Pre $(4, 2) \in S$ je $VK + m_4 - m_2 = 1 + 5 - 7 < 0$. (Krok 2)
11. $S = \{(4, 5)\}$. (Krok 3)
12. Pre $(4, 5) \in S$ je $VK + m_4 - m_5 = 1 + 5 - 6 \geq 0$, $c_5 = 30 > c_4 = 15$ a
    $f^* = 77 < f - c_4 + c_5 = 66 - 15 + 30$, preto:
    $\boldsymbol{f^* = 66 - 15 + 30 = 81}$, $z^* = 4$, $n^* = 5$ a $Akt = 1$. (Krok 2)
13. $S = \emptyset$. (Krok 3)
14. $f = 81$, $VK = VK + m_{z^*} - m_{n^*} = 1 + 5 - 6 = 0$, $Z = \{1, 3, 5\}$, $N = \{2, 4\}$,
    teda $\boldsymbol{x} = <1, 0, 1, 0, 1>$, $Akt = 0$ a $S = \{(1, 2), (1, 4), (3, 2), (3, 4), (5, 2),$

(5, 4)}. (Krok 4)

15. Pri ďalšom postupe pre žiadnu dvojicu z nového zoznamu *S* neboli splnené všetky podmienky z bodu 2 heuristiky, ktoré sú potrebné pre aktualizáciu súčasného najlepšieho riešenia. Nakoniec zostal zoznam *S* prázdny a príznak pre aktualizáciu $Akt = 0$. (Kroky 2 až 4)
16. Koniec. $\boldsymbol{x} = <1, 0, 1, 0, 1>$, $Z = \{1, 3, 5\}$, $f = 81$, $VK = 0$. (Krok 5)

Cvičenie 3.06

Na východiskové prípustné riešenie $\boldsymbol{x} = <1, 1, 0, 0, 0>$ s hodnotu účelovej funkcie $f(\boldsymbol{x}) = 59$ pre úlohu o batohu zadanú tabuľkou 3.01 aplikujte výmennú heuristiku so stratégiou „prvý vhodný“ a výmennú heuristiku so stratégiou „najlepší vhodný“!

3.3.2 Heuristické metódy pre umiestňovacie úlohy

Kapacitne neobmedzená umiestňovacia úloha (Uncapacitated Facility Location Problem - UFLP) je ďalším typom úlohy, vhodným pre riešenie heuristikou. Tento typ úlohy, ktorý tiež nazývame aj **lokačnou úlohou**, tiež patrí medzi NP-ťažké kombinatorické úlohy. Okrem iného to znamená aj to, že exaktné algoritmy (hľadajúce optimálne riešenia lokačných úloh) môžu zlyhať pri riešení rozsiahlych úloh. Zlyhanie môže nastať kvôli nedostatku pamäte počítača alebo kvôli neúmernej dĺžke výpočtu optimálneho riešenia (niekoľko desiatok hodín až dní). Heuristikou dosiahneme nájdenie prípustného riešenia v krátkom až zanedbateľnom čase, no bez dôkazu, či nájdené riešenie je optimálne a bez možnosti učenia odchýlky nájdeného riešenia od optimálneho.

Umiestňovacia úloha môže byť zadaná nasledovne:

Množina I je množina uzlov dopravnej siete, kde môžu byť umiestnené sklady, množina J je množina uzlov dopravnej siete, kde sú umiestnení zákazníci, ktorí budú obsluhovaní z umiestnených (postavených) skladov. Každý sklad by dokázal obslúžiť všetkých zákazníkov, aj keby bol umiestnený iba jeden sklad, a každý zákazník musí byť obslúžený práve z jedného z umiestnených skladov. Sú známe fixné náklady f_i *spojené s umiestnením skladu v mieste* $i \in I$ *a aj náklady* c_{ij} *pre uspokojenie požiadavky každého zákazníka* $j \in J$ *z každého kandidáta na umiestnenie skladu* $i \in I$*. Úlohou je* *určiť, ktoré sklady budú umiestnené tak, aby celkové fixné náklady umiestnených skladov spolu s nákladmi na uspokojenie všetkých zákazníkov boli* ***minimálne****.*

V tabuľke 3.03 sú uvedené fixné náklady a náklady na uspokojenie zákazníkov zo skladov pre konkrétnu úlohu s desiatimi zákazníkmi Z_1 *až* Z_{10} *a s piatimi možnými umiestneniami skladov* S_1 *až* S_5*. Údaje z tabuľky použijeme pre ukážku práce navrhnutých heuristík.*

Tabuľka 3.03

Fixné náklady f_i a náklady na uspokojenie zákazníkov zo skladov c_{ij}

	f_i
S_1	25
S_2	95
S_3	40
S_4	73
S_5	62

c_{ij}	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}
S_1	10	20	20	65	50	85	100	45	75	10
S_2	10	5	80	55	60	30	35	10	20	50
S_3	85	75	15	95	50	5	20	90	40	90
S_4	70	85	100	20	40	55	10	85	30	80
S_5	100	40	15	70	30	40	90	5	65	20

V matematickom modeli kapacitne neobmedzenej umiestňovacej úlohy bivalentná (0-1) premenná y_i modeluje rozhodnutie o tom, či bude v mieste $i \in I$ umiestnený sklad ($y_i = 1$) alebo nebude ($y_i = 0$). Bivalentná (0-1) premenná z_{ij} modeluje rozhodnutie o tom, či zákazník $j \in J$ bude priradený mo6n0mu umiestneniu skladu $i \in I$ ($z_{ij} = 1$) alebo nebude ($z_{ij} = 0$). Matematický model kapacitne neobmedzenej umiestňovacej úlohy má tvar:

$$minimalizujte\ f(\boldsymbol{x}) = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \tag{3.04}$$

$$za\ podmienok \quad \sum_{i \in I} z_{ij} = 1 \quad pre\ j \in J \tag{3.05}$$

$$z_{ij} \le y_i \quad pre\ i \in I, j \in J \tag{3.06}$$

$$y_i \in \{0,1\} \quad pre\ i \in I \tag{3.07}$$

$$z_{ij} \in \{0,1\} \quad pre\ i \in I, j \in J \tag{3.08}$$

Účelová funkcia (3.04) vyjadruje celkové náklady na umiestnenie stredísk a obsluhu zákazníkov z umiestnených skladov. Podmienka (3.05) zabezpečuje podmienku, aby každý zákazník bol priradený práve k jednému skladu a podmienka (3.06) vyjadruje podmienku, že ak zákazník j bude priradený k mo6n0mu umiestneniu skladu i, potom sklad v mieste i musí byť umiestnený (ak z_{ij} = 1, potom y_i = 1).

Pre kapacitne neobmedzenú umiestňovaciu úlohu navrhneme dve heuristiky, *primárnu vkladaciu heuristiku* a *primárnu výmennú heuristiku*.

V súlade s matematickým modelom (3.04) – (3.08) môžeme prípustné riešenia, ktoré postupne získame heuristikou, uchovávať v nula-jednotkovom vektore $\boldsymbol{y} = \langle y_1, y_2, y_3, y_4, y_5 \rangle$ a v matici $\boldsymbol{z} = \begin{pmatrix} z_{11} & \cdots & z_{1,10} \\ \vdots & \ddots & \vdots \\ z_{51} & \cdots & z_{5,10} \end{pmatrix}$. Maticu $\boldsymbol{z}$ však nemusíme aktualizovať zakaždým, keď prejdeme k novému prípustnému riešeniu, ale stačí ju naplniť číslami 0 a 1 až po ukončení heuristiky na základe riešenia $\boldsymbol{y}$, teda podľa toho, ktoré sklady určila heuristika pre umiestnenie. Podmienka (3.05) v modeli umiestňovacej úlohy zabezpečuje, že zákazník bude priradený práve k jednému skladu a keďže účelová funkcia (3.04) je minimalizačná, stačí nájsť pre zákazníka j ten z umiestnených skladov, ktorý má najmenší koeficient c_{ij} a zodpovedajúcu premennú x_{ij} nastaviť na hodnotu 1. Takto získame maticu riešení $\boldsymbol{z}$, ktorá má v každom stĺpci práve jednu jednotku, okrem toho matica $\boldsymbol{z}$ bude obsahovať iba samé nuly.

