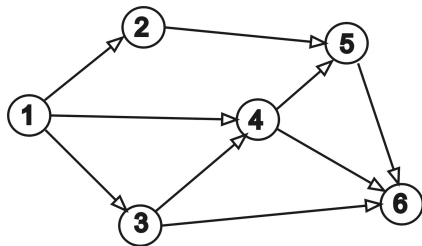
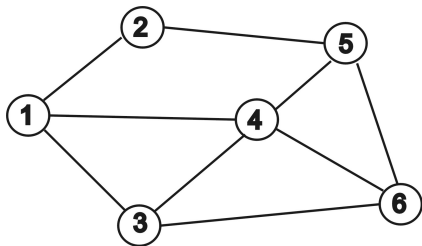


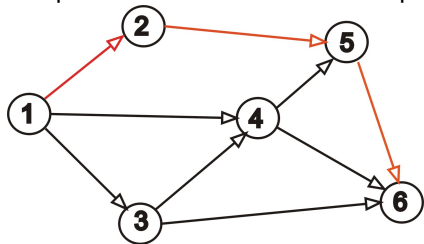
Optimalizace v grafech - základní pojmy

Řadu reálných systémů (např. distribuční síť, projekt, rozmístění stanovišť v podniku) lze modelovat rovinnými grafy. Graf je tvořen **uzly**, které budeme značit u_1, \dots, u_n a **hranami**, přičemž hranu mezi uzly u_i a u_j označíme h_{ij} . V rovině můžeme znázornit graf pomocí bodů (koleček) a spojnic mezi nimi. Hranu, které umožňují pohyb v obou směrech nazýváme **neorientované**. Je-li povolen pouze jeden směr pohybu, znázorníme to na grafu šipkou a takové "jednosměrné"hrany nazýváme **orientované**. Neorientovaným grafem nazveme graf obsahující pouze neorientované hrany, jinak jej nazveme orientovaným. Na obrázku je znázorněn neorientovaný a orientovaný graf.



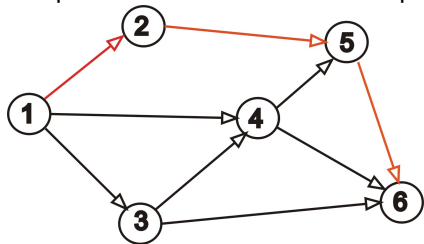
Optimalizace v grafech - základní pojmy

Cestou z uzlu u_i do uzlu u_j nazveme posloupnost na sebe navazujících hran, z nichž první začíná v u_i a poslední končí v u_j . Pokud cesta respektuje orientaci hran, nazývá se orientovaná (v opačném případě neorientovaná). Na obrázku je znázorněna jedna z orientovaných cest z uzlu 1 do uzlu 6. Naopak z uzlu 6 do uzlu 1 vedou pouze neorientované cesty.



Optimalizace v grafech - základní pojmy

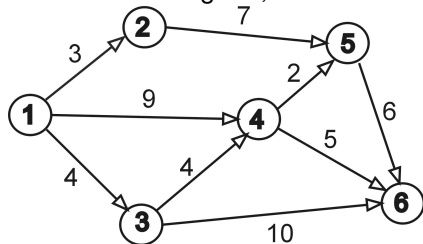
Cestou z uzlu u_i do uzlu u_j nazveme posloupnost na sebe navazujících hran, z nichž první začíná v u_i a poslední končí v u_j . Pokud cesta respektuje orientaci hran, nazývá se orientovaná (v opačném případě neorientovaná). Na obrázku je znázorněna jedna z orientovaných cest z uzlu 1 do uzlu 6. Naopak z uzlu 6 do uzlu 1 vedou pouze neorientované cesty.



Cestu, pro kterou $u_i = u_j$, nazveme **cyklus**, v případě neorientovaného grafu **kružnice**. Zobrazený graf obsahuje například kružnici 1 – 3 – 4 – 1. Graf, ve kterém mezi libovolnými dvěma uzly existuje aspoň jedna neorientovaná cesta, se nazývá **souvislý**. Každý souvislý neorientovaný graf, který neobsahuje kružnici, se nazývá **strom**.

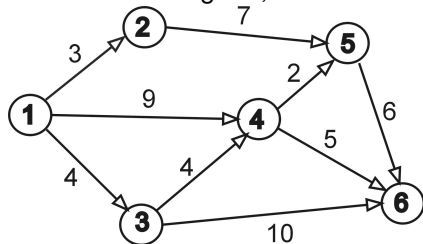
Optimalizace v grafech - základní pojmy

Při řešení optimalizačních úloh zpravidla pracujeme s **hranově ohodnocenými** grafy. Hranám jsou přiřazeny hodnoty y_{ij} podle ekonomického významu (např. vzdálenosti mezi distribučními centry či náklady na přepravu mezi centry, apod.) Souvislý orientovaný a nezáporně ohodnocený graf se dvěma speciálními uzly (vstupem a výstupem) nazveme **síť**. Přidáme-li ohodnocení hran do našeho grafu, získáme síť se vstupním uzlem 1 a výstupním uzlem 6.



Optimalizace v grafech - základní pojmy

Při řešení optimalizačních úloh zpravidla pracujeme s **hranově ohodnocenými** grafy. Hranám jsou přiřazeny hodnoty y_{ij} podle ekonomického významu (např. vzdálenosti mezi distribučními centry či náklady na přepravu mezi centry, apod.) Souvislý orientovaný a nezáporně ohodnocený graf se dvěma speciálními uzly (vstupem a výstupem) nazveme **sít'**. Přidáme-li ohodnocení hran do našeho grafu, získáme síť se vstupním uzlem 1 a výstupním uzlem 6.



Délkou cesty nazveme součet ohodnocení jejích hran. Například mezi délka orientované cesty 1-2-5-6 je $3+7+6=16$. **Pozor!** Graf je definován pomocí množiny uzlů a množiny hran, nikoliv zakreslením. Délky spojnic nemusí a často ani nemohou odpovídat ohodnocení hran.

Optimalizace v grafech - úlohy

Na grafech se řeší řada úloh:

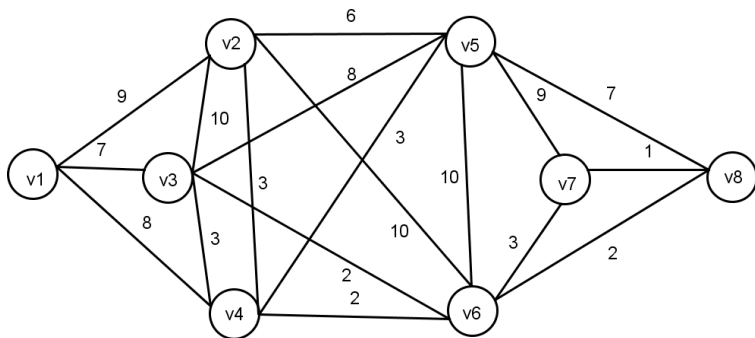
- Standartní optimalizační úlohou je hledání **nejkratší cesty** mezi dvěma uzly. Úloha se řeší v orientovaných i neorientovaných grafech. existuje více algoritmů, některé k určení celé matice vzdáleností. Jeden z nejznámějších algoritmů je Dijkstrův algoritmus.
- Hledání **minimální kostry** grafu - úkolem je vybrat takovou podmnožinu hran, aby mezi každými dvěma uzly existovala cesta a aby celkové ohodnocení bylo minimální (kostra nesmí obsahovat cyklus).
- Určení **maximálního toku** v síti (propustnosti sítě): Představuje - li ohodnocení v síti přepravní kapacitu hran, pak úkolem je určení maximálního počtu jednotek, které je možné přepravit ze vstupního do výstupního uzlu.
- Další úlohy, jako problém barvení grafu, problém čínského pošťáka, problém obchodního cestujícího, medián grafu, centr grafu, atd.

