EMM1 – teoretické otázky

Vážení kolegové, v tomto materiálu najdete teoretické otázky, na základě kterých vysvětluji teorii požadovanou na zkoušce z EMM1 na ČZU. U většiny otázek je také uvedena stránka z materiálu „KCKurzy – Jak udělat zkoušku z EMM1“, která souvisí s danou otázkou. Zadání otázky je vždy modře, správné řešení níže černě. Další dobré materiály ke zkoušce můžete najít na Facebooku **EMM 1 ČZU.** Přihlásit na doučování se můžete na **www.kckurzy.cz.**

Přeji příjemné učení☺. RNDr. Marian Rybář – lektor EMM1

Téma 1: Úvod do EMM. Modely lineárního programování (LP), prostor řešení (str. 2-7)

1) Co je to modelování? Proveďte klasifikaci modelů podle alespoň jednoho hlediska. Ke každému typu modelů uveďte příklad.

Modelování je způsob zkoumání reality, při němž složitost, chování a další vlastnosti nějakého reálného celku (systému) vyjadřujeme složitostí, chováním a vlastnostmi jiného celku – modelu

Model je záměrně zjednodušený obraz skutečnosti vytvořený pomocí zvolených zobrazovacích prostředků – Např. Model lineárního programování pro pěstování plodin A, B, C,.. je tvořen omezujícími podmínkami, účelovou funkcí a podmínkami nezápornosti (zobrazovací prostředky) čímž nahrazujeme nějaký reálný celek (systém).

Klasifikace modelů z hlediska reprezentace:

1) **Ikonické modely** – jsou fyzikální (hmatatelnou) replikou nějakého reálného systému
Např: modely strojů, modely staveb (mohou být zmenšené, zvětšené i ve stejném měřítku)
2) **Symbolické modely** – jsou abstrakcí reálného systému (mohou být vyjádřené například graficky, slovně, rovnicemi)
Např. matematické modely lineárního programování – popisují nějaký konkrétní systém omezujícími podmínkami, účelovou funkcí a podmínkami nezápornosti – např. model pěstování plodin A, B, C

2) Co je to systém? Jaký je jeho význam v procesu systémového modelování?

Systém je nějaký reálný celek složený z neprázdné skupiny **prvků, vazeb** mezi nimi a **kritéria systému**.

Součástí systému jsou:
Prvky systému – např. plodiny A, B, C
Vazby systému – např. součet ploch plodin A, B, C nesmí překročit 10 ha
Kritérium systému - např. maximalizujeme zisk z prodeje plodin

3) Uveďte podstatu a význam (možnosti aplikace) modelů lineárního programování

Podstatou modelů lineárního programování je nalézt vhodnou kombinaci proměnných (např. plodin A,B,C), která vyhovuje daným **lineárním** omezujícím podmínkám a maximalizuje či minimalizuje **lineární** účelovou funkci. Poznámka pro pochopení: **Lineární** funkce je vždy tvořena koeficienty (čísly) vynásobenými proměnnými x_i v **lineární podobě** (např. $z_{max} = 2.x_1+5.x_2+4.x_3$), v modelech LP se tedy nesetkáváme s mocninami, odmocninami apod (viz vysvětlení na doučování).

Význam modelů LP (možnosti aplikace):

- Maximalizace zisku při pěstování plodin (Úloha o výrobním plánu)
- Minimalizace nákladů při míchání směsí (Směšovací úloha)
- Maximalizace zisku při nařezání tyčí (Úloha o řezném plánu)
- Minimalizace nákladů na pracovníky (Úloha o plánování směn)
- další úlohy viz doučování

4) Uveďte a stručně popište komponenty modelů lineárního programování (str. 2)

a) Proměnné => Musíme vždy uvádět jejich ***jednotky***

- Značí se xi (Např. x1 = plocha plodiny A, x2 = plocha plodiny B,…)
- Zachycují ***počet realizací*** daného procesu (Neboli česky: Kolik budu ve výsledku pěstovat dané plodiny, abych měl maximální zisk)

b) Omezující podmínky

- Vymezují přípustné kombinace hodnot proměnných (Např. Můžeme pěstovat maximálně 14 ha plodin, tedy x1+x2 ≤ 14)
- ***Základní typy*** omezujících podmínek:
  - *kapacitní **„ ≤ “***
  - *požadavkové **„ ≥ “***
  - *určení **„ = “***

c) Účelová funkce

- Je vyjádřena jako násobky jednotkových cen proměnných ci a jejich hodnot xi (Např. Z_{MAX}=3.x1+2.x2)
- Základní typy účelových funkcí dle kritéria, kterého chceme v modelu dosáhnout
  - Minimalizační => $\mathbf{Z_{MIN}}$
  - Maximalizační => $\mathbf{Z_{MAX}}$

d) Podmínky nezápornosti

- Požadujeme pro ***všechny proměnné*** *hodnotu* ***≥ 0***
- Zajišťují praktickou aplikovatelnost a reálnost řešení (Neboli česky: Nemůžeme pěstovat -3 hektary žita☺)

5) Uveďte a stručně charakterizujte dva základní způsoby grafického řešení modelů lineárního programování. Za jakých podmínek je možné je použít? (str. 2-7)

a) Prostor řešení (str. 2-4)

- tento způsob používáme v modelech, které mají nejvýše 2 proměnné x1, x2 a neomezený počet omezujících podmínek. Osy souřadnic jsou tvořeny proměnnými x1, x2.

Postup řešení (podrobně str. 2-4):

Krok 1: Označím si neznámé proměnné

Krok 2: Slovní zadání přepíšu do rovnic a nerovnic (formulace modelu lineárního programování)

Krok 3: a) Zakreslím graficky množinu omezujících podmínek do **prostoru řešení** (vznikne tzv. konvexní polyedr = simplex)

b) Zakreslím jinou barvou graf účelové funkce (přímku) pro nějaký libovolný zisk. Posunu rovnoběžku s touto „pomocnou“ účelovou funkcí až do bodu, kde se co nejdále od počátku „vpravo a nahoře“ (v případě maximalizace) dotkne konvexního polyedru. V tomto místě je optimální bod, který mi určí optimální kombinaci proměnných x1 a x2, při které bude maximální zisk.

Krok 4: Určím souřadnice optimálního bodu pomocí pravítka nebo výpočtem průsečíku dvou funkcí, jejichž je průsečíkem

Krok 5: Dosadíme výsledné optimální hodnoty x1 a x2 do účelové funkce a dopočítáme výsledný maximální zisk

Krok 6: Slovní odpověď

b) Prostor požadavků (str. 5-7)

- tento způsob používáme v modelech, které mají neomezený počet proměnných x1, x2, x3, x4, …, ale pouze 2 omezující podmínky (požadavky). Osy souřadnic jsou tvořeny dvěma požadavky (Požadavek 1, Požadavek 2)

Postup řešení (podrobně str. 5-7):

Krok 1: Označím si neznámé proměnné

Krok 2: Slovní zadání přepíšu do rovnic a nerovnic (formulace modelu lineárního programování)

Krok 3: Pomocí tzv. **doplňkových proměnných d1, d2** převedeme nerovnice na rovnice

- kapacitní podmínky (≤) - přičteme hodnotu doplňkové proměnné k levé straně (+d)
- požadavkové podmínky (≥) - od levé strany hodnotu doplňkové proměnné odečteme (-d)
- podmínky určení (=) - žádná transformace není potřeba
- v účelové funkci dáme před všechny doplňkové proměnné 0

Krok 4: Vydělíme koeficienty v obou rovnicích omezujících podmínek cenami z účelové funkce a získáme pro každou proměnnou x1, x2, x3 a x4 jeden vektor a1, a2, a3 a a4 (např. pro proměnnou x1 vektor a1 atd). Ukázka viz doučování.

Krok 5: Zakreslíme výše vypočítané vektory a1, a2, a3, a4,.., d1, d2,.. a vektor pravých stran b do **prostoru požadavků.** V případě **minimalizace** hledáme spojnici mezi konci vektorů, která „nejdále vpravo nahoře“ protne vektor b. V případě **maximalizace** hledáme spojnici mezi konci vektorů, která „nejblíže vlevo dole“ protne vektor b. Viz ukázka doučování.

Krok 6: V rovnicích omezujících podmínek z Kroku 3 necháme pouze 2 výsledné nejvýhodnější proměnné z Kroku 5 a vyřešíme vzniklou soustavu 2 rovnic o dvou neznámých:

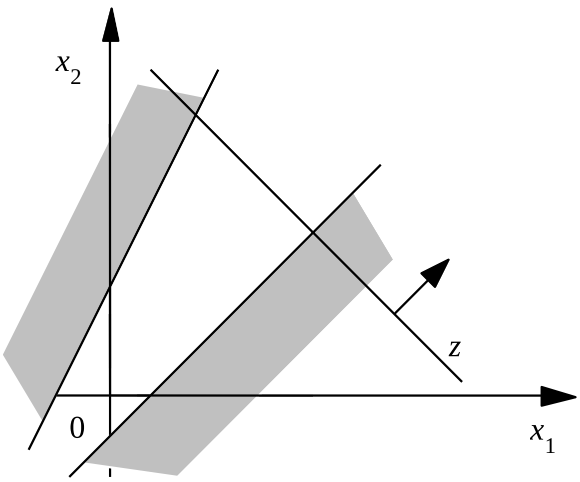
Krok 7: Dosadíme výsledné optimální hodnoty dvou proměnných do účelové funkce a dopočítáme její výslednou hodnotu

Krok 8: Slovní odpověď

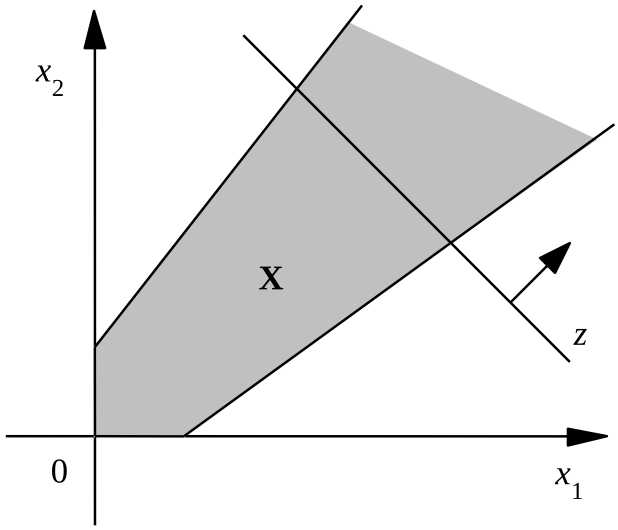
6) Uveďte 4 možné výsledky řešení modelů lineárního programování a znázorněte je graficky v prostoru řešení