Ak označíme symbolom I_1 množinu umiestnených skladov (t. j. tých, pre ktoré je $y_i = 1$, $i \in I$), potom účelovú funkciu (3.04) spočítame ako

$$f(I_1) = \sum_{i \in I_1} y_i + \sum_{j \in J} min\{\, c_{ij} : i \in I_1 \,\}\,. \tag{3.09}$$

Uvedieme ukážku výpočtu účelovej funkcie podľa vzťahu (3.09) pre úlohu zadanú tabuľkou 3.03. Predpokladajme, že pre umiestnenie boli vybrané sklady 1 a 5, potom vektor umiestnení pre zadanú úlohu bude $\boldsymbol{y} = \langle 1,0,0,0,1 \rangle$ a množina I_1 = {1,5}. Zodpovedajúce koeficienty, ktorých sčítaním získame hodnotu účelovej funkcie rovnú číslu 477, sú zvýraznené v tabuľke 3.04.

Tabuľka 3.04

Výpočet hodnoty účelovej funkcie pre množinu $I_1 = \{1, 5\}$

	f_i
S_1	**25**
S_2	95
S_3	40
S_4	73
S_5	**62**

c_{ij}	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}
S_1	**10**	**20**	20	**65**	50	85	100	45	75	**10**
S_2	10	5	80	55	60	30	35	10	20	50
S_3	85	75	15	95	50	5	20	90	40	90
S_4	70	85	100	20	40	55	10	85	30	80
S_5	100	40	**15**	70	**30**	**40**	**90**	**5**	**65**	20

Vektor priradení zákazníkov ku skladom ***z*** pre umiestnenia z $I_1 = \{1,5\}$ potom bude obsahovať hodnoty 1 tam, kde sú v tabuľke 3.04 zvýraznené hodnoty, a bude obsahovať hodnoty 0 na ostatných pozíciách. Tabuľka 3.05 zobrazuje vektor umiestnení ***y*** a maticu priradení ***z***.

Tabuľka 3.05

Riešenie – vektor *y* a matica *z*

y
1
0
0
0
1

z

1	**1**	0	**1**	0	0	0	0	0	**1**
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	**1**	0	**1**	**1**	**1**	**1**	**1**	0

Cvičenie 3.07

Ak pre umiestňovaciu úlohu zadanú tabuľkou 3.03 pridáme podmienku, že smieme umiestniť práve jeden sklad, ktorý to bude?

Cvičenie 3.08

Pre umiestňovaciu úlohu zadanú tabuľkou 3.03 vypočítajte hodnotu účelovej funkcie, ak vektor umiestnení je ***y*** = <0, 1, 0, 1, 1>!

Primárna vkladacia heuristika pre kapacitne neobmedzenú umiestňovaciu úlohu

Počas vykonávania heuristiky stačí uchovávať iba informáciu o umiestneniach skladov. Môžeme použiť množinu I_1, ktorá bude obsahovať indexy umiestnených skladov alebo môžeme na to využiť vektor $\boldsymbol{y}$, ktorý bude obsahovať hodnoty 1 na pozíciách, zodpovedajúcich umiestneným skladom, a hodnoty 0 na ostatných pozíciách.

Navrhnutá heuristika postupne vyberá doposiaľ nepreskúmané sklady. V prípade, žeby pridaním skladu do množiny umiestnených skladov I_1 klesla hodnota účelovej funkcie (3.04), vloží ho do nej. Heuristika končí, ak sme preskúmali každý sklad.

Lokálne kritérium pre vloženie skladu do riešenia, teda poradie spracovávania skladov, môžeme navrhnúť rôznymi spôsobmi. Sklady môžu byť preskúmané napríklad:

1. v poradí rastúcich hodnôt indexov $i \in I$,
2. v poradí určeným generátorom náhodných čísel,
3. v poradí rastúcich hodnôt $(f_i + \sum_{j \in J} c_{ij})$ pre kandidátov $i \in I$,
4. v poradí klesajúcich koeficientov výhodnosti. Výhodnosť umiestnenia skladu v mieste i môže byť definovaná napríklad počtom zákazníkov, pre ktorých je daný sklad najbližšie. Čím viac je takých zákazníkov, tým väčší je koeficient výhodnosti.

Ak ešte pred samotnou heuristikou určíme poradie spracovania skladov pre uvedené lokálne kritéria, možno prípady 2 až 4 po preindexovaní považovať za prípad 1.

Postup primárnej vkladacej heuristiky pre kapacitne neobmedzenú umiestňovaciu úlohu môžeme zhrnúť do nasledujúcich krokov:

1: Vytvor zoznam indexov kandidátov na umiestnenie $I_v = (i_1, ..., i_n)$ usporiadaný podľa zadaného lokálneho kritéria. Množina I_v obsahuje na začiatku heuristiky všetky (nepreskúmané) sklady z množiny I.

2: Inicializuj množinu umiestnených skladov $I_1 = \{i_1\}$ (aktuálne riešenie). Vypočítaj aktuálnu hodnotu účelovej funkcie f^* pre sklad i_1 podľa vzťahu

$f^* = f_{i_1} + \sum_{j\in J} c_{i_1 j}$. Nastav počítadlo skladov $s = 2$.

3: Ak $s > n$, prejdi na krok 4. V opačnom prípade vyber ďalšieho kandidáta k na umiestnenie strediska zo zoznamu I_v. Vyhodnoť hodnotu účelovej funkcie podľa vzťahu

$$f(I_1 \cup \{i_k\}) = \sum_{i\in I_1\cup\{i_k\}} f_i + \sum_{j\in J} \min\left\{c_{ij} : i \in I_1 \cup \{i_k\}\right\}.$$

Ak je $f^* > f(I_1 \cup \{i_k\})$, tak polož $f^* = f(I_1 \cup \{i_k\}), I_1 = I_1 \cup \{i_k\}$, $p = p + 1$ a zopakuj krok 3. V opačnom prípade prejdi na krok 4.

4: Koniec algoritmu. Množina I_1 obsahuje umiestnené sklady, f^* obsahuje hodnotu účelovej funkcie riešenia.

Vektor umiestnení vytvoríme takto: pre všetky $i \in I_1$ položíme $y_i = 1$ a zostávajúce zložky vektora $\boldsymbol{y}$ nastavíme na hodnotu 0. Maticu priradení $\boldsymbol{z}$ určíme takto: každého zákazníka $j \in J$ priradíme k tomu skladu $k \in I_1$, pre ktorý platí: $k = solmin\{c_{ij} : i \in I_1\}$, t. j. pre každého zákazníka $j \in J$ položíme $z_{kj} = 1$ a ostatné prvky v stĺpci j matice $\boldsymbol{z}$ položíme rovné hodnote 0. Matica $\boldsymbol{z}$ bude mať teda v každom stĺpci práve jednu hodnotu rovnú 1.

Jednotlivé kroky tejto heuristiky ukážeme na konkrétnom príklade kapacitne neobmedzenej umiestňovacej úlohy zadanej tabuľkou 3.03. Pre lokálne kritérium použijeme koeficienty výhodnosti, ktoré pre jednotlivé možné umiestnenia skladov určíme ako počty zákazníkov, pre ktorých je daný sklad najbližšie. Lokálne kritérium potom definujeme takto: „*Sklad, o ktorom chceme práve rozhodovať, či bude alebo nebude umiestnený, je sklad s najvyšším koeficientom výhodnosti z tých skladov, o ktorých sme doposiaľ nerozhodovali.*“ Koeficienty spolu s poradím výhodnosti uvádzame v tabuľke 3.06.