Optimalizace v grafech - minimální kostra

Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezněme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:

Optimalizace v grafech - minimální kostra

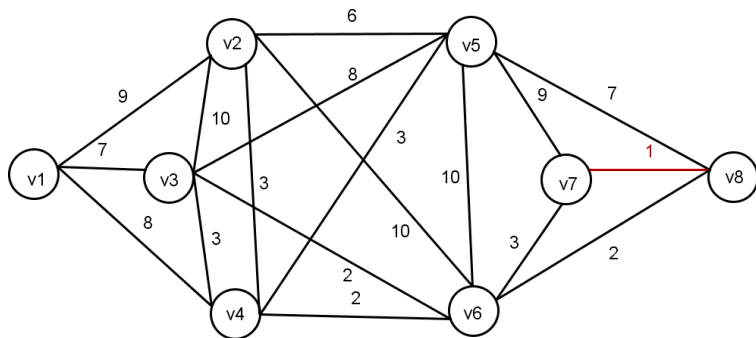
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezněme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Nejprve přidáme hranu s hodnotou 1.

Optimalizace v grafech - minimální kostra

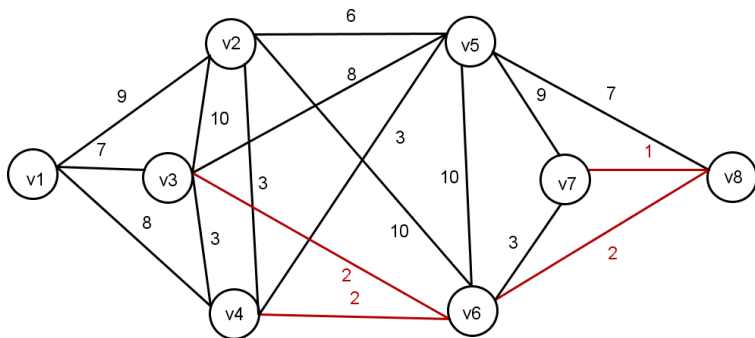
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezněme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Dále vybereme všechny hrany s hodnotou 2.

Optimalizace v grafech - minimální kostra

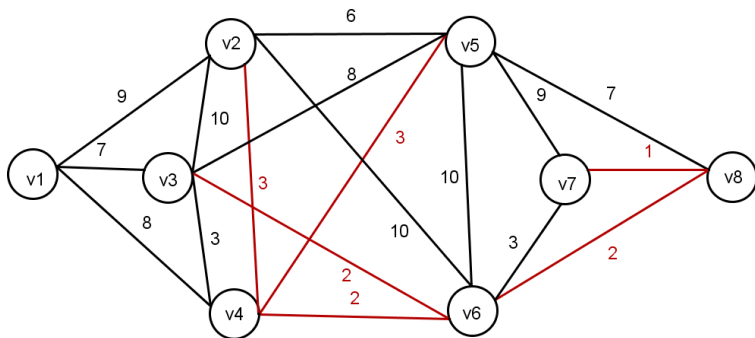
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezneme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Přidáme hrany s hodnotou 3 kromě v3-v4, která by uzavřela okruh.

Optimalizace v grafech - minimální kostra

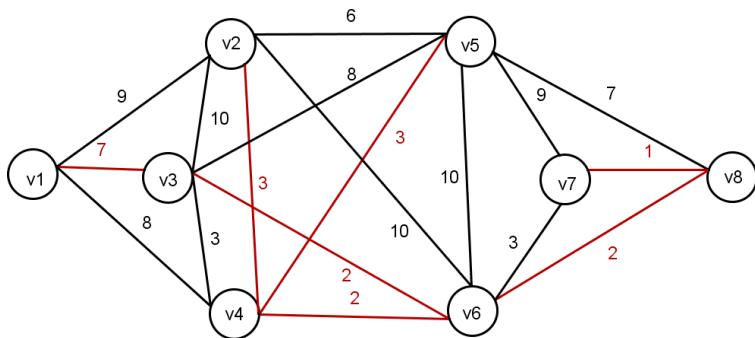
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezneme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Hrana s hodnotou 6 nemůže být do kostry zahrnuta, přidáme tedy hranu s hodnotou 7.

Optimalizace v grafech - minimální kostra

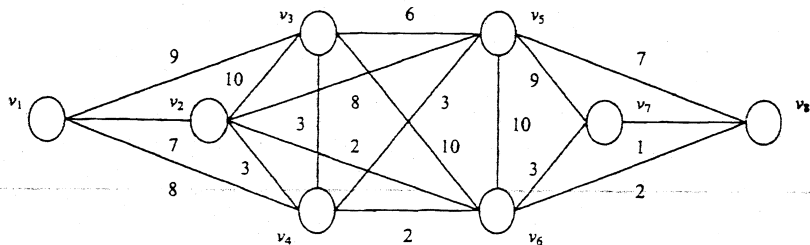
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezneme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Dostali jsem již souvislý podgraf, tedy kostru, celková hodnota jejích hran je 20.

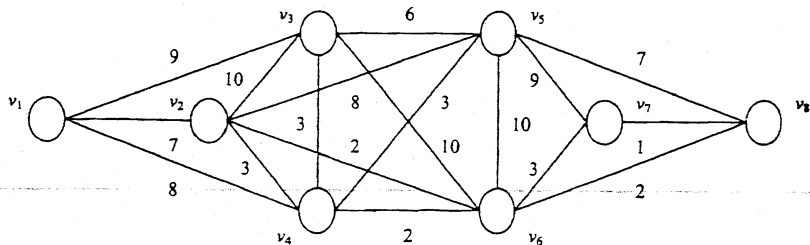
Optimalizace v grafech - nejkratší cesta

Pro hledání nejkratší cesty z uzlu v_1 do ostatních uzlů lze použít **Dijkstrův algoritmus**: předvedme na grafu z předchozího příkladu:



Optimalizace v grafech - nejkratší cesta

Pro hledání nejkratší cesty z uzlu v_1 do ostatních uzlů lze použít **Dijkstrův algoritmus**: předvedme na grafu z předchozího příkladu:



Algoritmus rozděljuje uzly podle toho, zda už do nich nejkratší cestu známe nebo ne. Začínáme s cestou délky 0 z v_1 do v_1 . Nejkratší cestu prodloužíme vždy o jednu hranu. Projdeme všechny uzly se známou délkou cesty z v_1 , sečteme tyto délky postupně s hodnotami hran z uzlů vycházejících a vybereme ze všech součtů nejmenší, tak získáme délku cesty do nějakého nového uzlu a postup dále opakujeme.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0							

V prvním kroku známe jen vzdálenost do v1, projdeme jeho sousedy a vybereme hranu s nejmenší hodnotou, což je v1-v2 s délkou 7. Vrchol v2 tedy zařadíme do skupiny se známou vzdáleností z v1.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7						

V druhém kroku procházíme kromě sousedů v1 také sousedy v2, délky cest přes v2 jsou 7+10, 7+3, 7+8, 7+2, takže nejkratší novou cestou bude v1-v4 délky 8. Přidáme v4 mezi uzly, do nichž nejkratší cestu známe.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7		8				

V dalším kroku procházíme dosud nezařazené sousedy uzlů v1, v2 a v4. Nejmenší celková délka cest přes tyto uzly do jejich sousedů je v1-v3 s hodnotou 9. To bude tedy vzdálenost do uzlu v3.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8				

Stejnou vzdálenost (také 9) má cesta přes v2 do v6.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8		9		

Z dosud nezařazených uzlů má nejmenší celkovou vzdálenost přes jednu hranu z již vytvořených cest uzel v5, který napojíme z v4, jeho vzdálenost z v1 je 11.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8	11	9		

Stejně vzdálenosti 11 dosáhneme napojením uzlu v8 přes v6.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8	11	9		11

Nakonec připojíme uzel v7, nejkratší možnost je přes v6, celková vzdálenost do něj je 12.

Optimalizace v grafech - medián grafu

Medián grafu minimalizuje součet vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění skladu pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	SUMA
Kralupy	0	4	16	8	18	25	17	88
Veltrusy	4	0	20	12	22	28	13	99
Slaný	16	20	0	12	7	14	17	86
Velvary	8	12	12	0	10	17	10	69
Zlonice	18	22	7	10	0	7	10	74
Vraný	25	28	14	17	7	0	15	106
Bříza	17	13	17	10	10	15	0	82

Optimalizace v grafech - medián grafu

Medián grafu minimalizuje součet vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění skladu pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	SUMA
Kralupy	0	4	16	8	18	25	17	88
Veltrusy	4	0	20	12	22	28	13	99
Slaný	16	20	0	12	7	14	17	86
Velvary	8	12	12	0	10	17	10	69
Zlonice	18	22	7	10	0	7	10	74
Vraný	25	28	14	17	7	0	15	106
Bříza	17	13	17	10	10	15	0	82

Aby se do skladu nejezdilo celkem co nejméně kilometrů, je nejlepší jej umístit ve Velvarech.