1) *Model nemá žádné přípustné řešení*
2) *Model má přípustné řešení, ale hodnota účelové funkce může neomezeně růst nebo klesat*
3) *Model má právě jedno optimální řešení*
4) *Model má nekonečně mnoho optimálních řešení*

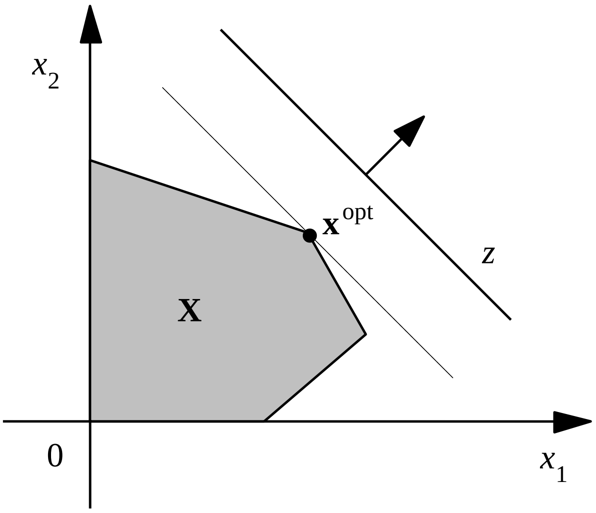
Obr1 – *Model nemá žádné přípustné řešení*



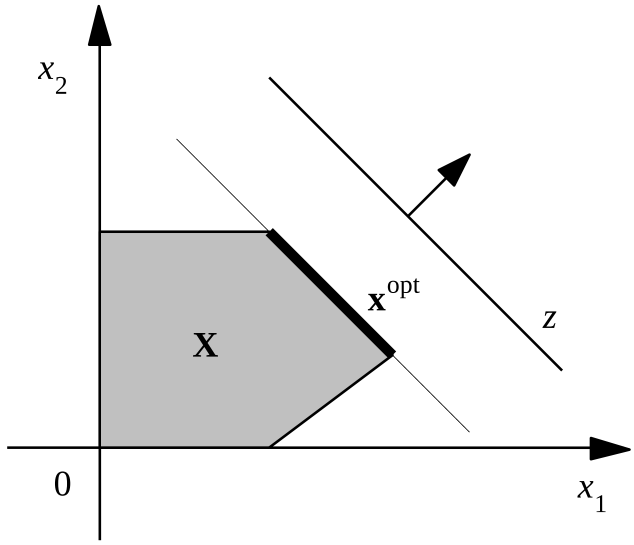
Obr2 – *Model má přípustné řešení, ale hodnota účelové funkce může neomezeně růst nebo klesat*



Obr3 – *Model má právě jedno optimální řešení*



Obr4 – *Model má nekonečně mnoho optimálních řešení*



Téma 2: Grafické řešení modelu LP v prostoru požadavků. Bázická a nebázická řešení (str. 2-7)

1) Uveďte a stručně komentujte základní vlastnosti modelů lineárního programování

- **Linearita** – pro sestavení účelové funkce i omezujících podmínek používáme pouze lineární funkce
- **Determinističnost** (nenáhodnost) – např. potřebný počet hodin na obdělání 1 ha pšenice x1 je deterministický (nenáhodný - přesně určený) na rozdíl od stochastického přístupu, kdy by byl odhadovaný pouze s určitou pravděpodobností
- **Statičnost** (statický charakter) – podmínky modelu jsou neměnné (statické) v čase
- **Sčitatelnost** (aditivita) – při sestavování účelové funkce i omezujících podmínek je nutno mít stejné jednotky u všech členů i na pravé straně rovnice nebo nerovnice (např. nemůžu sčítat v jedné rovnici hektary pole a hodiny práce)
- **Spojitost** – funkce použité při sestavování účelové funkce i omezujících podmínek musí být spojité (např. používají se spojité lineární funkce a nepoužívá se nespojitá funkce typu tg atd)
- **Libovolná dělitelnost** účelové funkce a omezujících podmínek nějakou konstantou
- **Neomezená záměna faktorů** – nezáleží na pořadí faktorů

2) Charakterizujte pojmy: „přípustné řešení“, „optimální řešení“, „alternativní řešení“, „suboptimální řešení“ v kontextu modelů lineárního programování.

Přípustné řešení = jakékoli řešení, které **splňuje všechny omezující podmínky** (ale nemusí být tím nejlepším řešením)
Např. V prostoru řešení – konvexní polyedr (simplex) jako průnik polorovin omezujících podmínek
Např. V prostoru požadavků – spojnice vektorů ai a aj, které protnou vektor pravých stran b

Optimální řešení = přípustné řešení, které maximalizuje nebo minimalizuje účelovou funkci.
Např. V prostoru řešení – **nejvzdálenější** průsečík rovnoběžky s účelovou funkcí a konvexního polyedru (simplexu) od počátku v případě maximalizace
Např. V prostoru požadavků – **nejvzdálenější** spojnice vektorů ai a aj s vektorem pravých stran b v případě minimalizace

Alternativní řešení = je to řešení, které má hodnotu účelové funkce **stejnou jako optimální řešení.**

Suboptimální řešení = je to řešení, které má hodnotu účelové funkce **horší než optimální řešení.**

3) Co je to bázické a nebázické řešení modelu lineárního programování? Jak se bázické řešení reprezentuje graficky?

Bázické (základní) řešení - řešení, kdy všechny nebázické proměnné jsou rovny 0 a hodnoty bazických proměnných jsou určeny ze soustavy jako konkrétní čísla. Podstatou řešení úloh LP je, že optimální řešení hledáme právě v podobě bazického řešení, což nám umožňují „základní věty lineárního programování“ z otázky 5 níže.

Nebázické řešení - řešení, kdy se za nebázické proměnné položí určité hodnoty a získají se konkrétní hodnoty i pro bazické proměnné. Neplatí tedy, že všechny nebázické proměnné jsou rovny 0.

Bázické (tedy i optimální) řešení **v prostoru řešení** je reprezentováno pomocí průsečíku konvexního polyedru (množiny přípustných řešení) a rovnoběžky s účelovou funkcí.

Bázické (tedy i optimální) řešení **v prostoru požadavků** je reprezentováno dvojicí proměnných ai a aj, pomocí jejichž spojnice dokážeme protnout vektor požadavků b v případě maximalizace co nejvíce vlevo dole, v případě minimalizace co nejvíce vpravo nahoře.

4) Co je to degenerované řešení modelu lineárního programování? Jak se degenerované řešení reprezentuje graficky?

Degenerované řešení - takové řešení, kde alespoň jedna z bazických proměnných ve výsledném vektoru řešení má nulovou hodnotu (normálně by měly být bazické proměnné vždy nenulové).

Grafická reprezentace degenerovaného řešení
Jedna z výsledných bazických proměnných v grafickém řešení vyjde nulová.

Např. V prostoru řešení – pokud nejvzdálenější průsečík rovnoběžky s účelovou funkcí a konvexního polyedru (simplexu) má jednu z proměnných x1, x2 nulovou (tedy leží na jedné z os souřadnic x1 nebo x2)
Např. V prostoru požadavků – pokud některá z výsledných proměnných xi, xj ze soustavy rovnic pro nejvzdálenější spojnici vektorů ai a aj s vektorem pravých stran b je nulová

5) K čemu slouží „základní věty lineárního programování“? Jaké mají důsledky pro hledání optimálního řešení modelu LP?

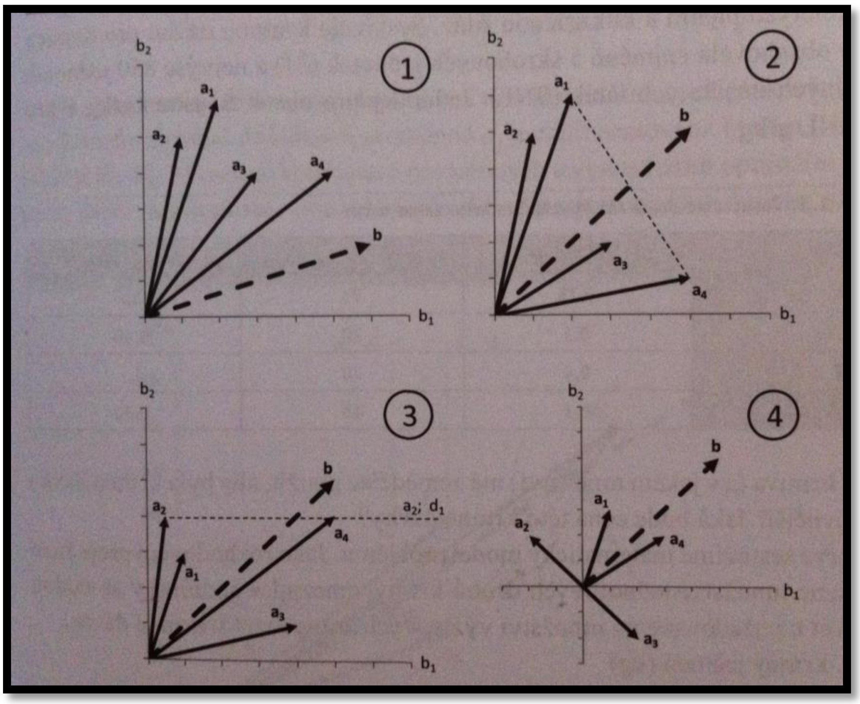
<u>**Základní věty LP:**</u>

- Optimální řešení úlohy LP leží vždy na hranici množiny přípustných řešení.
- Má-li úloha LP přípustné řešení, má i přípustné bázické řešení.
- Má-li úloha LP optimální řešení, má i optimální bázické řešení.
- Má-li úloha LP více než jedno optimální bázické řešení, je optimálním řešením i jejich lineární konvexní kombinace (např. i celá přímka, na které leží dvě optimální bázické řešení, je optimálním řešením)

Základní věty LP slouží při řešení úloh LP.