Tabuľka 3.06

Koeficienty výhodnosti pre umiestňovaciu úlohu

Sklady	1	2	3	4	5
Koeficienty výhodnosti	2	3	2	2	3
Poradie výhodnosti	3	1	4	5	2

Hľadanie riešenia pomocou heuristiky:

1. $I_v = (2, 5, 1, 3, 4)$. (Krok 1)
2. $I_1 = \{2\}$, $f^* = 450$, $p = 2$. (Krok 2)
3. $k = 5$, $f(\{2\}\cup\{5\}) = 382$, $f^* = 382$, $I_1 = \{2, 5\}$, $p = 3$. (Krok 3)
4. $k = 1$, $f(\{2, 5\}\cup\{1\}) = 395$, $f^* = 382$, $I_1 = \{2, 5\}$, $p = 4$. (Krok 3)
5. $k = 3$, $f(\{2, 5\}\cup\{3\}) = 382$, $f^* = 382$, $I_1 = \{2, 5\}$, $p = 5$. (Krok 3)
6. $k = 4$, $f(\{2, 5\}\cup\{4\}) = 395$, $f^* = 382$, $I_1 = \{2, 5\}$, $p = 6$. (Krok 3)
7. Koniec. $f(\{2, 5\}) = 382$, $I_1 = \{2, 5\}$, t. j. $\boldsymbol{y} = <0, 1, 0, 0, 1>$. Nenulové zložky matice $\boldsymbol{z}$: $z_{21} = z_{22} = z_{24} = z_{26} = z_{27} = z_{29} = 1$ (sklad 2 obslúži zákazníkov 1, 2, 4, 6, 7 a 9) a $z_{53} = z_{55} = z_{58} = z_{5,10} = 1$ (sklad 5 obslúži zákazníkov 3, 5, 8 a 10). (Krok 4)

Primárna výmenná heuristika pre umiestňovaciu úlohu so stratégiou „prvý vhodný“

Teraz navrhneme algoritmus, ktorý sa pokúsi vylepšiť zadané prípustné riešenie. V umiestňovacej úlohe budeme navzájom vymieňať umiestnený sklad za neumiestnený. Výmenu uskutočníme, ak sa pritom zníži hodnota účelovej funkcie. Ak sa nepodarí heuristikou vylepšiť riešenie, nie je to preto, žeby toto riešenie bolo optimálne.

1: Z prípustného riešenia umiestňovacej úlohy s aktuálnou hodnotou účelovej funkcie f^* vytvor množinu indexov umiestnených skladov U a množinu indexov neumiestnených skladov N. Z množín U a N vytvor zoznam S usporiadaných dvojíc (u, n) takých, že $u \in U$ a $n \in N$.

2: Ak je zoznam S prázdny, polož $I_1 = U$ (zoznam indexov umiestnených skladov aktuálneho riešenia) a prejdi na krok 4. V opačnom prípade vezmi dvojicu $(u, n) \in S$, polož $I_1 = (U - \{u\}) \cup \{n\}$, prepočítaj hodnotu účelovej funkcie $f(I_1)$ podľa vzťahu (3.09) a odstráň dvojicu (u, n) zo zoznamu S.

   Ak je $f^* > f(I_1)$, vykonaj výmenu skladov: $U = (U - \{u\}) \cup \{n\} = I_1$ a $N = (N - \{n\}) \cup \{u\}$ (t. j. $y_u = 0$ a $y_n = 1$), polož $f^* = f(I_1)$ a prejdi na krok 3. V opačnom prípade prejdi na krok 2.

3: Z množín U a N vytvor **nový** zoznam S usporiadaných dvojíc (u, n) takých, že $u \in U$ a $n \in N$ a pokračuj krokom 2.

4: Koniec algoritmu. Množina I_1 (aj U) obsahuje umiestnené sklady, f^* obsahuje hodnotu účelovej funkcie riešenia. Vytvorenie vektora umiestnení $\boldsymbol{y}$ a matice priradení $\boldsymbol{z}$ sa zhoduje s krokom 4 vkladacej heuristiky pre kapacitne neobmedzenú umiestňovaciu úlohu.

Primárna výmenná heuristika pre umiestňovaciu úlohu so stratégiou „najlepší vhodný“

1: Z prípustného riešenia umiestňovacej úlohy s aktuálnou hodnotou účelovej funkcie f^* vytvor množinu indexov umiestnených skladov aktuálneho riešenia U a množinu indexov neumiestnených skladov N. Z množín U a N vytvor zoznam S usporiadaných dvojíc (u, n) takých, že $u \in U$ a $n \in N$. Polož $f_b = f^*$.

2: Ak je zoznam S prázdny, prejdi na krok 3. V opačnom prípade vezmi dvojicu $(u, n) \in S$, polož $I_1 = (U - \{u\}) \cup \{n\}$, prepočítaj hodnotu účelovej funkcie $f(I_1)$ podľa vzťahu (3.09) a odstráň dvojicu (u, n) zo zoznamu S. Ak je $f_b > f(I_1)$, vykonaj $U_b = (U - \{u\}) \cup \{n\}$, $f_b = f(I_1)$, inak nevykonaj úpravy. Prejdi na krok 2.

3: Ak je $f^* \leq f_b$, polož $I_1 = U$ (zoznam indexov umiestnených skladov aktuálneho riešenia) a prejdi na krok 5. V opačnom prípade aktualizuj množinu umiestnených skladov U a hodnotu účelovej funkcie f^*, teda $U = U_b$ a $f^* = f_b$. Polož $N = I - U$. Pokračuj krokom 4.

4: Z množín U a N vytvor **nový** zoznam S usporiadaných dvojíc (u, n) takých, že $u \in U$ a $n \in N$, polož $f_b = f^*$ a pokračuj krokom 2.

5: Koniec algoritmu. Množina I_1 (aj U) obsahuje umiestnené sklady, f^* obsahuje hodnotu účelovej funkcie riešenia. Vytvorenie vektora umiestnení $\boldsymbol{y}$ a matice priradení $\boldsymbol{z}$ sa zhoduje s krokom 4 vkladacej heuristiky pre kapacitne neobmedzenú umiestňovaciu úlohu.

Výsledkom riešenia primárnej heuristiky pre umiestňovaciu úlohu zadanú tabuľkou 3.03 bolo nájdenie dvoch umiestnení skladov a to v miestach 2 a 5 ($I_1 = \{2, 5\}$) s hodnotou účelovej funkcie $f(I_1) = 382$. Ak toto riešenie použijeme ako východiskové riešenie pre výmennú heuristiku so stratégiou prvý vhodný aj so stratégiou najlepší vhodný, potom v oboch prípadoch získame nové riešenie $\boldsymbol{y} = \langle 1, 0, 1, 0, 0 \rangle$, t. j. $I_1 = \{1, 3\}$, s hodnotou účelovej funkcie $f(I_1) = 345$.