Pozn.: Při umístění dvou skladů hovoříme o dvoumediánu, atd.

Optimalizace v grafech - centr grafu

Centr grafu minimalizuje maximum vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění hasičské stanice pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	MAX
Kralupy	0	4	16	8	18	25	17	25
Veltrusy	4	0	20	12	22	28	13	28
Slaný	16	20	0	12	7	14	17	20
Velvary	8	12	12	0	10	17	10	17
Zlonice	18	22	7	10	0	7	10	22
Vraný	25	28	14	17	7	0	15	28
Bříza	17	13	17	10	10	15	0	17

Optimalizace v grafech - centr grafu

Centr grafu minimalizuje maximum vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění hasičské stanice pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	MAX
Kralupy	0	4	16	8	18	25	17	25
Veltrusy	4	0	20	12	22	28	13	28
Slaný	16	20	0	12	7	14	17	20
Velvary	8	12	12	0	10	17	10	17
Zlonice	18	22	7	10	0	7	10	22
Vraný	25	28	14	17	7	0	15	28
Bříza	17	13	17	10	10	15	0	17

Aby města byla pro hasiče co nejrychleji dostupná, nejlepší je umístit stanici ve Velvarech nebo Bříze.

Pozn.: Při umístění dvou stanic hovoříme o dvoucentru, atd.

Optimalizace v síťových grafech

Nejznámější úlohou je hledání maximálního toku v síti (např. vodovodní potrubí, silniční či datová síť,...). Pro řešení potřebujeme znát:

- popis potrubí (graf, zpravidla orientovaný)
- odkud voda vytéká (speciální uzel: zdroj)
- kam voda teče (speciální uzel: cíl, spotřebič)
- kolik vody může danou hranou protéct (kapacita = nazáporné ohodnocení hran)

Graf s vyjmenovanými vlastnostmi označujeme jako **síť**.

Optimalizace v síťových grafech

Nejznámější úlohou je hledání maximálního **toku v síti** (např. vodovodní potrubí, silniční či datová síť,...). Pro řešení potřebujeme znát:

- popis potrubí (graf, zpravidla orientovaný)
- odkud voda vytéká (speciální uzel: zdroj)
- kam voda teče (speciální uzel: cíl, spotřebič)
- kolik vody může danou hranou protéct (kapacita = nazáporné ohodnocení hran)

Graf s vyjmenovanými vlastnostmi označujeme jako **síť**.

Tokem nazveme funkci přiřazující každé hraně nezáporné číslo nepřesahující kapacitu splňující tzv. Kirchhoffovy zákony: **součet toků na hranách vstupujících je stejný jako na vystupujících**. Tyto podmínky musí platit ve všech uzlech kromě zdroje a cíle. **Velikost toku** je součet toků vycházejících ze zdroje. Pokud ze zdroje nic nevytéká, jde o nulový tok.

Optimalizace v síťových grafech

Pro hledání maximálního toku v síti se používají zlepšovací algoritmy: začne se zpravidla s nulovým tokem a hledá se cesta, na níž mají všechny hrany nenulovou rezervu (rozdíl kapacity a skutečného toku). Tok lze navýšit na dané cestě o minimální hodnotu rezervy. Postupujeme dále prohledáváním grafu a hledáním další zlepšující cesty. Prostou aplikací tohoto postupu bychom ale nemuseli nalézt nejlepší řešení, někdy je potřeba na některých hranách naopak tok zmenšit, aby se mohl vést lepší cestou a celková velikost toku se zvýšila.

Optimalizace v síťových grafech

Pro hledání maximálního toku v síti se používají zlepšovací algoritmy: začne se zpravidla s nulovým tokem a hledá se cesta, na níž mají všechny hrany nenulovou rezervu (rozdíl kapacity a skutečného toku). Tok lze navýšit na dané cestě o minimální hodnotu rezervy. Postupujeme dále prohledáváním grafu a hledáním další zlepšující cesty. Prostou aplikací tohoto postupu bychom ale nemuseli nalézt nejlepší řešení, někdy je potřeba na některých hranách naopak tok zmenšit, aby se mohl vést lepší cestou a celková velikost toku se zvýšila. Vylepšený algoritmus (**Ford-Fulkerson**) je následující:

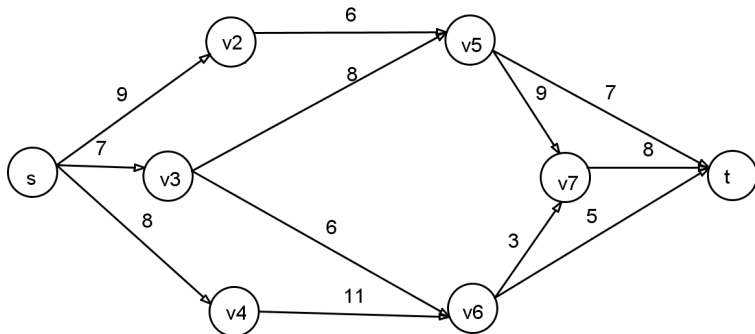
- U každé hrany evidujeme její rezervu i "rezervu v protisměru" (o kolik lze tok snížit). Na začátku u výchozího nulového toku jsou rezervy rovny kapacitám (rezervy v protisměru u orientovaných hran nulové)
- Vybereme zlepšující cestu a navýšíme tok o její rezervu. Současně o stejnou hodnotu snížíme rezervy všech hran na cestě a navýšíme jejich rezervy v protisměru.
- Postupujeme takto dál, dokud lze nalézt cestu s nenulovou rezervou.

Maximální tok v síti: Ford-Fulkerson

Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:

Maximální tok v síti: Ford-Fulkerson

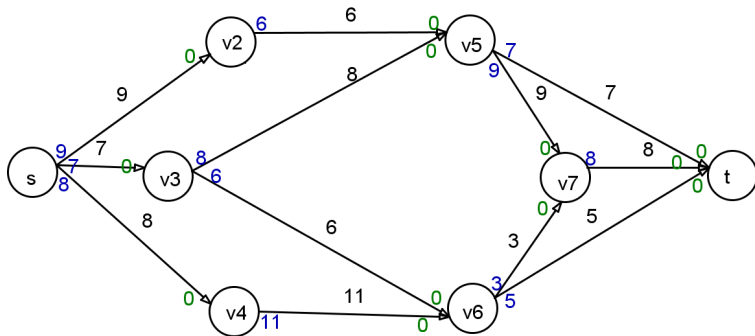
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Začneme s nulovým tokem, na hranách modře vyznačíme rezervy, zeleně "rezervy v protisměru".

Maximální tok v síti: Ford-Fulkerson

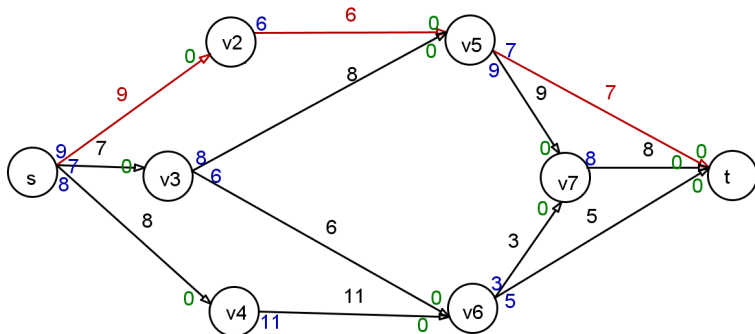
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Najdeme libovolnou zlepšující cestu.

Maximální tok v síti: Ford-Fulkerson

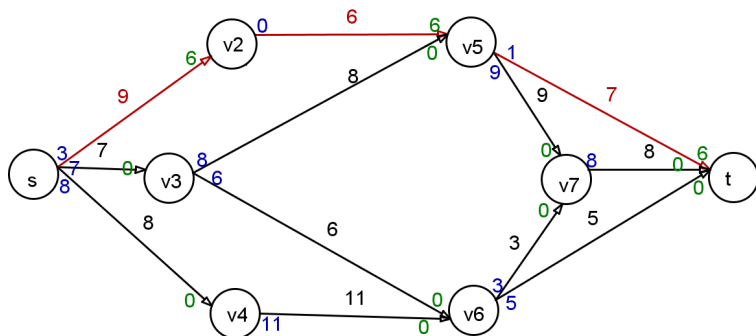
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Na zvolené cestě je nejmenší rezerva 6, o tuto hodnotu navýšíme tok na cestě. Rezervy na všech vybraných hranách tedy klesnou o šest a o stejnou hodnotu se zvýší protismerné rezervy.