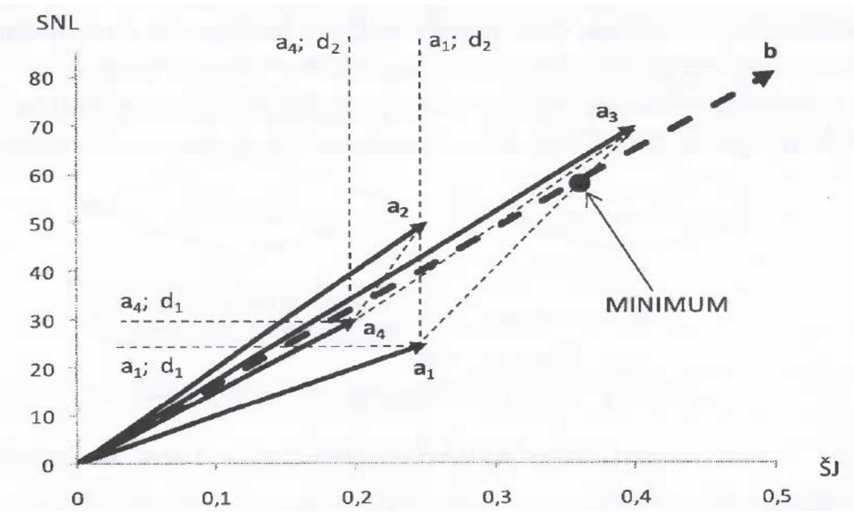
6) Uveďte 4 možné výsledky řešení modelů lineárního programování a znázorněte je graficky v prostoru požadavků.

1) Model nemá přípustné řešení (*MINIMALIZAČNÍ ÚLOHA)*
2) Model má právě jedno optimální řešení *(MINIMALIZAČNÍ ÚLOHA)*
3) Alternativní optimální řešení *(MINIMALIZAČNÍ ÚLOHA)*
4) Model má přípustné řešení, ale hodnota účelové funkce může neomezeně růst nebo klesat *(MAXIMALIZAČNÍ ÚLOHA)*



7) Uveďte, jak v prostoru požadavků určíte přípustné řešení modelu lineárního programování a jak vyberete řešení optimální. Dokumentujte rovněž graficky

Viz Krok 5 str. 6: Zakreslíme vektory a1, a2, a3, a4,… d1, d2,.. a vektor pravých stran b do **prostoru požadavků**.



Přípustné řešení – najdu jako všechny spojnice vektorů ai a aj (případně i d), které protnou vektor pravých stran b

- V grafu je více přípustných řešení, například dvojice a1, a3 nebo dvojice a1, a2 *(podmínka protnutí přímky b je tu splněna)*
- Jako přípustné řešení se nebere například a2,a3 *(neprotínají přímku b)*

Optimální řešení - v případě **minimalizace** hledáme spojnici mezi konci vektorů „a“ a „d“, která „nejdále vpravo nahoře“ protne vektor b. V případě **maximalizace** hledáme spojnici mezi konci vektorů „a“ a „d“, která „nejblíže vlevo dole“ protne vektor b. Viz ukázka doučování.

Téma 3: Simplexový algoritmus (str. 8-13)

1) Uveďte dvě základní podmínky pro aplikovatelnost simplexového algoritmu. Jaký je jejich význam, proč je jejich splnění nutné?

- **Nezápornost složek vektoru pravých stran (b)**
  - všechny pravé strany musí být nezáporné
  - pokud není splněno, lze příslušné omezující podmínky vynásobit hodnotou (-1).
- **Matice soustavy musí být v kanonickém tvaru** - matice musí mít tolik jednotkových vektorů, kolik je rovnic omezujících podmínek
  - Krok 1: převod z nerovnicového tvaru na rovnicový pomocí doplňkových proměnných (viz otázka 2 níže)
  - Krok 2: převod na kanonický tvar modelu pomocí pomocných (umělých) proměnných (viz otázka 3 níže)

2) Popište postup převodu modelu z nerovnicového do rovnicového tvaru. Proč tento krok při řešení modelu lineárního programování provádíme? (str. 8, krok 3)

Jelikož omezující podmínky v úloze LP bývají zpravidla ve tvaru nerovnic, je prvním krokem převod této soustavy lineárních nerovnic na soustavu lineárních rovnic pomocí tzv. **doplňkových proměnných d1, d2,.. nebo x4, x5,…** Do simplexové tabulky musí omezující podmínky totiž vstupovat v rovnicové podobě (viz dále).

Postup:

- kapacitní podmínky (≤) - přičteme hodnotu doplňkové proměnné k levé straně (+d = rezerva)
- požadavkové podmínky (≥) - od levé strany hodnotu doplňkové proměnné odečteme (-d = překročení požadavku)
- podmínky určení (=) - žádná transformace není potřeba
- doplňkové proměnné d1, d2, d3 číslujeme podle pořadí rovnice
- v účelové funkci dáme před všechny doplňkové proměnné cenu 0 (nevyděláváme na nich)

<u>Např:</u>
Původní nerovnicový tvar
x1+4x2+7x3 ≤ 100
x1+3x2 +5x3 ≥ 20
5x1+4x2+x3 = 300
Z (max)= 8x1+6x2+x3

Převod na rovnicový tvar:
x1+4x2+7x3 + **d1** =100
x1+3x2 +5x3 - **d2** =20
5x1+4x2+x3 =300
Z(max)= 8x1+6x2+x3+0d1+0d2

3) Popište postup převodu modelu z rovnicového do kanonického tvaru. Proč tento krok při řešení modelu lineárního programování provádíme? (str. 12, krok 4)

Převod modelu z rovnicového do kanonického tvaru provádíme, protože jednou z podmínek pro aplikaci simplexového algoritmu je, že matice soustavy musí být v **kanonickém tvaru**, abychom mohli využít Jordanovu eliminační metodu.

Definice: Model je v **kanonickém tvaru**, pokud obsahuje tolik jednotkových vektorů, kolik má rovnic. Úpravu modelu na kanonický tvar provedeme pomocí **pomocných (umělých) proměnných p1, p2, … nebo např. x6, x7, …**

Postup:

- Nerovnice převedeme na rovnice pomocí doplňkových proměnných – viz otázka 2
- Zajistíme úplnou jednotkovou submatici (tolik jednotkových vektorů, kolik je rovnic) pomocí **pomocných proměnných**
- pomocné proměnné p1, p2, p3 číslujeme podle pořadí rovnice
- Před pomocné proměnné dáme tzv. **prohibitivní sazby**. V případě maximalizace např. M = -100. Volím extrémně nevýhodnou sazbu, aby bylo jasné, že pomocná proměnná nemůže být vybrána do báze (řešení). „Pokud mám na nějaké proměnné zisk -100, je jasné, že nebude vybrána do řešení“. Obdobně u minimalizace volíme prohibitivní sazbu např. M = +100. Viz ukázka doučování.

<u>Např:</u>

Původní nerovnicový tvar

x1+4x2+7x3 ≤ 100
x1+3x2 +5x3 ≥ 20
5x1+4x2+x3 = 300
Z (max)= 8x1+6x2+x3

Převod na rovnicový tvar:

x1+4x2+7x3 + **d1** =100
x1+3x2 +5x3 - **d2** =20
5x1+4x2+x3 =300
Z(max)= 8x1+6x2+x3+0d1+0d2

Převod na kanonický tvar:

x1+4x2+7x3 + **d1** =100
x1+3x2 +5x3 - **d2** + **p2** =20
5x1+4x2+x3 + **p3** =300
Z(max)= 8x1+6x2+x3+0d1+0d2-100p2 -100p3

4) Uveďte a stručně popište typy proměnných v modelech lineárního programování. Ke každému typu proměnných uveďte příklad interpretace

<u>Typy proměnných:</u>

x … strukturní (rozhodovací) proměnné

- Jsou definovány již přímo v samotném zadání příkladu
- Udávají, v jakém množství se vyskytuje daný výrobek, plodina, …
- V účelové funkci mají před sebou většinou hodnotu zisku z jednotky dané proměnné ze zadání
- Např. x1 = plocha plodiny A, x2 = plocha plodiny B,…

d … doplňkové proměnné

- Jsou využívány při převodu z nerovnicového tvaru na rovnicový.
- Jejich hodnota se interpretuje jako „rezerva“, která zbývá do vyčerpání omezující podmínky typu <=, nebo jako „překročení požadavku“ v omezující podmínce typu >=
- V účelové funkci mají před sebou vždy nulovou hodnotu (nevyděláváme na nich)

- Např. d1 = nevyužitá rezerva d1 počtu pracovních hodin, d2 = překročení požadavku na počet krmných jednotek o d2

p … pomocné (umělé) proměnné

- Jsou využívány při převodu z rovnicového tvaru na kanonický tvar
- Jejich hodnota se neinterpretuje (umělé proměnné)
- V účelové funkci před pomocné proměnné dáváme tzv. **prohibitivní sazby**. V případě maximalizace např. M = -100. Volím extrémně nevýhodnou sazbu, aby bylo jasné, že pomocná proměnná nemůže být vybrána do báze (řešení). „Pokud mám na nějaké proměnné zisk -100, je jasné, že nebude vybrána do řešení". Obdobně u minimalizace volíme prohibitivní sazbu např. M = +100.

5) Uveďte a stručně popište typy omezujících podmínek v modelech lineárního programování. Ke každému typu uveďte příklad použití

- <u>**Exogenní (vnější) vazby systému**</u>
  a) **kapacitní „ ≤ ";**
  *Pěstuji pšenici, ječmen a žito, a chci, aby celková rozloha byla **nejvýše** 140 ha => x1+x2+x3 ≤ 140*
  b) **požadavkové„ ≥ ";**
  *Pěstuji pšenici, ječmen a žito, a chci, aby celková rozloha byla **nejméně** 140 ha => x1+x2+x3 ≥ 140*
  c) **určení „ = ".**
  *Pěstuji pšenici, ječmen a žito, a chci, aby celková rozloha byla **právě** 140 ha => x1+x2+x3 = 140*

- <u>**Endogenní (vnitřní) vazby systému**</u>
  a) **Bilanční** : x1 – x2 = 0
  Chci, aby součet příjmů x1 a nákladů x2 byl nulový (vyrovnaná bilance)
  b) **Poměrové** : x1/x2 = 2
  Chci, aby podíl příjmů x1 k nákladům x2 byl dvojnásobný

6) Prezentujte obecnou simplexovou tabulku. Jaké informace simplexová tabulka poskytuje? (str. 9, krok 4)

a) Do tabulky nahoru napíšeme všechny proměnné x1, …, x6 (případně např. x1, x2, x3, d1, d2, p1 atd) a nad ně do horního řádku dáme koeficienty (čísla), které byly před nimi v účelové funkci.
b) Pod to do každého řádku přepíšeme jednu omezující podmínku (pouze čísla) včetně pravých stran.
c) Před tabulku zleva do sloupce Z (báze) dáme proměnné, které jsou v té chvíli v bázi (proměnné, pod kterými jsou v tabulce jednotkové vektory).
d) Do sloupce ci dáme koeficienty, které jsou před bázovými proměnnými v účelové funkci (tedy i v tabulce nad bázovými proměnnými).
e) Vypočteme řádek zj-cj pod tabulkou: vynásobíme vždy mezi sebou postupně všechna čísla ze sloupce ci se všemi čísly ve sloupci pod danou proměnnou x1, x2, x3 (tzv. skalární součin), atd a odečteme od výsledku číslo z horního řádku. Výsledek napíšeme dolů do řádku zj-cj pod danou proměnnou.

c_i $i \in Z$	Z	1 x_1	2 x_2	4 x_3	0 x_4	0 x_5	0 x_6	x_i $i \in Z$	θ_{min}
0	x_4	1	0	0	1	0	0	2	-
0	x_5	1	1	2	0	1	0	4	4/2
0	x_6	0	3	4	0	0	1	6	6/4
$z_j - c_j$		−1	−2	−4	0	0	0	0	

f) V řádku zj-cj pod tabulkou vidíme záporná čísla. V případě maximalizace to znamená, že řešení ještě není optimální (**tzv. test optimality**). Musíme pokračovat dále. Viz doučování.