3.3.3 Heuristické metódy pre úlohy obchodného cestujúceho (Travelling salesman problem - TSP)

Medzi NP-ťažké kombinatorické úlohy zaraďujeme aj úlohu obchodného cestujúceho. Ide o úlohu na dopravnej sieti o n uzloch, kde obchodný cestujúci musí prejsť každým uzlom práve raz, pričom svoju cestu začína a končí v tom istom uzle dopravnej siete a dĺžka trasy má byť najkratšia zo všetkých trás. Z hľadiska teórie grafov ide o úlohu nájdenia najlacnejšej Hamiltonovskej kružnice v grafe, z hľadiska diskrétnej optimalizácie ide o vytvorenie čo najkratšej, resp. čo najlacnejšej okružnej jazdy automobilu, ktorým zásobujeme zákazníkov. V tom druhom prípade riešime úlohu na matici vzdialeností, čo v teórii grafov znamená, že hľadáme Hamiltonovskú kružnicu na úplnom grafe. Všetkých trás môže byť na úplnom grafe dopravnej sieti o n uzloch celkom až $(n-1)!$, vypočítať všetky ich dĺžky v reálnom čase nie je možné ani výkonným počítačom už pre malý počet uzlov. Štandardné optimalizačné algoritmy lineárneho programovania zas nedokážu vyriešiť rozsiahlejšie úlohy okružných jázd pre veľkosť matematického modelu. Preto aj pre riešenie úlohy obchodného cestujúceho používame najčastejšie heuristické algoritmy.

Úlohu obchodného cestujúceho v matematickom programovaní riešime na matici vzdialeností $D = \{d_{ij}\}$, v ktorej medzi vzdialenosťami platí trojuholníková nerovnosť t. j. pre každú navzájom rôznu trojicu uzlov i, j, k v dopravnej sieti platí, že $d_{ij} \leq d_{ik} + d_{kj}$. Aby sme úlohu obchodného cestujúceho mohli riešiť musí tiež platiť, že ľubovoľné dva uzly siete sú spojené cestou, čo znamená, že v matici vzdialeností sú všetky jej koeficienty vyjadrené nezáporným číslom rôznym od ∞. Namiesto matice vzdialeností môžeme úlohu riešiť aj na matici s koeficientami c_{ij}, $i = 1, ..., n$, $j = 1, ..., n$, ktoré vyjadrujú náklady na prejdenie úseku z uzla i do uzla j.

V tejto podkapitole budeme riešiť úlohu symetrického obchodného cestujúceho, t. j. v matici je $d_{ij} = d_{ji}$, resp. $c_{ij} = c_{ji}$, $i = 1, ..., n, j = 1, ..., n$.

Riešenie úlohy obchodného cestujúceho zapíšeme ako postupnosť uzlov, ktoré má obchodný cestujúci prejsť v uvedenom poradí. Keďže úloha je riešená na matici vzdialeností, vo vytvorenej trase sa uzly nesmú opakovať, samozrejme okrem prvého a posledného uzla, pretože obchodný cestujúci sa musí vrátiť do uzla, kde začal svoju trasu (hľadáme uzavretú Hamiltonovskú kružnicu).

Metóda najbližšieho suseda

Metóda najbližšieho suseda patrí medzi najjednoduchšie **duálne vkladacie heuristiky** pre tvorbu trasy obchodného cestujúceho. Podľa tejto metódy trasa obchodného cestujúceho začína v ľubovoľnom uzle dopravnej siete. Ako druhý v poradí zaradíme do trasy ten uzol z doposiaľ nezaradených uzlov, ktorý je najbližší k prvému uzlu. Postupne na koniec trasy pridáme vždy taký uzol z doposiaľ nezaradených uzlov, ktorý je najbližšie k poslednému uzlu súčasnej trasy. Nakoniec uzol, ktorý bol zaradený do trasy ako posledný, spojíme s prvým uzlom, čím vytvoríme trasu obchodného cestujúceho (okružnú jazdu automobilu). Zároveň s pridaním uzla do súčasnej trasy aktualizujeme aj dĺžku doposiaľ vytvorenej trasy. Podrobnejší popis algoritmu je v učebnici [13].

Tabuľka 3.07

Matica vzdialeností 10x10

Uzly	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	2	4	8	9	7	5	8	11	6
2	2	0	2	6	7	5	3	6	9	4
3	4	2	0	8	9	7	5	6	11	6
4	8	6	8	0	7	6	3	7	4	5
5	9	7	9	7	0	5	4	4	8	6
6	7	5	7	6	5	0	7	1	8	9
7	5	3	5	3	4	7	0	8	7	2
8	8	6	6	7	4	1	8	0	7	10
9	11	9	11	4	8	8	7	7	0	5
10	6	4	6	5	6	9	2	10	5	0

V tabuľke 3.07 je matica vzdialeností *D* pre 10 uzlov dopravnej siete. Ak túto maticu použijeme pre nájdenie trasy obchodného cestujúceho, kde prvý uzol v trase je uzol 5, potom jedno z možných riešení nájdených metódou najbližšieho suseda bude trasa tvorená uzlami 5-7-10-2-1-3-8-6-4-9-5 v uvedenom poradí. Dĺžka celej trasy je 41. Pretože tabuľka 3.07 obsahuje viacero rovnakých hodnôt, toto riešenie nemusí byť jediné. Ďalšie riešenie úlohy, ktoré začína v uzle 5, môže byť takéto: 5-8-6-2-1-3-7-10-4-9-5. Táto trasa má dĺžku 40.

Metóda najbližšieho suseda s viacnásobným opakovaním

Je zrejmé, že ak pri metóde najbližšieho suseda použijeme pre tvorbu trasy obchodného cestujúceho vždy iný východiskový uzol, môžeme získať inú trasu s inou dĺžkou. Metóda najbližšieho suseda nájde veľmi rýchlo trasu obchodného cestujúceho aj pre úlohy s veľkým počtom uzlov v dopravnej sieti. Môžeme preto túto metódu zopakovať niekoľkokrát pre rôzne východiskové uzly. Pre úlohy s malým počtom uzlom n môžeme heuristiku zopakovať až n-krát, vždy pre iný východiskový uzol. Zo všetkých riešení potom určíme ako hľadané riešenie to, ktorého dĺžka trasy bude najmenšia zo všetkých riešení. Ani v prípade, že metódou najbližšieho suseda použijeme až n-krát (vždy pre iný východiskový uzol), nie je zaručené, že nájdeme optimálne riešenie. Taktiež nevieme vyčísliť, o koľko je nájdené riešenie horšie od optimálneho riešenia úlohy obchodného cestujúceho.

Cvičenie 3.09

Metódou najbližšieho suseda s viacnásobným opakovaním riešte úlohu obchodného cestujúceho s maticou vzdialeností v tabuľke 3.07. Heuristiku opakujte pre východiskové uzly 1, 4, 7 a 8!

Frekvenčný algoritmus najbližšieho suseda

Algoritmus sa od algoritmu najbližšieho suseda líši tým, že namiesto matice ohodnotenia úsekov používa maticu frekvencií. Frekvencie pre vypočítajte podľa vzorca:

$$f_{ij} = n * c_{ij} - \sum_{k=1}^{n}(c_{ik} + c_{kj}) \qquad pre\ i = 1, ..., n, j = 1, ..., n \tag{3.10}$$

Algoritmus zväčšovania trasy o najbližší uzol

Algoritmus vychádza zo základnej neprípustnej trasy $i_1 - i_2 - i_3 - i_1$, ktorú v každom kroku zväčší vsunutím spracovávaného uzla medzi dva už zaradené uzly, ktoré nasledujú po sebe v súčasnej trase. Spracovávaný uzol sa vyberie z množiny doposiaľ nezaradených uzlov ako uzol, ktorý je najmenej vzdialený od už zaradených uzlov v trase (podľa súčtového kritéria – súčet vzdialeností kandidáta na spracovanie od každého uzla zaradeného v trase). Miesto, na ktoré je spracovávaný uzol zaradený, je určené tak, aby sa súčasná trasa zväčšila o čo najmenej.

Základnú neprípustnú trasu $i_1 - i_2 - i_3 - i_1$ môžeme učiť rôznymi spôsobmi. Prvým uzlom i_1 môže byť napríklad centrum, ďalší uzol i_2 môže byť napríklad doposiaľ nezaradený uzol, ktorý je najmenej vzdialený od uzla i_1 a i_3 bude doposiaľ nezaradený uzol, ktorý je najmenej vzdialený od uzla i_2.