Maximální tok v síti: Ford-Fulkerson

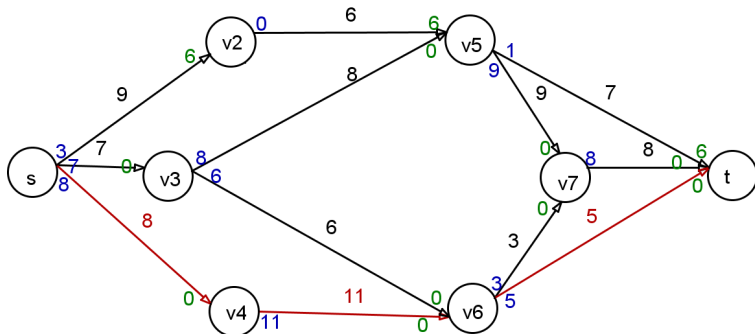
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Najdeme další zlepšující cestu.

Maximální tok v síti: Ford-Fulkerson

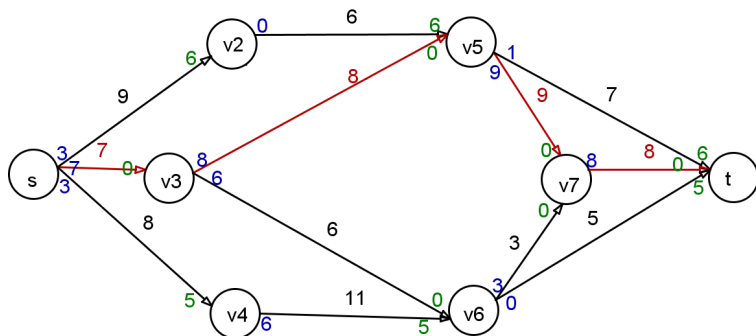
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Na této cestě je hrana s rezervou 5, nemůžeme tedy tok zvýšit o víc.

Maximální tok v síti: Ford-Fulkerson

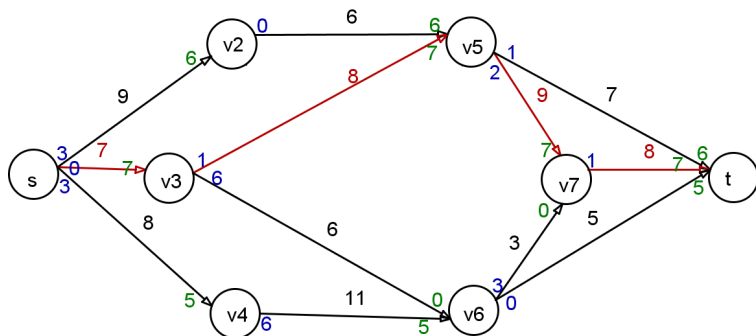
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Po navýšení najdeme další cestu s nenulovou rezervou.

Maximální tok v síti: Ford-Fulkerson

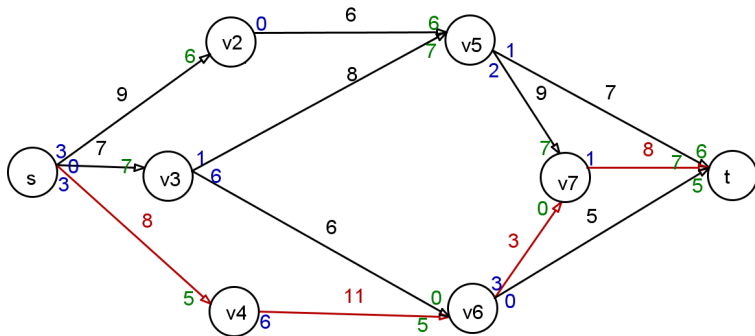
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Na této cestě jsme navýšili tok o 7.

Maximální tok v síti: Ford-Fulkerson

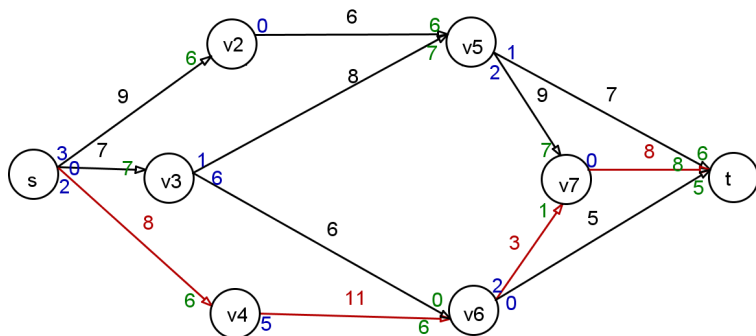
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Existuje ještě zlepšující cesta s rezervou 1.

Maximální tok v síti: Ford-Fulkerson

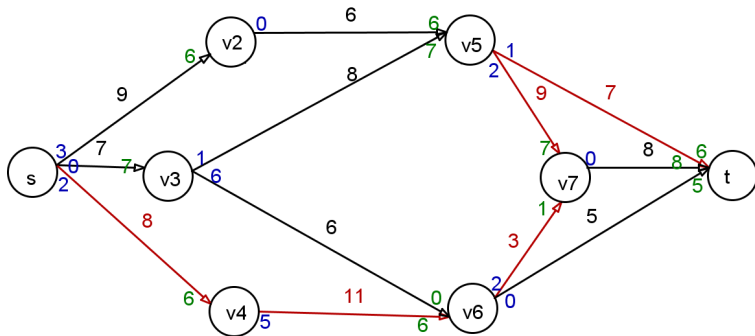
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Provedli jsme navýšení.

Maximální tok v síti: Ford-Fulkerson

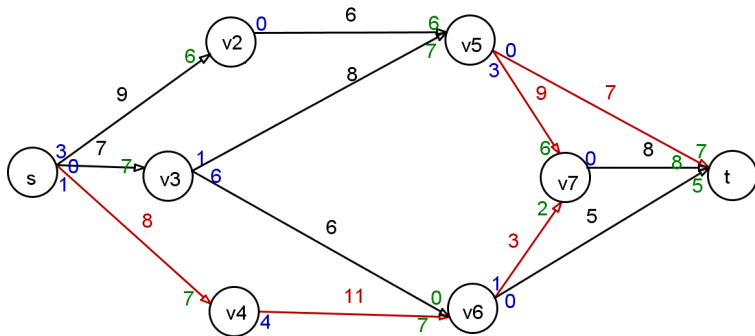
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Můžeme tok ještě zlepšit po cestě, která mezi v_5 a v_7 vede "v protisměru".

Maximální tok v síti: Ford-Fulkerson

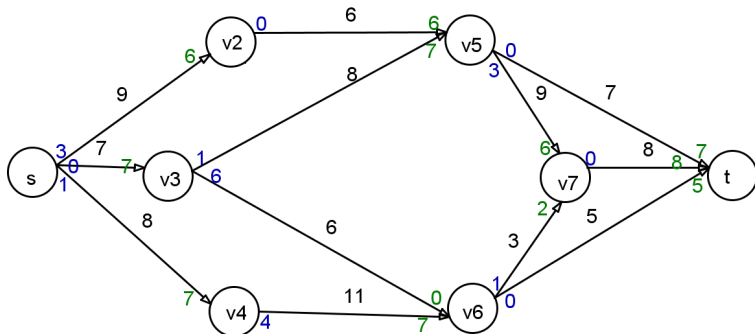
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Provedeme navýšení o nejmenší rezervu (1, na poslední hraně). Mezi v_5 a v_7 se tok fakticky snížil o 1.

Maximální tok v síti: Ford-Fulkerson

Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Nalezený tok má velikost $7+8+5=20$ (viz zelené hodnoty u cíle). Evidentně je to optimální řešení, do cíle už není volná další kapacita.