7) Popište účel, princip a postup provedení testu optimality v simplexové tabulce (str. 9, krok 4 e, f)

Účelem testu optimality je zjistit, jestli dané bazické proměnné v bázi jsou již optimálním řešením, nebo musíme upravovat simplexovou tabulku dále. Počítáme skalární součin **Zj-Cj** (viz předchozí otázka 6, krok e, f). Vynásobíme vždy mezi sebou postupně všechna čísla ze sloupce ci se všemi čísly ve sloupci pod danou proměnnou x1, x2, x3, atd a odečteme od výsledku číslo z horního řádku. Výsledek napíšeme dolů do řádku zj-cj pod danou proměnnou. Pod jednotkovými (bázovými) vektory musí vyjít 0.

Aby výsledek byl optimální, musí vyjít následující čísla v indexovém řádku **Zj-Cj** => **Pro MAXIMALIZACI -> kladné, pro MINimalizaci -> záporné. J**elikož většinou na počátku test optima nevyjde, musíme počítat dále *(určíme si klíčový sloupec a vypočítáme test přípustnosti=> klíčový řádek, přechod na nové řešení – viz následující otázka 8.)*

8) Popište účel, princip a postup provedení testu přípustnosti v simplexové tabulce (str. 9, krok 5)

Účelem testu přípustnosti je zjistit, která „vhodná“ proměnná bude nově **vstupovat do báze** (proměnná z klíčového sloupce) a která „nevhodná“ proměnná bude **vystupovat z báze** (proměnná z klíčového řádku).

Postup:

a) Vyberu z posledního řádku zj-cj nejvíce záporný prvek (pokud se jedná o maximalizaci). Pokud se jedná o minimalizaci, bereme nejvíce kladný prvek.
b) Toto číslo mi určí tzv. **klíčový sloupec**. Ten ukazuje na proměnnou, která je nejvíce výhodná a zanedlouho vstoupí do báze (do výběru výhodných proměnných).
c) Dále dělíme čísla pravých stran z předposledního sloupce xi čísly z klíčového sloupce a výsledky zapisujeme do posledního sloupce. Nejmenší z těchto výsledků dělení mi určí tzv. **klíčový řádek.** Ten ukazuje na proměnnou, která je nejméně výhodná a zanedlouho vystoupí z báze (z výběru výhodných proměnných).

Téma 4: Interpretace výsledku, dualita

1) Uveďte způsob, jak v simplexové tabulce identifikujete bázické a nebázické proměnné. Rovněž uveďte, jak určíte hodnoty všech proměnných v daném bázickém řešení (str. 9)

Bázické proměnné
- sloupce bazických proměnných v matici soustavy a simplexové tabulce jsou vždy tvořeny jednotkovými vektory
- v prvním levém sloupci simplexové tabulky najdeme vektor cen proměnných, které jsou v bázi
- v druhém levém sloupci simplexové tabulky najdeme názvy těchto bázických proměnných

Nebázické proměnné
- sloupce nebázických proměnných v matici soustavy a simplexové tabulce, které nejsou tvořeny jednotkovými vektory

2) Co je to matice báze B a inverzní matice báze B^{-1} v modelech lineárního programování? Jak tyto matice určíme a jaký je jejich význam? (str. 16)

Obě matice slouží jako základ pro další postoptimalizační úvahy.

Matice báze B
Sloupce matice báze B najdeme ve výchozí simplexové tabulce nad sloupci jednotkových vektorů, které se nachází ve výsledné tabulce. Pořadí je od sloupce s 1 v nejvyšším řádku až po 1 v nejnižším řádku.
Inverzní matice báze B^{-1}
Sloupce inverzní matice báze B^{-1} najdeme v těch sloupcích výsledné tabulky, v nichž ve výchozí tabulce byly jednotkové vektory. Pořadí je od sloupce s 1 v nejvyšším řádku až po 1 v nejnižším řádku.

3) Co jsou to duální ceny proměnných? Jak je určíme a interpretujeme? (str. 17)

Duální hodnoty čteme v řádku kritéria optimality zj-cj **výsledné** simplexové tabulky

Duální hodnoty prvého druhu se nalézají ve sloupcích **strukturních** (rozhodovacích)proměnných a udávají, o kolik se zhorší hodnota účelové funkce, zařadíme-li navíc jednu jednotku dané proměnné do řešení

Duální hodnoty druhého druhu se nalézají ve sloupcích **doplňkových** proměnných a udávají, o kolik se zhorší hodnota účelové funkce, necháme-li jednu jednotku daného činitele nevyužitou.

4) Určete obsah vektoru bázického řešení a vektoru obecného řešení modelu lineárního programování. Uveďte příklady obou vektorů. (str. 15, 16)

Vektor bazického řešení (str. 15)

Do vektoru bazického (základního) řešení dáme všechny hodnoty proměnných. U bázových proměnných bude hodnota z pravého sloupce výsledné tabulky, u nebázových bude hodnota 0.

Např: X_B = (5;4;0;0;1;0;60;0)

X_1 = 5….plocha plodiny A
X_2 = 4….plocha plodiny B
X_5 = 1….nevyužitá disponibilní orná půda
X_7 = 60….nevyužité pracovní hodiny v srpnu

Vektor obecného řešení (str. 16)

U bázových proměnných převedu všechny hodnoty proměnných z daného řádku na pravou stranu tabulky (kromě samotné proměnné, o kterou se jedná). U nebázových proměnných píšu jen jejich název.

$$X_O = \begin{pmatrix} 5+\frac{1}{3}x3+\frac{1}{3}x4-\frac{1}{15}x6+\frac{1}{30}x8 \\ 4-\frac{2}{3}x3-\frac{2}{3}x4+\frac{1}{30}x6-\frac{1}{15}x8 \\ x3 \\ x4 \\ 1-\frac{2}{3}x3-\frac{2}{3}x4+\frac{1}{30}x6+\frac{1}{30}x8 \\ x6 \\ 60-\frac{20}{3}x3+\frac{10}{3}x4+\frac{1}{3}x6+\frac{1}{3}x8 \\ x8 \end{pmatrix}$$

5) Co je to dualita modelů lineárního programování? Uveďte alespoň jeden příklad, kdy nám teorie duality výrazně zjednodušuje řešení úlohy. (str. 14)

Dualita je **vztah mezi dvěma modely LP**. V případě úloh LP se dualita projevuje tak, že ke každému **primárnímu** modelu LP se dá přiřadit jeho **duální** model LP, která má tytéž parametry (čísla), jako původní model, ale s jinou interpretací.

Příklad využití duality modelů LP

- Využití při převodu primárního modelu na duální model, který je možno řešit jednodušeji a přehledněji – např. graficky. Např. Model se dvěma omezujícími podmínkami a třemi proměnnými x1, x2, x3 nemůžeme řešit v prostoru řešení (viz doučování). Po převodu na duální model má již model pouze dvě proměnné y1, y2 a tři omezující podmínky a lze přehledně řešit v prostoru řešení.
- Využití při analýze výsledků, kdy tzv. duální ceny poskytují důležité informace pro rozhodování (viz Otázka 3)

6) Popište vztahy mezi prvky duálně sdružených úloh. (str. 14)

Převod z Maximalizace (primární úloha) na Minimalizaci (duální úloha)

Krok 1: Čísla nalevo z **řádků** omezujících podmínek primárního modelu dám nalevo do **sloupců** omezujících podmínek duálního modelu

Krok 2: Čísla napravo z omezujících podmínek primárního modelu dám do účelové funkce duálního modelu

Krok 3: Čísla z účelové funkce primárního modelu dám napravo do omezujících podmínek duálního modelu

Krok 4: Změním Z_{MAX} na Z_{MIN}

Krok 5: Znaménka z omezujících podmínek primárního modelu **otočím** a dám je do podmínek nezápornosti duálního modelu

Krok 6: Znaménka z podmínek nezápornosti primárního modelu **neotáčím** a dám je do omezujících podmínek duálního modelu

Převod z Minimalizace (primární úloha) na Maximalizaci (duální úloha)

Krok 1-3: Stejně jako u převodu z Maximalizace na Minimalizaci

Krok 4: Změním Z_{MIN} na Z_{MAX}

Krok 5: Znaménka z omezujících podmínek primárního modelu **neotáčím** a dám je do podmínek nezápornosti duálního modelu

Krok 6: Znaménka z podmínek nezápornosti primárního modelu **otočím** a dám je do omezujících podmínek duálního modelu

7) Co říká základní věta o dualitě? Jaký je její význam?

Má-li jedna z dvojice duálně sdružených úloh optimální řešení, má optimální řešení i úloha druhá a optimální hodnoty obou účelových funkcí jsou stejné.

Význam základní věty o dualitě

V důsledku věty o dualitě stačí vyřešit pouze jednu z duálně sdružených úloh, protože optimální řešení a hodnotu účelové funkce druhé úlohy lze z tohoto řešení odvodit.

Téma 5 a 6: Postoptimalizační analýza, praktické aplikace

1) Uveďte postup stanovení nebázického řešení modelu LP. Jak určíte minimální a maximální hodnotu, kterou může nebázická proměnná nabýt? Co víte o optimálnosti nebázických řešení? (str. 18, 19, kapitola f)

Postup pro stanovení nového nebázického řešení při zařazení nové nebázické proměnné

Od sloupce pravých stran výsledné simplexové tabulky odečteme hodnotu nově zařazované nebázické proměnné vynásobenou sloupcem výsledné tabulky pod zařazovanou nezákladní proměnnou. Předtím ale musím ještě určit interval přípustných hodnot.

Postup pro výpočet nové hodnoty účelové funkce

Dosadíme nově vypočtené hodnoty proměnných do účelové funkce a vypočítáme její novou hodnotu.

Určení minimální a maximální hodnoty, kterou může nebázická proměnná nabýt (interval přípustných hodnot)

Zajímá nás, na jaké maximální úrovni můžeme nebázickou proměnnou do řešení zařadit, aby řešení zůstalo přípustné a v téže bázi (viz doučování) – mluvíme o intervalu přípustných hodnot nebázické proměnné

Minimální přípustná hodnota zařazované nebazické strukturní proměnné x_k je vždy 0.