Pre maticu vzdialeností v tabuľke 3.07 môže byť základná trasa tvorená vrcholmi $i_1 = 4$ (centrum), $i_2 = 7$ a $i_3 = 10$. Trasa 4-7-10-4 bude mať dĺžku 10.

Ako ďalší uzol do trasy zaradíme uzol 2, pretože súčet $d_{24}+d_{27}+d_{2,10} = 13$ je minimálny zo všetkých súčtov $d_{k,4}+d_{k,7}+d_{k,10}$, kde k je jeden z uzlov množiny {1, 2, 3, 5, 6, 8, 9}, ktoré zatiaľ neboli zaradené do trasy. Teraz musíme rozhodnúť, kam zaradíme uzol 2 v trase 4-7-10-4. Súčasná dĺžka trasy sa po zaradení uzla 2 do trasy musí zväčšiť o čo najmenšiu hodnotu. Ak by sme zaradili uzol 2 medzi uzly 4 a 7, zväčšila by sa pôvodná dĺžka trasy o hodnotu $d_{42}+d_{27}-d_{47} = 6 + 3 – 3 = 6$. Zaradenie uzla 2 medzi uzly 7 a 10 by predĺžilo pôvodnú trasu o 5 a medzi uzly 10 a 4 tiež o 5. Súčasná trasa po vložení uzla 2 môže byť 4-7-2-10-4 alebo 4-7-10-2-4. Tento postup opakujeme, kým nebudú do trasy zaradené všetky uzly.

Podobne môžeme navrhnúť ďalšie algoritmy pre nájdenie trasy obchodného cestujúceho. Môžu sa líšiť v spôsobe určenia základnej trasy $i_1 - i_2 - i_3 - i_1$, ale aj v spôsobe výberu uzla pre zaradenie do aktuálnej neprípustnej trasy. Musia však zohľadňovať kritérium pre čo najmenšie zväčšenie dĺžky aktuálnej trasy.

Cvičenie 3.10

Metódou zväčšovania o najbližší uzol dokončite vytvorenie trasy obchodného cestujúceho s maticou vzdialeností v tabuľke 3.07, ak súčasná neprípustná trasa je 4-7-2-10-4 s dĺžkou 15!

Pre jednoduchší popis manipulácie s uzlami v trase obchodného cestujúceho nazveme skupinu za sebou idúcich uzlov v trase **reťazcom**. Počet uzlov v reťazci udáva jeho **dĺžku**. Samostatný uzol môžeme nazvať reťazcom dĺžky 1. **Prázdnym** reťazcom nazveme reťazec s dĺžkou 0. Každý reťazec má najviac jeden začiatočný uzol Z a najviac jeden koncový uzol K. V reťazci dĺžky 1 sú uzly Z a K totožné. Každý reťazec v trase má práve jedného predchodcu P, čo je uzol, ktorý sa v trase nachádza tesne pred uzlom Z. Okrem toho má každý reťazec v trase práve jedného nasledovníka N, čo je uzol, ktorý v trase nasleduje tesne za uzlom K. V prázdnom reťazci po predchodcovi P nasleduje ihneď nasledovník N. Ak má reťazec dĺžku väčšiu ako 2, môžu sa medzi uzlami Z a K nachádzať ďalšie uzly. Ak pracujeme s viacerými reťazcami, môžeme k označeniu uzlov P, Z, K a N pridať kvôli rozlíšeniu indexy. Napríklad P_1-N_1 označuje prázdny reťazec s predchodcom P_1 a nasledovníkom N_1 a P_2-Z_2-K_2-N_2 označuje reťazec Z_2-K_2 dĺžky 2 s predchodcom P_2 a nasledovníkom N_2.

Algoritmus zväčšovania trasy vkladaním reťazca

Podobne, ako sme v predchádzajúcom algoritme zväčšovali trasu pridávaním uzlov, môžeme základnú neprípustnú trasu $i_1 - i_2 - i_3 - i_1$ postupne zväčšovať pridávaním reťazcov uzlov dĺžky 2 a viac. Keďže dĺžku trasy obchodného cestujúceho minimalizujeme, mali by vzdialenosti uzlov vo vkladanom reťazci byť čo najkratšie. Na výber uzlov do reťazca môžeme použiť podobné kritéria, ako pri výbere najbližšieho uzla v algoritme zväčšovania o najbližší uzol.

Uzly súčasnej trasy, medzi ktoré je spracovávaný reťazec zaradený, určíme tak, aby sa súčasná trasa zväčšila o čo najmenej. Nestačí však zisťovať vhodnosť zaradenia reťazca medzi dva uzly, musíme zistiť aj to, či nie je vhodnejšie vložiť do súčasnej trasy spracovávaný reťazec v inverznom (opačnom) poradím uzlov.

Cvičenie 3.11

Metódou zväčšovania trasy vkladaním reťazca určte, kam vložíte reťazec 4-7 do súčasnej neprípustnej trasy 8-6-2-8 s dĺžkou 12 v úlohe obchodného cestujúceho s maticou vzdialeností v tabuľke 3.07!

Výmenné heuristiky pre úlohu obchodného cestujúceho

Predchádzajúce heuristiky tejto podkapitoly vytvárali čo najkratšiu, resp. čo najlacnejšiu trasu. Teraz navrhneme prosté heuristiky, ktoré môžu vylepšiť trasu z hľadiska jej dĺžky.

Vo vytvorenej trase je možné určiť, či by zmenou poradia uzlov v trase došlo k skráteniu dĺžky trasy, resp. k zníženiu nákladov. Poradie uzlov v existujúcej trase môžeme meniť napríklad:

- inverziou reťazca dĺžky 2 a viac,
- presunutím niektorého reťazca na iné miesto v trase (typ výmeny: neprázdny reťazec za prázdny reťazec),
- vzájomnou výmenou polohy dvoch reťazcov v trase (typ výmeny: neprázdny reťazec za neprázdny reťazec).

Výmenu alebo inverziu reťazcov musíme vopred vyhodnotiť. V úlohách so symetrickou maticou vzdialeností pri výmene reťazcov dĺžky väčšej ako 2, vyhodnotíme vloženie reťazca v pôvodnom poradí uzlov, môžeme však vyhodnotiť aj vloženie reťazca v inverznom poradí uzlov a výmenu zrealizovať pre výhodnejší variant.

Pri prostých heuristikách uskutočníme výmenu alebo inverziu reťazcov iba v prípade, ak by to prinieslo úsporu účelovej funkcie, t. j. skrátenie dĺžky trasy, resp. zníženie nákladov na realizáciu trasy obchodného cestujúceho. Pri vyhodnocovaní zmeny účelovej funkcie berieme do úvahy iba dĺžky úsekov medzi tými uzlami, ktorých sa zmena bezprostredne dotýka. Ak napríklad trasa obchodného cestujúceho pozostáva zo 100 uzlov a vymieňame navzájom iba 2 uzly (dva neprázdne reťazce dĺžky 1), nemá zmysel prepočítavať celú dĺžku pôvodnej trasy a celú dĺžku novej trasy, čo predstavuje vlastne vyhľadanie 200 koeficientov v matici nákladov a ich sčítanie, resp. odčítanie. Zmenu celej dĺžky trasy ovplyvňujú iba úseky, ktoré spájajú predchodcov s začiatočnými uzlami reťazcov, a úseky, ktoré spájajú koncové uzly reťazcov a nasledovníkmi. Ostatné úseky trasy, a teda ani ostatné úseky spracovávaných reťazcov, nevplývajú na zmenu celkovej dĺžky trasy.