Maximální tok v síti: Ford-Fulkerson

Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:

Řízení projektů

Řízení projektů je jednou z typických aplikací teorie grafů. Projekt můžeme obecně chápat jako soubor činností. Tyto činnosti lze charakterizovat předpokládanou dobou trvání, náklady na realizaci, požadavky na zajištění, výčtem činností, které realizaci musí předcházet, atd. Nejčastěji hledáme odpověď na tyto otázky:

- Jaká je nejkratší možná doba realizace projektu?
- Které činnosti jsou z hlediska dodržení termínu klíčové, tzv. **kritické činnosti**?
- Jaké jsou rezervy u nekritických činností
- Jaký je časový rozvrh pro realizaci jednotlivých činností?

Kromě **časové analýzy** projektů nás zajímají též náklady na projekt v závislosti na čase, hovoříme pak o **nákladové analýze**. Dále můžeme sledovat úroveň a rozložení zdrojů potřebných pro jednotlivé činnosti, tedy provádět **zdrojovou analýzu** projektu.

Řízení projektů - konstrukce síťového grafu

Projekt znázorňujeme pomocí ohodnoceného síťového grafu, kde hrany reprezentují činnosti, ohodnocení většinou dobu jejich trvání a uzly představují momenty zahájení či ukončení jednotlivých činností. Při analýze nejprve musíme vymezit jednotlivé činnosti, odhadnout délky jejich realizace a definovat **návaznosti** pomocí výčtu bezprostředně předcházejících činností.

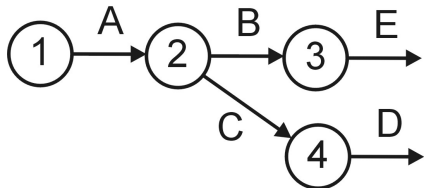
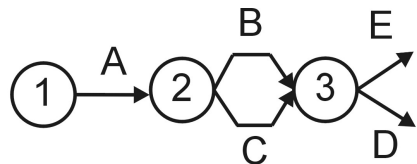
Řízení projektů - konstrukce síťového grafu

Projekt znázorňujeme pomocí ohodnoceného síťového grafu, kde hrany reprezentují činnosti, ohodnocení většinou dobu jejich trvání a uzly představují momenty zahájení či ukončení jednotlivých činností. Při analýze nejprve musíme vymezit jednotlivé činnosti, odhadnout délky jejich realizace a definovat **návaznosti** pomocí výčtu bezprostředně předcházejících činností. Ukažme si síťový graf pro projekt vytvoření nového obchodního střediska firmy Q-mark, a.s. (J.Jablonský: Operační výzkum). Před vlastním sestavením grafu definujeme elementární činnosti a jejich vlastnosti, viz. tabulka:

činnost	popis činnosti	trvání [týdny]	předchází
A	výběr a nákup projektu	6	-
B	zpracování projektu	4	A
C	obsazení pozice manažera	3	A
D	výběr personálu	3	B,C
E	rekonstrukce a vybavení objektu	8	B
F	školení personálu	2	D
G	výběr sortimentu zboží	2	B,C
H	uzavření smluv s dodavateli	5	G
I	nákup zboží	3	E,F,H
J	reklama	2	H

Řízení projektů - konstrukce síťového grafu

Při sestavování grafu můžeme narazit na problém u definice uzlů. Například činnosti D musí předcházet B,C, ale činnosti E předchází pouze B. Jak tedy správně znázornit návaznost? Příklady **nesprávného** znázornění, viz obr.:



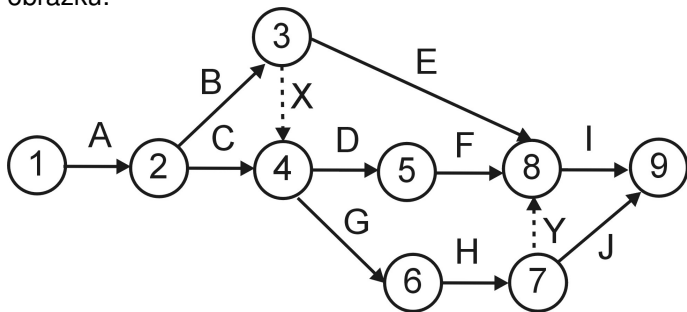
Podobný problém je u činností I,J, které mají společného předchůdce H, ale činnosti I navíc předchází i E,F.

Řízení projektů - konstrukce síťového grafu

Problém s návazností lze řešit zavedením **fiktivních činností**, jimž odpovídající fiktivní hrany doplní chybějící spoje. Znázorňují se přerušovanou čarou. Doba fiktivních činností je vždy nulová. Do našeho příkladu tedy doplníme dvě fiktivní činnosti: X zprostředkující návaznost mezi B a D a činnost Y pro návaznost mezi H a I.

Řízení projektů - konstrukce síťového grafu

Problém s návazností lze řešit zavedením **fiktivních činností**, jimž odpovídající fiktivní hrany doplní chybějící spoje. Znárodnují se přerušovanou čarou. Doba fiktivních činností je vždy nulová. Do našeho příkladu tedy doplníme dvě fiktivní činnosti: X zprostředkující návaznost mezi B a D a činnost Y pro návaznost mezi H a I. Jedno z možných znázornění sítě projektu je na obrázku:



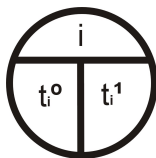
Při sestavování grafu je dobré držet se pravidla, aby index počátečního uzlu každé hrany byl nižší než index jejího koncového uzlu. Potom graf nebude obsahovat žádné orientované cykly.

Critical Path Method (CPM)

Metoda se používá od 50. let minulého století. Jejím principem je určit pro každou činnost tyto 4 charakteristiky:

- Nejdříve možný začátek provádění činnosti začínající v uzlu u_i , značíme jej: t_i^0
- Nejdříve možný konec provádění činnosti reprezentované hranou h_{ij} získáme přičtením doby trvání činnosti: $t_i^0 + y_{ij}$
- Nejpozději přípustný konec provádění činnosti končící v uzlu u_j , značíme jej: t_j^1
- Nejpozději přípustný začátek provádění činnosti reprezentované hranou h_{ij} získáme odečtením doby trvání činnosti: $t_j^1 - y_{ij}$

V síťovém grafu vyznačíme do jednotlivých uzlů i nejdříve možné začátky a nejpozději přípustné konce činností, které v něm začínají, resp. končí:



Critical Path Method (CPM)

Vlastní algoritmus metody probíhá ve čtyřech fázích:

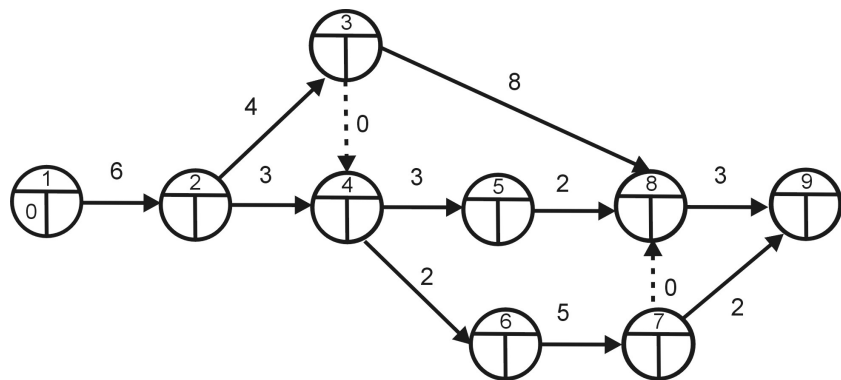
- 1 Výpočet nejprve možných začátků: Pro činnosti začínající v uzlu u_j se spočte jako maximum nejdříve možných konců činností, které do něj vstupují. $t_j^0 = \max_i(t_i^0 + y_{ij})$ Pro výstupní uzel tak dostaneme nejkratší možnou dobu realizace projektu.
- 2 Výpočet nejpozději přípustných konců: Pro činnosti končící v uzlu u_i se spočte jako minimum z nejpozději přípustných začátků činností z uzlu vystupujících. $t_i^1 = \min_j(t_j^1 - y_{ij})$
- 3 Výpočet celkových časových rezerv: Činnost reprezentovaná hranou h_{ij} má stanoveno, kdy může nejdříve začít (t_i^0) a kdy musí nejpozději skončit (t_j^1). Doba, během které se musí realizovat, je tedy $t_j^1 - t_i^0$ a protože její realizace trvá y_{ij} , dostaneme její pro časovou rezervu vztah:
 $R_{ij} = t_j^1 - t_i^0 - y_{ij}$
- 4 Sestavení harmonogramu činností

Critical Path Method (CPM) - 1. fáze

V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.