Maximální přípustná hodnota zařazované nebazické strukturní proměnné x_k se spočítá jako minimální hodnota z podílů pravých stran výsledné tabulky a pouze kladných koeficientů ve sloupci výsledné tabulky pod danou zařazovanou proměnnou.

Nebázická řešení většinou zhoršují hodnotu účelové funkce.

2) Bylo rozhodnuto zařadit do optimálního řešení nový proces (nebázickou strukturní proměnnou). Popište postup, jak určíte vliv této změny na další parametry modelu (hodnoty bázických proměnných a účelové funkce). (str. 18, 19, kapitola f)

Viz předchozí Otázka 1

3) Po provedené optimalizaci modelu LP došlo ke změně kapacity jednoho zdroje. Popište postup, jak určíte vliv této změny na další parametry modelu (hodnoty bázických proměnných a účelové funkce). (str. 20, kapitola g)

Postup pro výpočet nového řešení při změně pravé strany b_i

Od sloupce pravých stran výsledné simplexové tabulky odečteme (při snížení) nebo přičteme (při zvýšení) hodnotu změny pravé strany bi vynásobenou sloupcem výsledné tabulky pod doplňkovou proměnnou odpovídající dané pravé straně. (str. 20, kapitola g)

Postup pro výpočet nové hodnoty účelové funkce

Dosadíme nově vypočtené hodnoty proměnných do účelové funkce a vypočítáme její novou hodnotu.

Určení minimální a maximální hodnoty, kterou může daná pravá strana nabýt (interval přípustných hodnot pro pravou stranu b_k)
Úkolem je zjistit rozsah pravých stran tak, aby nedošlo ke změně báze. Změnu pravých stran si vyjádříme pomocí vektoru parametrů $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$. Např $\lambda_2 = 2$ znamená zvýšení pravé strany b_2 o 2 jednotky.

Postup pro výpočet intervalu přípustných hodnot pro parametry λ_k.
Minimální hodnotu spočítáme jako maximum ze záporných podílů pravých stran výsledné tabulky s kladnými čísly ve sloupci matice B^{-1}, který odpovídá pořadí dané pravé strany. Pokud nenajdu žádné kladné číslo, je dolní mez $-\infty$.
Maximální hodnotu spočítáme jako minimum ze záporných podílů pravých stran se zápornými čísly ve sloupci matice B^{-1}, který odpovídá pořadí dané pravé strany. Pokud nenajdu žádné záporné číslo, je horní mez ∞.

4) K čemu slouží analýza stability báze vzhledem ke složkám vektoru pravých stran? Popište rámcově způsob jejího provedení.

Viz předchozí Otázka 3

5) K čemu slouží analýza citlivosti řešení vzhledem ke změnám cenových koeficientů c_j? Popište rámcově způsob jejího provedení, rozlište postup pro bázické a nebázické proměnné. (str. 21, kapitola h)

Úkolem je zjistit, v jakém intervalu se mohou pohybovat ceny strukturních proměnných v účelové funkci, aby nedošlo ke změně báze (u nestrukturních nemá smysl uvažovat).
Změny cen si vyjádříme pomocí vektoru parametrů $v = (v_1, v_2, \ldots, v_p)$. Např $v_2 = 5$ znamená zvýšení ceny druhé proměnné v účelové funkci o 5.

Výpočet intervalu přípustných hodnot pro jednotkovou změnu c_j <u>nebazické</u> strukturní proměnné x_j:
$-\infty \leq v_j \leq z_j - c_j$

Výpočet intervalu přípustných hodnot pro jednotkovou změnu cj <u>bazické</u> strukturní proměnné xj

Minimální hodnotu spočítáme jako maximum ze záporných podílů řádku zj-cj s kladnými čísly v řádku výsledné tabulky, který odpovídá dané bazické strukturní proměnné.
Maximální hodnotu spočítáme jako minimum ze záporných podílů řádku zj-cj se zápornými čísly v řádku výsledné tabulky, který odpovídá dané bazické strukturní proměnné.

Poznámka: Hodnoty 0 v řádku zj-cj při podílech ignorujeme

6) Uveďte a stručně popište podstatu úlohy o výrobním programu a směšovací úlohy jako praktické aplikace modelu LP.

1) Úloha o výrobním programu – podstatou je nalézt optimální množství jednotlivých výrobků (např. plodin A, B, C..) tak, aby byly splněny všechny omezující podmínky a byl zajištěn maximální zisk, případně minimální náklady

2) Směšovací úloha – podstatou je nalézt optimální poměr pro smíchání nějakých komponent, tak abychom dostali výsledný produkt s požadovanými vlastnostmi a celý směsný plán byl co nejlevnější

7) Uveďte a stručně popište podstatu úlohy o řezných plánech a plánování směn jako praktické aplikace modelu LP.

1) Úloha o řezných plánech – podstatou je nalézt optimální řezný plán pro rozřezání předem daného počtu polotovarů, tak aby byly splněny požadavky na počet nařezaných dílů. Kritériem je minimalizace spotřebovaného materiálu, minimalizace odpadu, případně maximalizace zisku

2) Úloha o plánování směn – podstatou je optimálně naplánovat směny při splnění daných požadavků na počet pracovníků v daných hodinách. Kritériem je minimalizace počtu pracovníků a tedy i nákladů.

Téma 7: Jednostupňová dopravní úloha I

1) Uveďte podstatu a komponenty jednostupňové dopravní úlohy. (str. 22)

Podstatou je najít co nejvýhodnější plán přepravy produktu od dodavatelů ke spotřebitelům.
Model Jednostupňové dopravní úlohy a jeho komponenty: Je dáno m dodavatelů D_1, D_2, .., D_m a n spotřebitelů S_1,S_2, …S_n. Dodavatelé mají kapacity zboží a_1, a_2, …a_m a spotřebitelé mají kapacity zboží b_1, b_2, …b_n. Cena dopravy (vzdálenost) mezi dodavatelem D_i a spotřebitelem S_j je rovna c_{ij}. Cílem úlohy je minimalizovat přepravní náklady (celkový počet tunokilometrů).

2) Co je to vyváženost modelu jednostupňové dopravní úlohy? Jak se provádí?

Dopravní úloha je **vyvážená (vybilancovaná)**, pokud součet kapacit dodavatelů je roven součtu požadavků spotřebitelů.

Vyvážení DÚ provádíme pomocí fiktivního dodavatele (v případě převisu požadavků spotřebitelů) nebo fiktivního spotřebitele (v případě převisu kapacit dodavatelů). U fiktivního dodavatele i fiktivního spotřebitele zadáváme vždy všechny vzdálenosti (ceny) nulové.

3) Stručně popište princip metody severozápadního rohu v modelu jednostupňové dopravní úlohy. K čemu se tato metoda používá, jak dobré výsledky poskytuje? (str. 22)

Metoda SZ rohu se používá spíše ilustrativně a poskytuje kvalitativně nejhorší výsledky
Postup:

Krok 1: Zjistíme, zda je úloha vyvážená (vybilancovaná).

Krok 2: Navezeme maximum do levého horního (severozápadního) rohu a dodavatele nebo spotřebitele, který se naplní, vyškrtneme.

Krok 3: Tímto krokem se tabulka zmenší a pokračujeme dále až do rozvezení všeho zboží. Při kontrole nám musí množství rozvezeného zboží v tabulce odpovídat celkovému rozváženému množství v pravém dolním rohu tabulky.

Krok 4: Vypočítáme hodnotu účelové funkce (výsledný počet tunokilometrů)

Metoda SZ rohu se používá spíše ilustrativně a poskytuje kvalitativně nejhorší výsledky.

4) Stručně popište princip indexové metody v modelu jednostupňové dopravní úlohy. K čemu se tato metoda používá, jak dobré výsledky poskytuje? (str. 23)

Indexová metoda je založena na postupném obsazování polí podle nejmenších indexů (vzdáleností). Tedy začíná od nejmenšího tedy nejvýhodnějšího indexu v tabulce (tzv. "hladová metoda"). Dává kvalitativně lepší výsledky než metoda severozápadního rohu, ale horší než metoda VAM.
Postup:

Krok 1: Zjistíme, zda je úloha vyvážená (vybilancovaná).

Krok 2: Najdeme buňku s nejmenším indexem (vzdáleností) a obsadíme maximálním množstvím zboží. Nuly u případného fiktivního dodavatele či spotřebitele zpočátku ignorujeme a obsazujeme až nakonec. Dodavatele nebo spotřebitele, který se naplní, vyškrtneme.

Krok 3: Tím se tabulka zmenší a pokračujeme dále podle nejmenšího indexu až do rozvezení všeho zboží. Při kontrole nám musí množství rozvezeného zboží v tabulce odpovídat celkovému rozváženému množství v pravém dolním rohu tabulky.

5) Stručně popište princip Vogelovy aproximační metody (VAM) v modelu jednostupňové dopravní úlohy. K čemu se tato metoda používá, jak dobré výsledky poskytuje? (str. 24)

Tato metoda se rozhoduje podle největší diference (rozdílu) v příslušných řádcích a sloupcích. Jedná se o nejspolehlivější metodu ve srovnání s metodou severozápadní a indexovou.

Postup:

Krok 1: Zjistíme, zda je úloha vyvážená (vybilancovaná).

Krok 2: Vypočteme rozdíly mezi dvěma nejnižšími čísly ve všech řádcích a sloupcích – získáme tak **řádkové a sloupcové diference.**

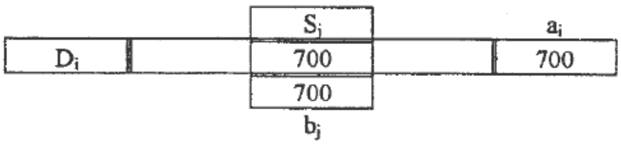
Krok 3: V řádku nebo sloupci, kde je největší diference se najde nejmenší číslo a obsadí se maximálním množstvím zboží. Dodavatel nebo odběratel, který je naplněn se vyškrtne. Vznikne nová tabulka, ve které musíme opět přepočítat nové diference. Takto postupujeme stejným postupem dále až do rozvezení celé kapacity zboží. Pokud mi už zbude poslední řádek či sloupec a nejdou spočítat diference, obsadím podle nejmenší vzdálenosti.

6) Co je to degenerace v modelu jednostupňové dopravní úlohy? Jak vzniká, jak se určuje a jak se odstraňuje? (str. 29)

Degenerace řešení nastává, pokud je počet buněk obsazených zbožím v dopravní tabulce menší než m+n-1, kde m = počet řádků, n = počet sloupců. Neboli v bázi (řešení) je menší počet buněk, než by měl být.