Predpokladajme, že chceme vyhodnotiť výmenu reťazcov Z_1-K_1 a Z_2-U_2-K_2 v trase obchodného cestujúceho a že nebudeme testovať vkladanie inverzných reťazcov. Aktuálna trasa pred výmenou reťazcov vyzerá takto: ...-P_1-Z_1-K_1-N_1-...-P_2-Z_2-U_2-K_2-N_2-..., kde vybodkované časti zastupujú ostatné uzly trasy. Po výmene zadaných reťazcov by trasa vyzerala takto: ...-P_1-Z_2-U_2-K_2-N_1-...-P_2-Z_1-K_1-N_2-..., pričom dĺžky úsekov medzi uzlami vo vybodkovaných častiach sa nemenia a nemenia sa ani dĺžky vnútorných úsekov reťazcov, t. j. dĺžky $d_{Z_1K_1}, d_{Z_2U_2}, d_{U_2K_2}$ sa nemenia, zmenila sa iba poloha uzlov zadaných reťazcov v trase. Preto **veľkosť úspory**, ktorá by vznikla po výmene reťazcov, určíme ako rozdiel súčtu dĺžok úsekov, ktoré zaniknú v pôvodnej trase, a súčtu dĺžok úsekov, ktoré pribudnú v novej trase. Úspora je teda vyjadrená vzťahom (3.11)

$$(d_{P_1Z_1} + d_{K_1N_1} + d_{P_2Z_2} + d_{K_2N_2}) - (d_{P_1Z_2} + d_{K_2N_1} + d_{P_2Z_1} + d_{K_1N_2}) \quad (3.11)$$

V prípade, že výraz (3.11) je kladný, jedná sa o úsporu, teda skrátenie dĺžky pôvodnej trasy, a výmenu reťazcov je potrebné vykonať.

Ak chceme vykonať inverziu reťazca v trase obchodného cestujúceho, vyjmeme reťazec z trasy a vložíme ho späť v obrátenom poradí uzlov. Úsporu opäť vypočítame z dĺžok zaniknutých a novovzniknutých úsekov. Pre reťazec Z-U_1-U_2-K, ktorého poloha vzhľadom k ostatným uzlov v trase je ...-P-Z-U_1-U_2-K-N-... a po inverzii je ...-P-K-U_2-U_1-Z-N-..., je úspora daná ako kladný rozdiel výrazu (3.12).

$$(d_{PZ} + d_{KN}) - (d_{PK} + d_{ZN}) \quad (3.12)$$

Princíp výmenných heuristík pre úlohu obchodného cestujúceho

Ako východiskové riešenie môžeme použiť riešenie získané niektorou vkladacou heuristikou. Na začiatku heuristiky zadefinujeme typ výmeny reťazcov resp. inverziu reťazca.

Výmena reťazcov dĺžky r_1 a r_2: Prvý reťazec dĺžky r_1 kandidujúci na výmenu je reťazec, ktorý začína v prvom uzle trasy a druhý reťazec dĺžky r_2 kandidujúci na výmenu je reťazec, ktorý začína v (r_1+1)-om uzle trasy. Ak by vznikla úspora výmenou týchto reťazcov, tak ju zrealizujeme, inak nie. Ďalšia dvojica kandidátov na výmenu je takáto: prvý reťazec ponecháme a druhý reťazec je reťazec dĺžky r_2, ktorý začína v (r_1+2)-om uzle trasy. Ak sa dostaneme na koniec trasy s druhým reťazcom, posunieme začiatok prvého reťazca do uzla 2 trasy a začiatok druhého reťazca do (r_1+1)-ho uzla, atď. Vždy, keď sa výmena uskutoční, vrátime sa na začiatok novej trasy a spustíme heuristiku od začiatku. Heuristiku ukončíme, ak po poslednom spustení heuristiky nedošlo k žiadnej výmene reťazcov.

Inverzia reťazca dĺžky r: ako prvý otestujeme reťazec, ktorý začína v prvom uzle trasy a má dĺžku r. Ak by vznikla úspora inverziou tohto reťazca, tak ju zrealizujeme, inak nie. Pokračujeme reťazcom dĺžky r, ktorého prvý uzol je v poradí druhým uzlom v trase, potom tretím uzlom atď. Vždy, keď sa inverzia uskutoční, mali by sme sa vrátiť na začiatok trasy obchodného cestujúceho a spustiť heuristiku od začiatku, úplne však postačí, ak sa vrátime o r-1 uzlov naspäť a tu spustíme opäť heuristiku. Heuristiku ukončíme v uzle trasy, z ktorého už nemôžeme vytvoriť reťazec dĺžky r.

3.3.4 Heuristické metódy pre dopravnú úlohu

Dopravnú úlohu môžeme riešiť všeobecnými primárnymi aj duálnymi exaktnými algoritmami, ktoré sú určené pre riešenie úloh lineárneho programovania (pozri podkapitolu 2.1.3 tejto učebnice).

Na tomto mieste popíšeme heuristiky pre nájdenie suboptimálneho riešenia (síce dobrého, ale bez záruky optimality). Budú to tieto metódy:

1. Indexová metóda, resp. metóda minimálneho prvku.
2. Vogelova aproximačná metóda (VAM).
3. Frekvenčná metóda.

Riešenia dopravnej úlohy, nájdené uvedenými heuristikami, môžeme použiť ako základné východiskové riešenia pre exaktné algoritmy, špeciálne vytvorené pre riešenie dopravnej úlohy [4][12][32].

V dopravnej úlohe prepravujeme rovnaký druh tovaru medzi m skladmi a n spotrebiteľmi. Sklady majú postupne kapacity a_1, a_2, ..., a_m a spotrebitelia majú postupne požiadavky b_1, b_2, ..., b_n. Dopravná úloha je **vyvážená (vybilancovaná)**, ak $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$ (ponuka sa rovná dopytu). Jednotkové prepravné náklady sú dané koeficientmi c_{ij} pre i=1, 2, ..., m a j=1, 2, ..., n. Koeficienty c_{ij} pre dopravnú úlohu zapisujeme do tabuľky nákladov $\boldsymbol{c}$ s m riadkami a n stĺpcami. Tabuľka nákladov zodpovedá matici nákladov.

Ak rozhodnutie o prepravovanom množstve tovaru medzi skladom i a spotrebiteľom j označíme x_{ij} pre i=1, 2, ..., m a j=1, 2, ..., n, potom maximálne prepravované množstvo medzi i-tym skladom a j-tym spotrebiteľom je rovné $min\{a_i, b_j\}$, teda $x_{ij}\in$<0, $min\{a_i, b_j\}$> pre i=1, 2, ..., m a j=1, 2, ..., n. Celkový počet rozhodovacích premenných (rozmer úlohy) je m x n.

Tabuľka 3.08

Dopravná úloha – koeficienty c_{ij}, a_i, b_j pre i=1, 2, 3 a j=1, 2, 3, 4

c_{ij}	1	2	3	4	a_i
1	8	6	5	9	10
2	4	3	8	5	14
3	10	5	9	6	12
b_j	11	9	6	10	

V tabuľke 3.08 sú zadané koeficienty vyváženej dopravnej úlohy. Aj bez dôkazu je čitateľovi určite jasné, že vyvážená dopravná úloha má nekonečne veľa riešení.

Tabuľka 3.09

Dopravná úloha – matica riešení *x*

x_{ij}	1	2	3	4	a_i
1	6			4	10
2	5	7	2		14
3		2	4	6	12
b_j	11	9	6	10	

Tabuľka 3.09 obsahuje jedno z možných prípustných riešení úlohy zadanej tabuľkou 3.08. Políčko v i-tom riadku a j-tom stĺpci tabuľky s riešením $\boldsymbol{x}$ udáva prepravované množstvo tovaru medzi i-tým skladom a j-tým spotrebiteľom. Prázdne políčka zodpovedajú prepravovanému množstvu rovnému 0. Riešenie v tabuľke 3.09 je prípustné, pretože riadkové súčty hodnôt matice $\boldsymbol{x}$ sa rovnajú kapacitám skladov a stĺpcové súčty hodnôt matice $\boldsymbol{x}$ sa rovnajú požiadavkám spotrebiteľov.