Critical Path Method (CPM) - 1. fáze

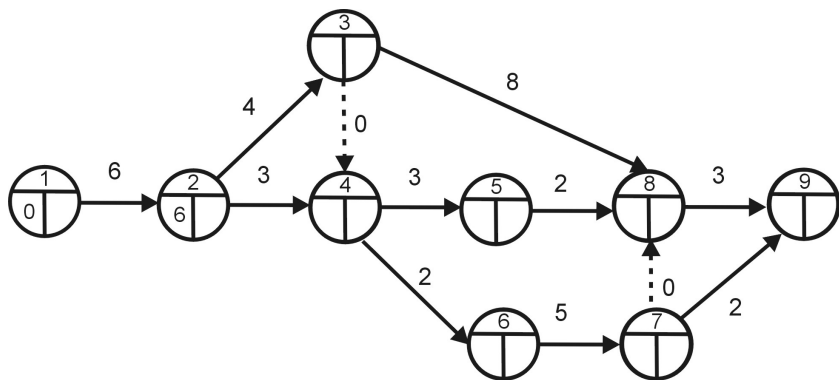
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejprve si zobrazíme graf projektu s dobami trvání činností.

Critical Path Method (CPM) - 1. fáze

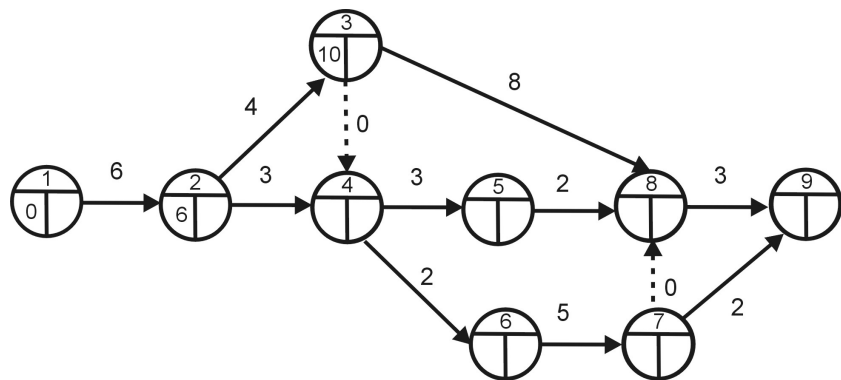
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činností B,C je $t_2^0 = 0 + 6 = 6$.

Critical Path Method (CPM) - 1. fáze

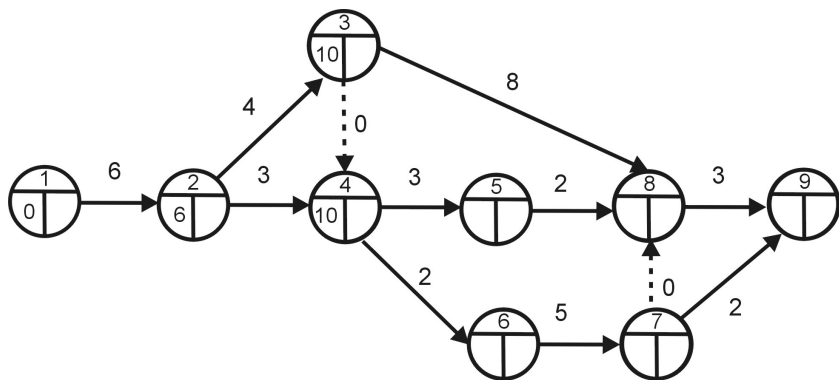
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činnosti E je $t_3^0 = 6 + 4 = 10$.

Critical Path Method (CPM) - 1. fáze

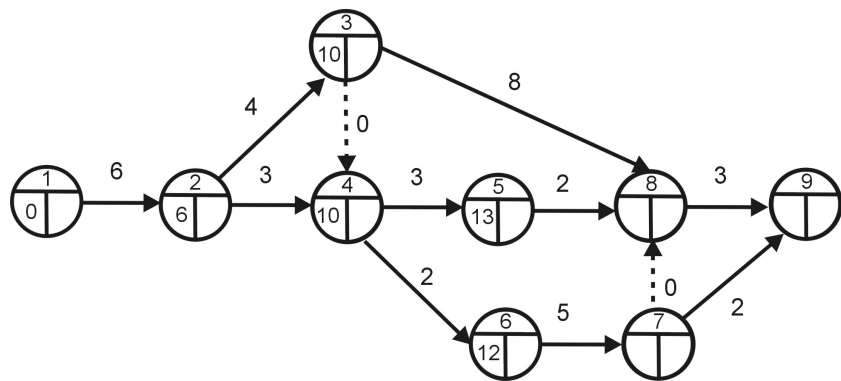
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činností D,G je $t_4^0 = \max(10 + 0, 6 + 3) = 10$.

Critical Path Method (CPM) - 1. fáze

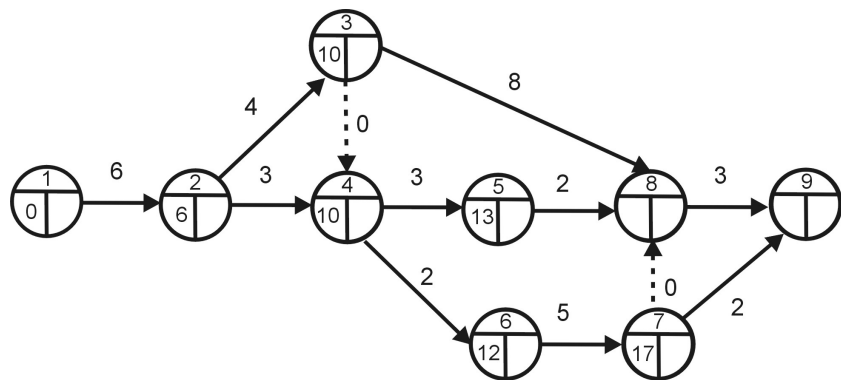
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činností F, resp. H je $t_5^0 = 10 + 3 = 13$, resp. $t_6^0 = 10 + 2 = 12$.

Critical Path Method (CPM) - 1. fáze

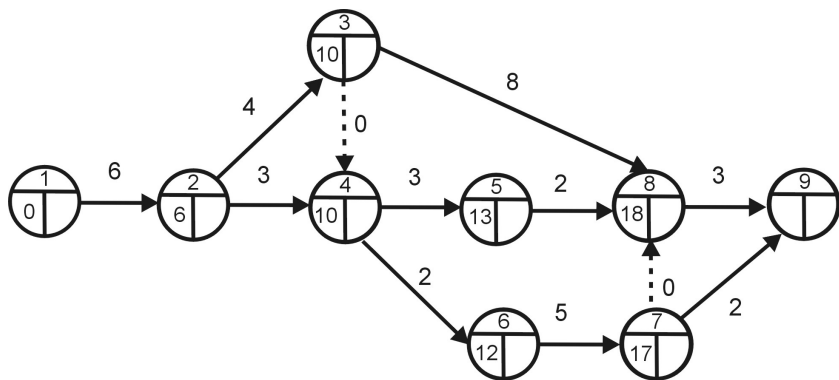
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činnosti J je $t_7^0 = 12 + 5 = 17$.

Critical Path Method (CPM) - 1. fáze

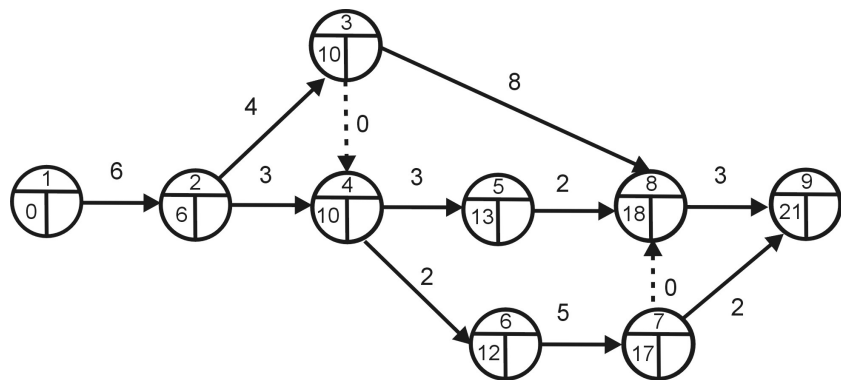
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činnosti I je $t_8^0 = \max(10 + 8, 13 + 2, 17 + 2) = 18$.