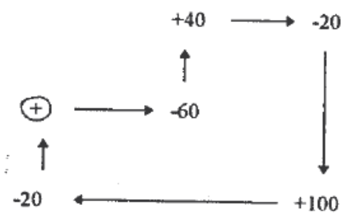
Degenerace řešení může vzniknout 2 způsoby:

1) Degenerace při konstrukci výchozího řešení – nastává, pokud obsadíme buňku, která najednou naplní dodavatele i spotřebitele se stejnými kapacitami. Vyškrtnu najednou oba dva (dodavatele i spotřebitele), ubyde mi jedno řešení z báze, a mám méně než m+n-1 plných buněk.



Odstranění: U dodavatele, nebo spotřebitele zvýšíme kapacitu o malý kousek ε (zanedbatelné množství např 1 g).

2) Degenerace při přesunu po Dantzigově obvodu – nastává, pokud v rozích označených zápornými znaménky jsou 2 stejné hodnoty. Při přesunu zboží po obvodu se namísto obvyklé 1 buňky vynulují 2 a počet obsazených buněk klesne pod m+n-1.



Odstranění: K jednomu ze 2 stejných čísel přidáme malý kousek +ε (zanedbatelné množství např. 1 g).

Téma 8 a 9: Jednostupňová dopravní úloha II, dvoustupňová úloha

1) Uveďte princip Modifikované distribuční metody MODI při řešení modelu jednostupňové dopravní úlohy. (str. 27)

Modifikovaná distribuční metoda MODI slouží k testování, zda řešení v dopravní tabulce je již optimální nebo musíme ještě pokračovat.

Princip:

Krok 1: Po stranách tabulky dopočítáme postupně duální hodnoty u a v. Položíme $u_1 = 0$ (pokud by se nepodařilo dopočítat, zvolíme jako 0 nějaké jiné u nebo v). Postupně dopočítáváme u a v, aby platilo u+v = c, ale pouze pro buňky, ve kterých je nějaké zboží.

Krok 2: V buňkách, kde není zboží, do **levého dolního rohu** napíšeme součty u+v = z. Do **levého horního rohu** napíšeme rozdíly r = z-c. Tedy: **levý horní roh = levý dolní - pravý horní roh**

Krok 3: Kdyby už vyšly všechny **levé horní rohy** r záporné nebo 0, je řešení optimální, nemusíme dál upravovat a máme výsledek. Pokud vyjdou nějaká kladná r, musíme pokračovat.

2) Co je to Danzigův uzavřený obvod? K čemu slouží při řešení modelu jednostupňové dopravní úlohy? (str. 27, 28)

Pomocí **Danzigova uzavřeného obvodu** se přechází v tabulce dopravní úlohy na lepší řešení (přechod na novou bázi) v případě, že ještě nevyšel test optimality metodou MODI. Po provedení Danzigova uzavřeného obvodu provádíme znovu test optimality až do nalezení optimálního řešení.

Postup:

Krok 1: Vybereme buňku s největším kladným **levým horním** rohem r a do této buňky dáme značku ⊕. Tam budeme přesouvat zboží pomocí Dantzigových uzavřených obvodů. Pravidla: Můžeme postupovat pouze rovně (nahoru, dolů, doleva, doprava) ne šikmo a pouze po buňkách, kde je zboží. Následně propojíme trasu čarami a buňky střídavě označíme + a -.

Krok 2: Vybereme nejmenší množství v buňce s označením – a přesuneme toto množství po buňkách Dantzigova obvodu. U + přičteme, u – odečteme.

Krok 3: Vznikne nám nová tabulka (viz podrobně na doučování), v ní pokračujeme stejným způsobem (znova u, v, dopočítám rohy buněk atd. ..). Provádíme tak dlouho, dokud nám nevyjdou všechny levé horní rohy r záporné. V tu chvíli jsme hotovi a máme optimální řešení.

Krok 4: Dopočítáme výslednou hodnotu účelové funkce

3) Co je to perspektivita dopravních tras? Jak se analýza perspektivity provádí? (str. 28)

Perspektivita spoje - je to absolutní hodnota levých horních rohů rij, neboli hodnota testu optimality v optimálním řešení u nerealizovaného spoje (tedy u prázdných buněk, kde nic nevezu). U optimálních spojů (obsazené buňky) je perspektivita rovná 0. Určuje míru zhoršení účelové funkce, pokud bychom využili tento spoj. Čím je toto číslo menší, tím je spoj výhodnější. Je li tato hodnota malá, můžeme po této trase převážet zboží se zanedbatelným zhoršením účelové funkce. Čím je tedy hodnota $|r_{ij}|$ nižší, tím je spoj perspektivnější. Je-li hodnota $|r_{ij}|$ dokonce nulová, existuje rovnocenné **alternativní řešení**. Po tomto spoji můžeme tedy převážet zboží až do výše propustnosti spoje (viz doučování) a účelová funkce se nezmění.

4) Co je to propustnost dopravních tras? Jak se analýza propustnosti provádí? (str. 28)

Propustnost spoje - říká, jaké je maximální možné množství zboží, které může být na daném spoji přepraveno. U optimálních spojů (obsazené buňky) je propustnost rovná přímo přepravovanému množství zboží (uprostřed buňky). U nerealizovaných spojů (prázdné buňky) se propustnost rovná hodnotě, kterou je možno přesunout po Dantzigově obvodu sestrojeném k tomuto spoji (tedy minimální hodnotě v buňkách tohoto obvodu označených -). Viz ukázka doučování.

5) Uveďte podstatu a komponenty jednostupňové dopravní úlohy.

Viz Téma 7, Otázka 1

6) Jaký je rozdíl mezi počtem rozměrů a počtem stupňů dopravní úlohy? Navrhněte a stručně popište možnou praktickou aplikaci alespoň dvou dopravních úloh, které se liší počtem stupňů i rozměrů.

Počet rozměrů úlohy

- počet faktorů, o nichž rozhodujeme;
- dvourozměrná – pouze trasy (odkud – kam);
- třírozměrná – trasy, vozidlo (odkud – kam – čím).

Počet stupňů úlohy

- počet dopravních uzlů na cestě od primárního dodavatele k finálnímu spotřebiteli.
- Jednostupňová dopravní úloha – pouze dodavatelé D a spotřebitelé S
- Dvoustupňová dopravní úloha - mezi dodavatele D a spotřebitele S vložíme tzv. mezisklady M

Téma 10: Další dopravní modely

1) Uveďte podstatu a komponenty přiřazovací úlohy. (str. 34)

Přiřazovací úloha se používá k optimálnímu přiřazení n prvků k n místům. Je to v podstatě analogie dopravního problému, ale máme n dodavatelů D_1, D_2, .., D_n a n spotřebitelů S_1,S_2, …S_n. Cena dopravy (vzdálenost) mezi dodavatelem D_i a spotřebitelem S_j je rovna c_{ij}. Všichni dodavatelé i spotřebitele mají kapacitu 1. Nejčastějším úkolem je rozvést při minimálních nákladech n strojů do n míst, přičemž máme v zadání jednotlivé vzdálenosti.

Komponenty

1. **dodavatelé** – objekty které jsou přiřazovány, celkově n (např. traktory)
2. **odběratelé** – místa kam je vezeno (přiřazováno) – těchto cílů je celkově také n
3. **matice sazeb** – jak drahé je přidělení jednoho dodavatelského objektu k jednomu objektu odběratelskému, na rozdíl od jednostupňové dopravní úlohy nás zajímá pouze využití nebo nevyužití dané trasy a počet přepravovaných jednotek zde nemá smysl uvažovat - bude vždy 1.

Pro výpočet přiřazovací úlohy se používá Maďarská metoda. Není možno využít klasický postup pro dopravní úlohu, protože přiřazovací úloha je často silně degenerovaná (v tabulce bude vždy obsazeno méně než m+n-1 polí, konkrétně m). Viz doučování.

2) K čemu slouží maďarská metoda? Stručně popište její princip. (str. 34)

Maďarská metoda se používá pro řešení přiřazovací úlohy. Důležitým termínem je tzv. **nezávislá nula** – je to taková nula, která je sama v příslušném řádku nebo sloupci.

Postup:

Krok 1: Provedeme řádkovou redukci tak, že od sazeb v jednotlivých řádcích odečteme vždy nejmenší sazbu.

Krok 2: Pokud některý ze sloupců neobsahuje žádnou nulu, provedeme ještě sloupcovou redukci tak, že v jednotlivých sloupcích odečteme nejnižší sazbu.

Krok 3: V redukované matici sazeb vybíráme nezávislé nuly (nuly, které jsou samy v řádku nebo sloupci). Nezávislou nulu vždy označíme rámečkem a škrtneme řadu (řádek či sloupec), která je **kolmá** na řadu ve které je nula sama (tedy tu druhou, kde jsou většinou ještě jiné nuly). Pokud je nula sama v řádku i sloupci (tzv. silně nezávislá nula), můžeme si vybrat, zda škrtneme řádek či sloupec (např. vždy řádek). Škrtnuté nuly (řady) už dále ignorujeme. Takto pokračujeme, dokud je možno vybírat nezávislé nuly. Postupujeme například vždy shora dolů po řádcích, abychom nic nepřeskočili.

Krok 4: Pokud po výběru nezískáme právě n nul (test optimality), provedeme **sekundární redukci** matice:

a) určíme nejmenší prvek z nepřeškrtnutých prvků
b) prvky přeškrtnuté 1 x necháme beze změny
c) prvky přeškrtnuté dvakrát o minimální prvek zvětšíme
d) prvky nepřeškrtnuté o minimální prvek snížíme

Krok 5: Postup opakujeme tak dlouho, dokud nezískáme právě n nul

Krok 6: Dopočítáme hodnotu účelové funkce

3) Uveďte podstatu a komponenty okružního dopravního problému (str. 40 - 46)

Používá se k nalezení nejkratší cesty, která začíná a končí ve stejném místě a obsahuje všechny vrcholy. Nejčastějším úkolem je projet co nejkratší cestou všechna města a vrátit se zpět do původního. Někdy též nazýváno **„Úloha obchodního cestujícího“** nebo **„Úloha čínského listonoše“.**

Komponenty modelu:

- počáteční (a tedy i konečné) místo
- ostatní navštěvovaná místa
- trasy mezi navštěvovanými místy
- ocenění tras, obvykle vzdálenosti mezi místy

4) Uveďte a stručně charakterizujte základní typy okružních dopravních problémů.

Jednookruhový okružní dopravní problém (str. 40 - 43)
Používá se k nalezení nejkratší cesty, která začíná a končí ve stejném místě a obsahuje všechny vrcholy. Viz Otázka 3.