Nevyváženú dopravnú úlohu pred heuristikou upravíme na vyváženú.

V prípade, že $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ (ponuka tovaru prevyšuje dopyt po ňom), odstránime nevyváženosť úlohy pridaním fiktívneho spotrebiteľa s indexom n+1. Jeho požiadavka bude $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$ a náklady $c_{i,n+1} = 0$ pre $i = 1, 2, ..., m$.

Ak dopyt po tovare prevyšuje ponuku, teda $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$, potom do úlohy pridáme fiktívny sklad s indexom m+1. Kapacitu fiktívneho skladu určíme ako $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{nm} a_{ji}$ a náklady $c_{m+1,j} = 0$ pre $j = 1, 2, ..., n$.

Cvičenie 3.12

V nevyváženej dopravnej úlohe s piatimi skladmi a s desiatimi spotrebiteľmi je $\sum_{i=1}^{5} a_i > \sum_{j=1}^{10} b_j$. Fiktívnemu spotrebiteľovi patrí index $j = 11$. Čo vieme povedať o riešení, v ktorom $x_{3,11} = 7$?

Cvičenie 3.13

V nevyváženej dopravnej úlohe s piatimi skladmi a s desiatimi spotrebiteľmi je $\sum_{i=1}^{5} a_i < \sum_{j=1}^{10} b_j$. Fiktívnemu skladu patrí index $j = 6$. Čo vieme povedať o riešení, v ktorom $x_{6,8} = 10$?

Princíp heuristík pre riešenie dopravnej úlohy

Postupné vytváranie riešenia v tabuľke $\boldsymbol{X}$ je rovnaké pri všetkých troch metódach. Jediný rozdiel je v určení, ktoré políčko tabuľky bude v nasledujúcom kroku vyplnené hodnotou $x_{ij} = min\{a_i,\ b_j\}$, teda metódy sa navzájom líšia lokálnym kritériom pre výber políčka tabuľky s riešením. Uvedieme preto teraz princíp heuristiky pre všetky metódy a po ňom vysvetlíme lokálne kritéria pre jednotlivé metódy a vyriešime danou metódou dopravnú úlohu zadanú tabuľkou 3.08.

1. Heuristiku aplikujeme na vyváženú dopravnú úlohu, zadanú tabuľkou, v ktorej sú uvedené kapacity skladov (a_i, i=1, 2, ..., m), požiadavky spotrebiteľov (b_j, j=1, 2, ..., n) a náklady na prepravu jednej jednotky tovaru medzi skladom i a spotrebiteľom j (c_{ij}, i=1, 2, ..., m, j=1, 2, ..., n). Hľadané riešenie zapisujeme do tabuľky s označením $\boldsymbol{X}$, tabuľka je na začiatku heuristiky prázdna (t. j. všetky premenné majú hodnotu 0).
2. Na základe lokálneho kritéria určíme políčko tabuľky $\boldsymbol{X}$ v r-tom riadku a s-tom stĺpci. Do políčka zapíšeme hodnotu $x_{rs} = min\{a_r,\ b_s\}$.
3. Určíme nové hodnoty $a_r = a_r - x_{rs}$ a $b_s = b_s - x_{rs}$.

   Ak $a_r = 0$ (r-ty sklad je prázdny), potom vyškrtneme (alebo inak označíme) r-ty riadok tabuľky, v ktorej je zadanie úlohy, aby sme v ďalších krokoch už tento riadok nebrali do úvahy. Ak $b_s = 0$ (s-ty spotrebiteľ už má splnenú svoju požiadavku), potom vyškrtneme s-ty stĺpec tabuľky so zadaním úlohy a v ďalších krokoch už tento stĺpec neberieme do úvahy. V prípade, že po úprave budú a_r aj b_s súčasne rovné 0, môžeme z tabuľky vyškrtnúť aj r-ty riadok aj s-ty stĺpec.

   Pri metóde VAM po vyškrtnutí riadka alebo stĺpca matice nákladov musíme prepočítať stĺpcové alebo riadkové diferencie.
4. Body 2 a 3 opakujeme, kým nie sú vyčerpané všetky kapacity skladov a uspokojené všetky požiadavky spotrebiteľov, t. j. na konci sú všetky koeficienty a_i, $i = 1, 2, ..., m$ a b_j, $j = 1, 2, ..., n$ rovné 0.

Indexová metóda (metóda minimálneho prvku)

Pri voľbe premennej, ktorej chceme priradiť hodnotu, budeme používať pre túto metódu nasledujúce lokálne kritérium:

„V nevyškrtanej časti tabuľky nákladových koeficientov zvoľ políčko v r-tom riadku a v s-tom stĺpci s minimálnou hodnotou nákladov c_{ij}, $i = 1, 2, ..., m$ a b_j, $j = 1, 2, ..., n$ a do zodpovedajúceho políčka tabuľky riešenia $\boldsymbol{x}$ zapíš maximálnu možnú hodnotu x_{rs}, t. j. $min\{a_r, b_s\}$".

Začiatok postupu riešenia indexovou metódou dopravnej úlohy zadanej tabuľkou 3.08 je v tabuľke 3.10. Vyškrtnutý stĺpec je podfarbený sivou farbou.

Tabuľka 3.10

Dopravná úloha – prvé rozhodnutie x_{22} nájdené indexovou metódou

x_{ij}	1	2	3	4	a_i
1					10
2		9			5
3					12
b_j	11	0	6	10	

c_{ij}	1	2	3	4	a_i
1	8	6	5	9	10
2	4	3	8	5	14
3	10	5	9	6	12
b_j	11	9	6	10	

V tabuľke 3.10 je ako prvé vyplnené políčko zodpovedajúce premennej x_{22}, pretože spomedzi všetkých nákladových koeficientov má koeficient c_{22} najmenšiu hodnotu. $x_{22} = min\{a_2, b_2\} = min\{14, 9\} = 9$. Spotrebiteľ 2 je po úprave uspokojený, t. j. $b_2 = 0$ ($b_2 = b_2 - x_{22} = 9-9 = 0$) a sklad 2 ponúka ešte 5 jednotiek tovaru ($a_2 = 14-9$). V tabuľke je z ďalších úvah vyškrtnutý stĺpec 2, pretože po úprave zostala požiadavka druhého spotrebiteľa nulová a to zodpovedá druhému stĺpcu.

Ďalšou minimálnou hodnotou v nevyškrtanej časti tabuľky nákladov je číslo 4 v druhom riadku a prvom stĺpci, a teda bázickou premennou bude x_{21}. Po úpravách bude $x_{21} = 5$, $a_2 = 0$, $b_1 = 6$ a vyškrtnutý bude druhý riadok v matici nákladov $\boldsymbol{c}$. Ďalšie políčka tabuľky riešení budeme vyberať v poradí zodpovedajúcom poradiu premenných x_{13}, x_{34}, x_{11} a x_{31}. Prípustné riešenie dopravnej úlohy nájdené indexovou metódou je: $x_{11} = 4$, $x_{13} = 6$,

$x_{21} = 5, x_{22} = 9, x_{31} = 2$ a $x_{34} = 10$.

Vogelova aproximačná metóda (VAM)

Táto metóda pri výbere políčka x_{ij} zvýhodňuje ten minimálny koeficient c_{ij} v matici nákladov $\boldsymbol{c}$, ktorého rozdiel od druhého najmenšieho koeficientu v riadku i alebo v stĺpci j je čo najväčší. Pre každý riadok preto vypočítame rozdiel (diferenciu) druhého najmenšieho a najmenšieho koeficientu c_{ij} a to isté urobíme aj pre každý stĺpec. Diferencie sa menia po každom škrtaní v matici nákladov. Ak škrtneme riadok v matici nákladov, musíme prepočítať diferencie tých stĺpcov, ktoré ešte nie sú vylúčené z tabuľky. V prípade, že škrtneme stĺpec v matici nákladov, je nutné prepočítať riadkové diferencie nevyškrtnutých riadkov.