Critical Path Method (CPM) - 1. fáze

V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



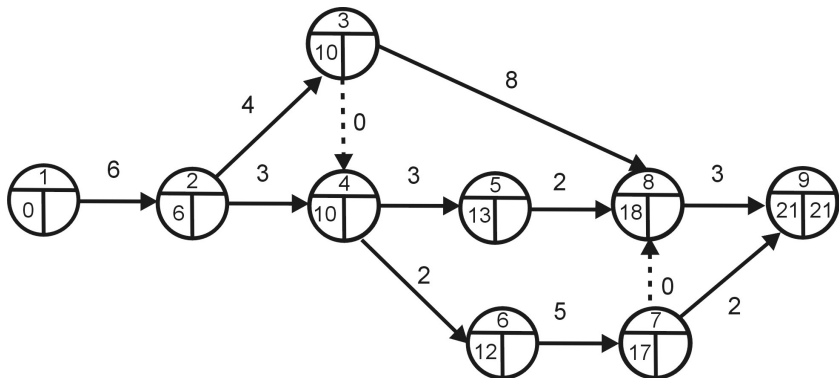
Nejdříve možná doba ukončení projektu je v čase
 $t_9^0 = \max(18 + 3, 17 + 2) = 21$ týdnů.

Critical Path Method (CPM) - 2. fáze

Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.

Critical Path Method (CPM) - 2. fáze

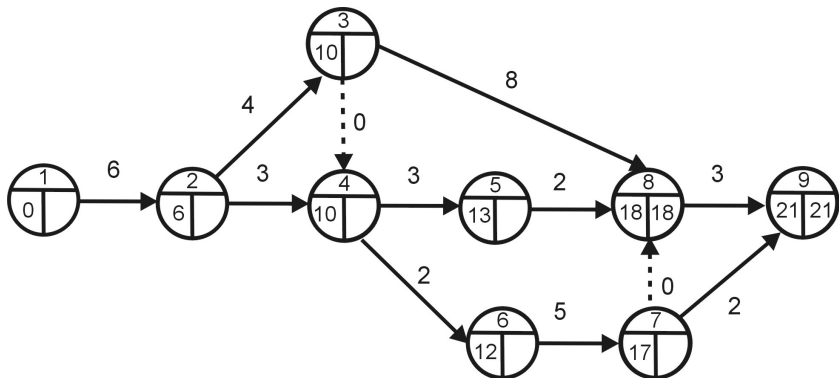
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Stanovili jsme minimální dobu trvání projektu na 21 týdnů. To bude i nejpozději přípustný konec činností I,J: $t_9^1 = 21$.

Critical Path Method (CPM) - 2. fáze

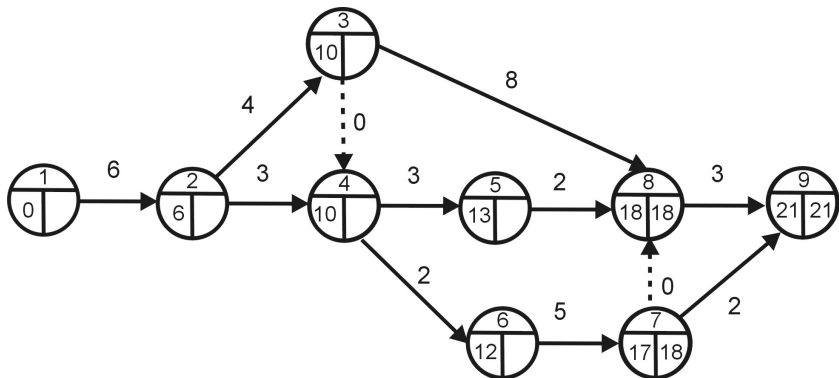
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činností F,E je $t_8^1 = 21 - 3 = 18$.

Critical Path Method (CPM) - 2. fáze

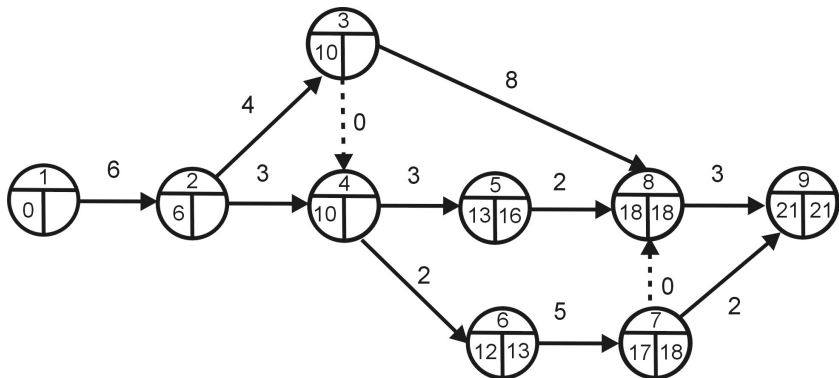
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti H je $t_7^1 = \min(21 - 2, 18 - 0) = 18$.

Critical Path Method (CPM) - 2. fáze

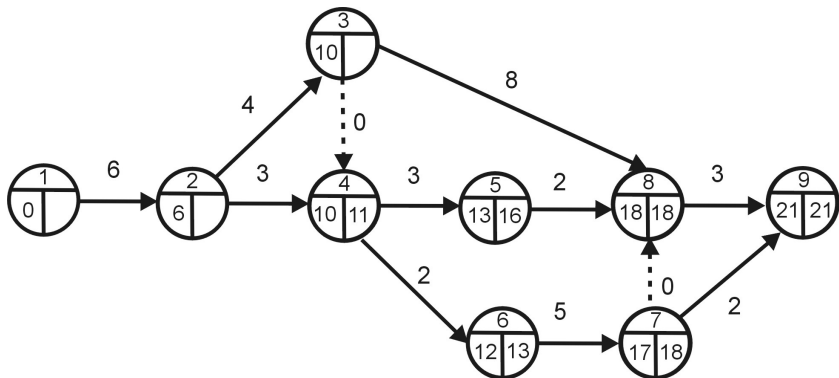
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činností D, resp. G je $t_5^1 = 18 - 2 = 16$, resp. $t_6^1 = 18 - 5 = 13$.

Critical Path Method (CPM) - 2. fáze

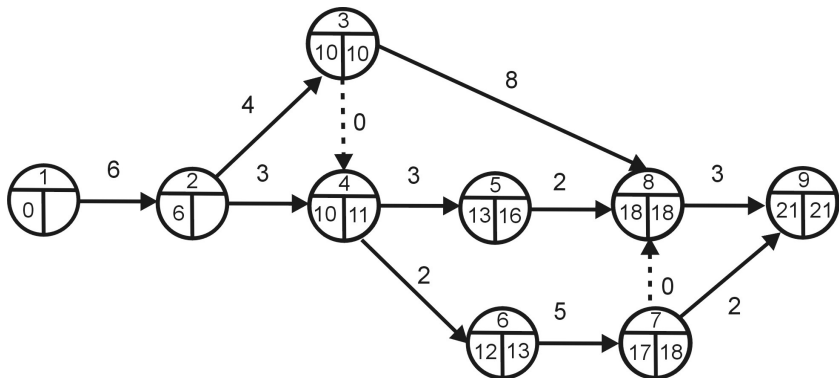
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti C je $t_4^1 = \min(13 - 2, 16 - 3) = 11$.

Critical Path Method (CPM) - 2. fáze

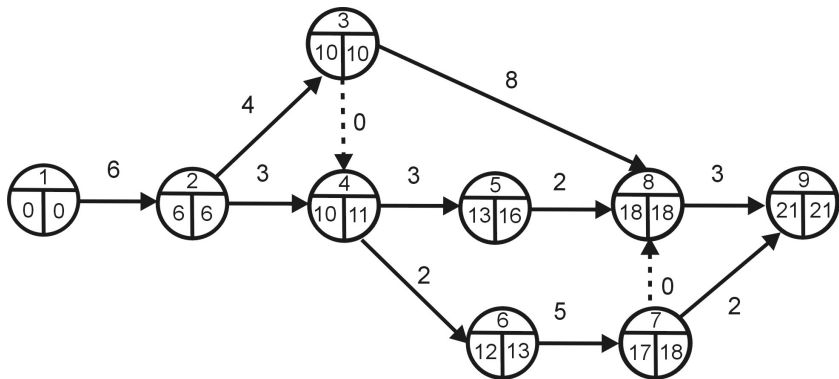
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti B je $t_3^1 = \min(18 - 8, 11 - 0) = 10$.