Víceokruhový okružní dopravní problém (str. 44 - 46)
Nejčastější příčinou, proč je třeba někdy rozdělit jednookruhový okružní problém na více okruhů, je kapacitní omezení vozidel. Je tedy třeba naplánovat více okruhů (každý pro jedno vozidlo) tak, aby začínal a končil v centrálním místě. Suma kapacit (požadavků) všech necentrálních míst na jednom okruhu přitom nesmí překročit předem danou kapacitu vozidla

5) Kde a k čemu se používá metoda nejbližšího souseda? Stručně popište její princip (str. 40)

Metoda nejbližšího souseda se používá při řešení jednookruhových okružních problémů.

Krok 1: Procházím postupně všechna místa. Ke každému najdu nejbližšího souseda, k němu opět nejbližšího souseda atd. až projedu všechna místa a vrátím se zpět.

Krok 2: Vyberu nejkratší trasu (trasy) a určím její (jejich) délku.

6) Popište modifikaci Vogelovy aproximační metody (VAM) pro řešení okružních dopravních problémů (str. 41-43)

Vogelova aproximační metoda se používá při řešení jednookruhových okružních problémů.

Krok 1: Vypočítáme si řádkové a sloupcové diference (rozdíl dvou nejmenších vzdáleností v daném řádku či sloupci – viz VAM metoda u dopravní úlohy)

Krok 2: Vybereme nejmenší sazbu v řadě s maximální diferencí – tuto trasu vybereme do okruhu. Škrtáme řádek a sloupec u vybrané trasy a trasu, která předčasně uzavírá okruh (u izolovaných tras je to pouze cesta zpátky).

Krok 3: Přepočítáme postupně opět diference již bez vyškrtnutého řádku a sloupce a volíme další trasu. Postupujeme tak dlouho, dokud neprojedeme všechna místa a nevrátíme se zpět do počátečního místa.

Krok 4: Navrhneme doporučenou trasu (tak, aby začínala v počátečním místě) a vypočteme délku cesty.

7) Kde a k čemu se používá Mayerova metoda? Stručně popište její princip (str. 44-46)

Mayerova metoda se používá při řešení víceokruhových okružních problémů.

Krok 1: V tabulce víceokruhové úlohy si seřadíme místa (v řádcích i sloupcích) sestupně podle vzdálenosti od centrálního místa (to ale v tabulce vynecháme). Přidáme napravo sloupec požadavků.
Krok 2: Označíme první sloupec této tabulky. Vyškrtneme první řádek a označíme tučně příslušný požadavek.
Krok 3: Pro každé z ostatních míst sečteme jeho požadavek s označeným a u všech míst, kde tento součet bude větší než kapacita vozidla, vyškrtneme ve vybraném prvním sloupci buňku v příslušném řádku. V případě prvního sloupce většinou žádná buňka.
Krok 4: Z nevyškrtnutých prvků v prvním sloupci vybereme minimální prvek (v případě shody ten vyšší v tabulce). Ten označuje místo, které půjde jako další do okružní trasy. Odpovídající sloupec opět zvýrazníme, příslušný řádek vyškrtneme a zvýrazníme příslušný požadavek.
Krok 5: Sečteme všechny zvýrazněné požadavky a pro ta místa, kde přičtením jejich požadavku s uvedeným součtem bude překročena kapacita, buňky v daných řádcích vyznačeného sloupce škrtneme. Z nevyškrtaných prvků v označeném sloupci bychom opět vybrali nejmenší prvek a zařadili do okružní trasy atd. Pokud ale máme již všechny prvky vyškrtnuty – máme první okruh.
Krok 6: Ve zbylé části tabulky hledáme opět stejným způsobem místa do dalších okružních tras.
Poznámka: Pokud vzhledem ke kapacitě automobilu je možno objet zbytek měst z centrálního místa už jen jedním okruhem, nemusíme dále počítat.
Krok 7: V každém ze samostatných okruhů najdeme optimální trasu jako u jednookružního dopravního problému (metodou nejbližšího souseda, nebo VAM metodou)

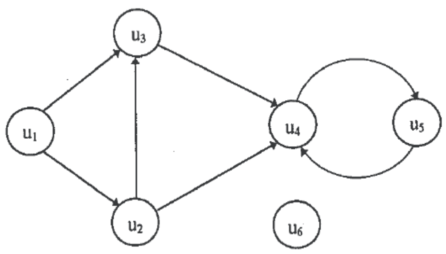
Téma 11: Modely teorie grafů

1) Co rozumíme termínem "graf" v teorii grafů? Jaký je rozdíl mezi orientovaným a neorientovaným grafem?

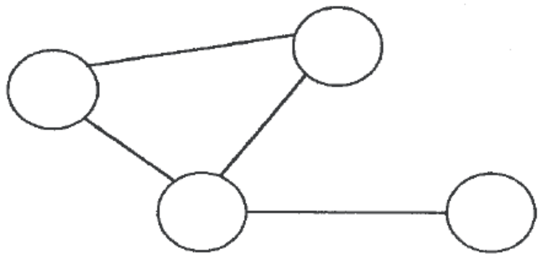
Grafem v teorii grafů rozumíme množinu, která se skládá z bodů a jejich spojnic. Body se nazývají uzly (tvar kruhu) a jejich spojnice hrany (tvar čáry). Uzly se označují symboly u1, u2..un a hrany, které spojují uzel ui s uzlem uj, symbolem (i,j).

Orientovaný graf - hranám grafu **lze** přisuzovat určitý směr, který vyznačujeme šipkou.

Neorientovaný graf - hranám grafu **nelze** přisuzovat určitý směr



Orientovaný graf



Neorientovaný graf

2) Charakterizujte a odlište pojmy "sled", "tah", "cesta" a "cyklus" v teorii grafů.

- ***Sled*** je posloupnost uzlů a hran, která spojuje dva vybrané uzly (nemusí být sousední)
- ***Tah*** je sled, který nepoužívá žádnou hranu více než jednou
- ***Cesta*** je tah, který nepoužívá žádný uzel více než jednou
- ***Cyklus*** je cesta, která začíná a končí ve stejném uzlu

3) Popište princip neinformovaného prohledávání grafů. Uveďte a stručně popište vybranou metodu neinformovaného prohledávání grafu.

Neinformované metody prohledávání nemají k dispozici žádné vhodné znalosti, které by jim umožnily urychlit cestu k cíli. Jsou tak odsouzeny k systematickému procházení všech uzlů, dokud nenaleznou řešení – tzv. **„slepé metody“.** Např. Při hledání nejkratší cesty mezi dvěma místy prohledávám systematicky všechny uzly, dokud nenajdu nejkratší cestu
Např.
- prohledávání do hloubky
- prohledávání do šířky

4) Popište princip informovaného prohledávání grafů. Uveďte a stručně popište vybranou metodu informovaného prohledávání grafu.

Informované metody prohledávání mají navíc znalosti, které jim umožňují odhadnout, jak daleko se nachází řešení od aktuálního stavu. Tento odhad reprezentuje hodnotící funkce f, kterou definuje na základě znalostí člověk, a informované metody jsou na ní závislé. Většinou rychlejší postup než v případě neinformovaného prohledávání grafů

Např. Dijkstrův algoritmus pro nalezení nejkratší cesty mezi dvěma místy – viz dále Otázka 6

5) Charakterizujte úlohu o hledání minimální kostry grafu, stručně popište princip jejího řešení (str. 47)

Používá se k nalezení nejkratšího propojení míst např. drahým kabelem. Na rozdíl od okružního problému se nevracíme zpátky, ale stačí jen místa propojit (**minimální kostra v grafu)**. Pro řešení se používá např**. Kruskalův algoritmus**

Postup:

Krok 1: Procházíme v grafu hrany od nejkratší po nejdelší a zařazujeme (zvýrazňujeme) postupně jen ty, které nevytvoří okruh. Délky hran, které už jsme zařadili nebo přeskočili z důvodu hrozícího okruhu, dáváme do kroužku.

Krok 2: Algoritmus končí, pokud máme propojeny všechny vrcholy.

Krok 3: Určíme délku minimální kostry

6) Charakterizujte úlohu o hledání nejkratší cesty v grafu, stručně popište princip jejího řešení (str. 48 - 50)

Používá se k nalezení nejkratší vzdálenosti z počátečního místa do nějakého finálního místa. Pro výpočet se používá např. **Dijkstrův algoritmus.**

Krok 1: Vytvoříme si tabulku, kde do řádku dáme všechny vrcholy A, B, C,…. Nad nimi vytvoříme řádek pro vzdálenosti. Nad počáteční vrchol A do tabulky dáme vzdálenost 0 (vzdálenost z A do A je nulová). Určíme vzdálenost z počátečního vrcholu A do všech uzlů, kde se dá z vrcholu A dostat a zapíšeme do tabulky. Vybereme vrchol s nejmenší vzdáleností a přejdeme v tabulce do tohoto bodu, nad něj napíšeme vzdálenost z bodu A.

Krok 2: Postupujeme podobným způsobem dále až do té doby, než dojdeme do konečného vrcholu – viz postup ze str. 46-48

7) Charakterizujte úlohu o hledání maximálního toku v síti, stručně popište princip jejího řešení (str. 51 - 52)

Používá se k nalezení maximálního toku, který můžeme pustit do dané sítě (např. potrubí, síť silnic). Pro výpočet se používá např. **Ford-Fulkersonův algoritmus.**

Krok 1: Postupujeme například vždy shora od výše položených cest. Vybereme si nějakou nenasycenou cestu (tedy cestu, která má kapacity všech hran větší než 0). V této cestě najdeme „nejužší" místo (kudy jde vést nejmenší množství). Tato hodnota je rovna maximálnímu toku, který můžeme pustit po této cestě. Na uvedené cestě musíme ještě snížit kapacitu všech hran o tuto hodnotu.

Krok 2: Vybereme si shora další nenasycenou cestu. V této cestě opět najdeme „nejužší" místo. Postupujeme stejným způsobem dále až do té doby, než už v dané síti nenajdeme žádnou nenasycenou cestu – viz postup ze str. 51-52

Téma 12: Modely projektového řízení

1) Co je to projekt? Uveďte vybranou definici a proveďte rozbor jejích klíčových slov.

Projekt je soubor provázaných **činností**, které je třeba provést k dosažení stanoveného cíle. Pro řešení metodou kritické cesty využíváme tzv. **síťový graf**, který se skládá z **uzlů** a **orientovaných hran**. Rozbor klíčových slov viz další otázky.