Lokálne kritérium pre túto metódu:

„V nevyškrtanej časti tabuľky nákladových koeficientov zvoľ políčko v r-tom riadku a v s-tom stĺpci s minimálnou hodnotou nákladov c_{ij} toho riadka alebo toho stĺpca, ktorý má najväčšiu diferenciu dvoch najmenších koeficientov a do zodpovedajúceho políčka tabuľky riešenia $\boldsymbol{x}$ zapíš maximálnu možnú hodnotu x_{rs}, t. j. $min\{a_r, b_s\}$. V prípade, že má viacero riadkov alebo stĺpcov rovnakú najväčšiu diferenciu, vyberieme políčko s ich spoločným minimom.“

Tabuľka 3.11 zobrazuje výpočet diferencií Vogelovej aproximačnej metódy pre riešenie dopravnej úlohy zadanej tabuľkou 3.08. Prvé tri úpravy riešenia sú zobrazené v tabuľkách 3.12 až 3.14. Pretože táto metóda pracuje s riadkovými a stĺpcovými diferenciami, nahradili sme v tabuľkách pôvodné hodnoty koeficientov a_i a b_j práve týmito diferenciami. Stĺpcové diferencie sú uvedené v riadku s označením dif_j a riadkové diferencie sú uvedené v stĺpci s označením dif_i. Diferencie vyškrtnutých riadkov alebo stĺpcov neuvádzame. Vyškrtnuté riadky a stĺpce sú v tabuľkách podfarbené sivou farbou.

Tabuľka 3.11

Dopravná úloha – výpočet diferencií pre metódou VAM

x_{ij}	1	2	3	4	a_i
1					10
2					14
3					12
b_j	11	9	6	10	

c_{ij}	1	2	3	4	dif_i
1	8	6	5	9	1
2	4	3	8	5	1
3	10	5	9	6	1
dif_j	4	2	3	1	

Tabuľka 3.12

Dopravná úloha – prvé rozhodnutie x_{21} nájdené metódou VAM

x_{ij}	1	2	3	4	a_i
1					10
2	11				3
3					12
b_j	0	9	6	10	

c_{ij}	1	2	3	4	dif_i
1	8	6	5	9	1
2	4	3	8	5	2
3	10	5	9	6	1
dif_j		2	3	1	

Tabuľka 3.13

Dopravná úloha – druhé rozhodnutie x_{13} nájdené metódou VAM

x_{ij}	1	2	3	4	a_i
1			6		4
2	11				3
3					12
b_j	0	9	0	10	

c_{ij}	1	2	3	4	dif_i
1	8	6	5	9	3
2	4	3	8	5	2
3	10	5	9	6	1
dif_j		2		1	

Tabuľka 3.18

Dopravná úloha – tretie rozhodnutie x_{12} nájdené metódou VAM

x_{ij}	1	2	3	4	a_i
1		4	6		0
2	11				3
3					12
b_j	0	5	0	10	

c_{ij}	1	2	3	4	dif_i
1	8	6	5	9	
2	4	3	8	5	2
3	10	5	9	6	1
dif_j		2		1	

Ďalšie políčka tabuľky riešení budeme vyberať v poradí zodpovedajúcom poradiu premenných x_{22}, x_{32} a x_{34}. Prípustné riešenie dopravnej úlohy nájdené Vogelovou metódou je potom: $x_{12} = 4$, $x_{13} = 6$, $x_{21} = 11$, $x_{22} = 3$, $x_{32} = 2$ a $x_{34} = 10$.

Frekvenčná metóda

Táto metóda je zhodná s indexovou metódou. Namiesto matice nákladov s koeficientmi c_{ij} pre i=1, 2, ..., m a j=1, 2, ..., n však používa frekvenčnú maticu (tabuľku) s koeficientmi f_{ij} pre i=1, 2, ..., m a j=1, 2, ..., n. Koeficient f_{ij} berie do úvahy nielen zodpovedajúci koeficient c_{ij}, ale aj jeho odchýlku od všetkých koeficientov v riadku i a v stĺpci j matice nákladov. Koeficienty frekvencií f_{ij} pre i =1, 2, ..., m a j = 1,, 2, ..., n vypočítame podľa vzťahu:

$$f_{ij} = (m+n) * c_{ij} - \sum_{l=1}^{n} c_{il} - \sum_{k=1}^{m} c_{kj} \tag{3.11}$$

Čím menší je koeficient f_{ij} v tabuľke frekvencií, tým viac je v tabuľke nákladov väčších koeficientov v okolí zodpovedajúceho koeficientu c_{ij} (v i-tom riadku a j-tom stĺpci). Preto uprednostňujeme v tabuľke nastaviť hodnotu tej premennej x_{ij}, ktorá zodpovedá najmenšiemu koeficientu f_{ij} z tých koeficientov v tabuľke frekvencií, o ktorých ešte môžeme rozhodovať. Keby sme tak neurobili, museli by sme pre určenie hodnoty premennej vybrať políčko, ktoré zodpovedá vyššiemu nákladovému koeficientu ako je uvažované c_{ij}, čo by mohlo zvýšiť hodnotu výslednej účelovej funkcie. Tabuľka (matica) frekvencií je vlastne tabuľka koeficientov výhodnosti – je nevýhodné nevybrať najmenší koeficient f_{ij}.

Lokálne kritérium pre túto metódu je preto takéto:

„V nevyškrtanej časti frekvenčnej tabuľky zvoľ políčko v r-tom riadku a v s-tom stĺpci s minimálnou hodnotou f_{ij} a do zodpovedajúceho políčka tabuľky riešenia $\boldsymbol{x}$ zapíš hodnotu $x_{rs} = \min\{a_r, b_s\}$".

V pravej časti tabuľky 3.15 sa namiesto koeficientov c_{ij} nachádzajú koeficienty frekvenčnej matice $\boldsymbol{f}$, ktoré boli vypočítané podľa vzťahu (3.11). Ako prvý najmenší koeficient bol určený koeficient f_{13}. Ďalší postup je zhodný s indexovou metódou.

Tabuľka 3.15

Dopravná úloha – prvé rozhodnutie x_{13} dané frekvenčnou metódou

x_{ij}	1	2	3	4	a_i
1			6		4
2					14
3					12
b_j	11	9	0	10	

c_{ij}	1	2	3	4	a_i
1	6	0	-15	15	10
2	-14	-13	14	-5	14
3	18	-9	11	-8	12
b_j	11	9	6	10	

Prípustné riešenie dopravnej úlohy nájdené frekvenčnou metódou je potom: $x_{13} = 6$, $x_{14} = 4$, $x_{21} = 11$, $x_{22} = 3$, $x_{32} = 6$ a $x_{34} = 6$.

Cvičenie 3.13

V tabuľke 3.16 je zadaná nevyvážená dopravná úloha. Nájdite prípustné riešenie úlohy všetkými tromi heuristikami z tejto podkapitoly!

Tabuľka 3.16

Dopravná úloha – koeficienty c_{ij}, a_i, b_j pre i=1, 2, 3 a j=1, 2, 3, 4

c_{ij}	1	2	3	4	a_i
1	6	8	10	7	50
2	11	8	5	8	70
3	7	6	16	10	30
b_j	60	20	50	60	

Cvičenie 3.14

Upravte kapacity skladov v tabuľke 3.16 na hodnoty $a_1 = 80$, $a_2 = 70$, $a_3 = 60$ a nájdite prípustne riešenie upravenej dopravnej úlohy všetkými tromi heuristikami z tejto podkapitoly!