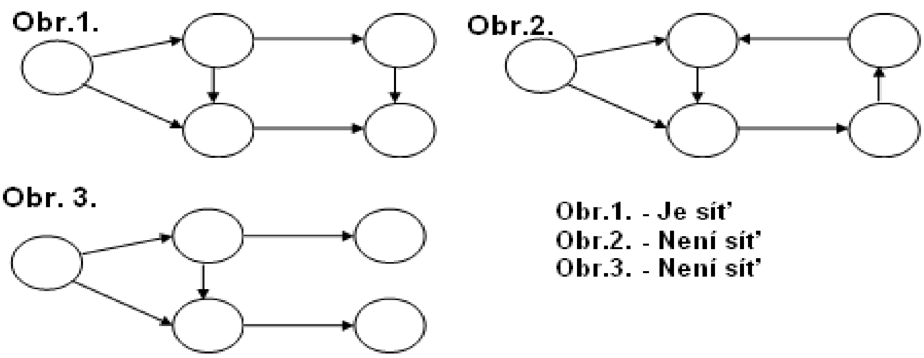
2) Charakterizujte pojmy "činnost" a "zdroj" v projektovém řízení. Vždy uveďte příklady z praxe.

Činnost
- základní jednotka projektu; určitá aktivita, kterou je třeba vykonat
- např. kopání základů domu

Zdroj
- faktor zabezpečující činnost, v průběhu projektu se využívá nebo spotřebovává;
- např.: zedník, řidič, vedoucí projektu, ale i osobní automobil, kancelář nebo písek, PHM.

3) Charakterizujte graf typu síť (síťový graf), dokumentujte rovněž graficky.

Síť (síťový graf) - je graf, který je **spojitý** (existuje cesta mezi všemi dvojicemi prvků), **konečný** (má maximálně konečný počet vrcholů), **orientovaný** (musí být dána orientace všech hran v grafu), **acyklický** (nesmí tvořit žádný cyklus) a **má jeden počáteční a jeden koncový uzel**.



Obr. 2 – tvoří cyklus
Obr. 3 – není spojitý

4) Uveďte podstatu a vlastnosti metody CPM. Jaké informace nám umožňuje zjistit? (str. 53-57)

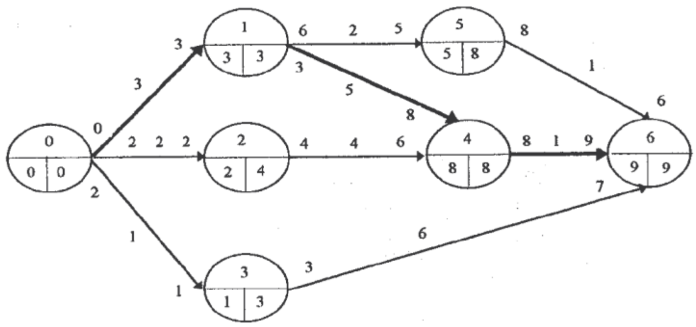
Metoda kritické cesty CPM (Critical path method) – metoda, která se využívá pro řešení úloh projektového řízení v síťových grafech.

Umožňuje zjistit:
- celkovou dobu trvání projektu
- termíny nejdříve možné a nejpozději přípustné doby realizace uzlů
- časové rezervy pro uzly a činnosti - říkají nám, jak se můžeme zdržet v určitém uzlu, nebo protáhnout určitou činnost
- kritickou cestu – posloupnost činností s nulovými časovými rezervami, zdržení by znamenalo zpoždění celého projektu

Metoda CPM je tzv. konjunktivně deterministická

konjunktivní = musí být realizovány všechny činnosti, které dané činnosti předchází, kdyby byl v nějakém bodě souběh více činností, musí se počkat na tu nejpomalejší z nich
deterministická = všechny časové údaje považujeme za pevně dané, např. očekáváme délku trvání nějaké činnosti 10 týdnů a neodhadujeme její délku pouze s nějakou pravděpodobností

5) Popište způsob provedení časové analýzy v metodě CPM (str. 55-56)



Krok 1: Z tabulky činností vytvoříme síťový graf

Krok 2: Rozdělíme uzly na 3 části, viz obrázek výše a do prvního uzlu dáme vždy 0 nalevo.

Krok 3: Přičítáme k levé části uzlu střed šipky (délku trvání činnosti) a výsledek píšeme na konec šipky. Pokud vstupuje do uzlu více šipek, píšeme do levé části uzlu vždy NEJVĚTŠÍ číslo z konců šipek.

Krok 5: Číslo nalevo v posledním uzlu je **délka trvání projektu**, tedy jeden z výsledků úlohy. Toto číslo přepíšu do pravé části uzlu a postupuji pozpátku – viz dále.

Krok 6: Odečítáme od pravé části uzlu střed šipky (délku trvání činnosti) a výsledek píšeme na začátek šipky. Pokud vystupuje z uzlu více šipek, píšeme do pravé části uzlu vždy NEJMENŠÍ číslo ze začátků šipek.

Krok 7: Pokud jsme postupovali správně, musí nám vyjít v pravé části prvního uzlu opět 0.

Krok 8: **Kritická cesta** vede přes činnosti a uzly, kde se konec předchozí šipky rovná začátku následující šipky (tedy přes uzly 0,1,4,6). Všechny časové rezervy viz dále jsou tedy v kritické cestě nulové

Krok 9: Pokud by byla v grafu tzv. fiktivní činnost (čas trvání fiktivní činnosti je 0), kritická cesta by se nezměnila.

Krok 10: Dopočítáme rezervy pro jednotlivé činnosti (všimněme si, že rezervy na kritické cestě jsou všechny nulové☺).

6) Co je to kritická cesta v grafu projektu? Popište způsob, jak ji určíte (str. 55-56)

kritická cesta – je to posloupnost činností, u kterých jsou všechny časové rezervy rovny 0. Ve výsledném síťovém grafu vede přes činnosti a uzly, kde se konec předchozí šipky rovná začátku následující šipky. Vyznačuje se v grafu jinou barvou. Zdržení na kritické cestě znamená zpoždění celého projektu. Pokud umíme v projektu identifikovat kritickou cestu, můžeme pracovní síly z nekritických činností přesouvat na kritické a tím zkrátit délku trvání projektu

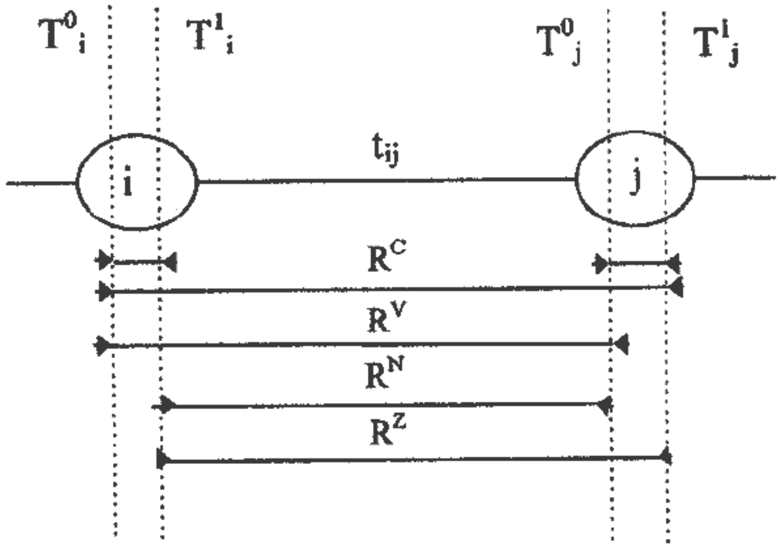
7) Uveďte význam, interpretaci a způsob výpočtu jednotlivých rezerv činností a uzlů v metodě CPM (str. 54-56)

V projektu je vhodné určit pro nekritické činnosti, jakou časovou rezervu dané činnosti mají, aniž by se opozdil projekt. Činnosti, které tvoří kritickou cestu, mají všechny rezervy činností nulové.

t_{ij} - doba trvání činnosti (i, j)
t^0_i - termín nejdříve možného zahájení činnosti (i, j)
t^0_j - termín nejdříve možného ukončení činnosti (i, j)
t^1_i - termín nejpozději možného zahájení činnosti (i, j)
t^1_j - termín nejpozději přípustného ukončení činnosti (i, j)
T^0_i- termín nejdříve možného výskytu počátečního uzlu činnosti (i, j)
T^0_j- termín nejdříve možného výskytu koncového uzlu činnosti (i, j)
T^1_i- termín nejpozději přípustného výskytu počátečního uzlu činnosti (i, j)
T^1_j- termín nejpozději přípustného výskytu koncového uzlu činnosti (i, j)

Přehled časových rezerv

Celková časová rezerva	$R^c_{ij} = T^1_j - T^0_i - t_{ij}$
Volná časová rezerva	$R^V_{ij} = T^0_j - T^0_i - t_{ij}$
Nezávislá časová rezerva	$R^n_{ij} = T^0_j - T^1_i - t_{ij}$
Zvláštní časová rezerva	$R^z_{ij} = T^1_j - T^1_i - t_{ij}$
Interferenční (kritická rezerva uzlu)	$R_i = T^1_i - T^0_i$



1) **<u>CELKOVÁ ČASOVÁ REZERVA</u>** $R^{c}_{ij} = T^{1}_{j} - T^{0}_{i} - t_{ij}$

- udává počet časových jednotek, o který je možné dobu trvání činnosti prodloužit nebo její nejdříve možný začátek oddálit, aniž se tím ovlivní **termín ukončení celého projektu.**

2) **<u>VOLNÁ ČASOVÁREZERVA</u>** $R^{V}_{ij} = T^{0}_{j} - T^{0}_{i} - t_{ij}$

- udává počet časových jednotek, o který je možné dobu trvání činnosti prodloužit nebo její nejdříve možný začátek oddálit, aniž se tím ovlivní **nejdříve možné začátky následujících činností.**

3) **<u>NEZÁVISLÁ REZERVA</u>** $R^{n}_{ij} = T^{0}_{j} - T^{1}_{i} - t_{ij}$

- udává počet časových jednotek, o který je možné dobu trvání činnosti prodloužit nebo její nejdříve možný začátek oddálit, aniž se tím ovlivní **některá z dalších činností.**

4) **<u>ZVLÁŠTNÍ ČASOVÁ REZERVA</u>** $R^{z}_{ij} = T^{1}_{j} - T^{1}_{i} - t_{ij}$

- udává počet časových jednotek, o který můžeme dobu trvání dané činnosti prodloužit nebo její začátek oddálit, aniž by se změnily **nejpozději přípustné začátky následujících činností.**

5) **<u>INTERFERENČNÍ REZERVA UZLU</u>** $R_{i} = T^{1}_{i} - T^{0}_{i}$

- udává zpoždění, které si můžeme dovolit mezi koncem předchozí a začátkem následující činnosti.