Critical Path Method (CPM) - 2. fáze

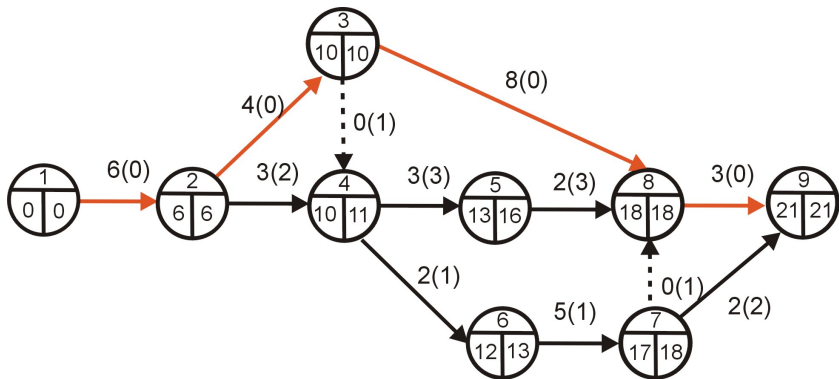
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti A je $t_2^1 = \min(10 - 4, 11 - 3) = 6$. Protože jsme vycházeli z nejrychlejší možné realizace projektu, samozřejmě vyšlo $t_1^1 = 6 - 6 = 0$.

Critical Path Method (CPM) - 3. fáze

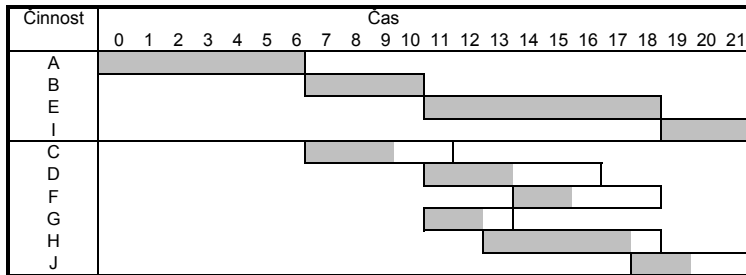
Dopočítáme rezervy jednotlivých činností podle vztahu $R_{ij} = t_j^1 - t_i^0 - y_{ij}$.
Například činnost J musí proběhnout mezi 17. a 21. týdnem a trvá 2 týdny, její rezerva je tedy $R_{79} = 21 - 17 - 2 = 2$ týdny, apod.



Rezervy jednotlivých činností jsou uvedeny v závorkách. Červeně je znázorněna kritická cesta sestávající z činností A,B,E,I, které nemají žádnou časovou rezervu.

Critical Path Method (CPM) - 4. fáze

Poslední, ale velmi důležitou fází je rozvržení realizace činností v čase. Je nutné určit, které činnosti mohou probíhat paralelně a které na sebe musí navazovat. To nám umožní dále rozvrhovat zdroje potřebné pro jednotlivé činnosti. Ukažme si rozvrh činností v diagramu, kde rámečky tvoří nejdříve možný začátek a nejpozději přípustný konec činností, stínování naznačuje dobu jejich trvání. V horní části tabulky jsou kritické činnosti s nulovými rezervami - tyto musí na sebe bezprostředně navazovat, aby nedošlo ke zpoždění projektu.



Metoda PERT

Metoda **PERT** (Program Evaluation and Review Technique) je pravděpodobnostním rozšířením metody CPM. V praxi často není reálné stanovit realizační doby činností, proto jsou hodnoty y_{ij} nahrazeny náhodnými veličinami, které se realizují na nějakém intervalu $\langle a_{ij}, b_{ij} \rangle$. Kromě tohoto mezního **optimistického**, resp. **pesimistického** odhadu se pracuje s nejpravděpodobnější dobou trvání činnosti m_{ij} , ta se označuje jako **modální** odhad.

Metoda PERT

Metoda **PERT** (Program Evaluation and Review Technique) je pravděpodobnostním rozšířením metody CPM. V praxi často není reálné stanovit realizační doby činností, proto jsou hodnoty y_{ij} nahrazeny náhodnými veličinami, které se realizují na nějakém intervalu $\langle a_{ij}, b_{ij} \rangle$. Kromě tohoto mezního **optimistického**, resp. **pesimistického** odhadu se pracuje s nejpravděpodobnější dobou trvání činnosti m_{ij} , ta se označuje jako **modální** odhad. Skutečné rozložení pravděpodobnosti náhodné veličiny není obecně známo, ale často se aproximuje β -rozdělením. Lze ukázat, že pro její střední hodnotu a rozptyl platí:

$$\mu_{ij} = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}$$

$$\sigma_{ij} = \frac{b_{ij} - a_{ij}}{6}$$

Metoda PERT

Metoda **PERT** (Program Evaluation and Review Technique) je pravděpodobnostním rozšířením metody CPM. V praxi často není reálné stanovit realizační doby činností, proto jsou hodnoty y_{ij} nahrazeny náhodnými veličinami, které se realizují na nějakém intervalu $\langle a_{ij}, b_{ij} \rangle$. Kromě tohoto mezního **optimistického**, resp. **pesimistického** odhadu se pracuje s nejpravděpodobnější dobou trvání činnosti m_{ij} , ta se označuje jako **modální** odhad. Skutečné rozložení pravděpodobnosti náhodné veličiny není obecně známo, ale často se aproximuje β -rozdělením. Lze ukázat, že pro její střední hodnotu a rozptyl platí:

$$\mu_{ij} = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}$$

$$\sigma_{ij} = \frac{b_{ij} - a_{ij}}{6}$$

Vlastní výpočet metodou PERT se neliší od metody CPM, jen se místo hodnot y_{ij} pracuje se středními hodnotami μ_{ij} . Za určitých předpokladů se dá dle centrální limitní věty aproximovat rozložení celkové délky projektu normální náhodnou veličinou a získat tak odpověď na otázky : Jaká je pravděpodobnost, že projekt bude ukončen v čase T , resp. v jakém čase bude projekt ukončen se stanovenou pravděpodobností p , apod.

Metoda PERT - příklad

Příklad : V závodě se má provést rekonstrukce výrobní linky spojená s výměnou výrobního zařízení, stavebními úpravami, generální opravou elektroinstalace a zlepšením pracovního prostředí. Projekt byl rozložen na dílčí činnosti, které jsou spolu s předpokládanou dobou jejich trvání (v týdnech) uvedeny v tabulce. Pomocí metody PERT stanovte kritickou cestu.

Popis činnosti	předchůdci	A	B	C
A Demontáž starého zařízení	-	5	8	10
B Oprava střechy výrobní haly	-	4	6	7
C Oprava podlahy	A	1	2	5
D Vnitřní stavební úpravy	B,C	2	4	6
E Generální oprava elektroinstalace	D	7	10	14
F Montáž nového výrobního zařízení	E	10	12	13
G Montáž klimatizačního zařízení	E	4	5	8
H Zkušební provoz	F	3	4	6
I Dokončení vnitřních stavebních úprav	G	1	3	5

Metoda PERT - příklad

Řešení: Stanovíme střední hodnoty a rozptyly dob trvání činností:

(i, j)	a_{ij}	m_{ij}	b_{ij}	μ_{ij}	σ_{ij}^2
A(1,2)	5	8	10	7,83	0,69
B(1,3)	4	6	7	5,83	0,25
C(2,3)	1	2	5	2,33	0,44
D(3,4)	2	4	6	4,00	0,44
E(4,5)	7	10	14	10,17	1,36
F(5,6)	10	12	13	11,83	0,25
G(5,7)	4	5	8	5,33	0,44
H(6,8)	3	4	6	4,17	0,25
I(7,8)	1	3	5	3,00	0,44

Metoda PERT - příklad

Řešení: Stanovíme střední hodnoty a rozptyly dob trvání činností:

(i, j)	a_{ij}	m_{ij}	b_{ij}	μ_{ij}	σ_{ij}^2
A(1,2)	5	8	10	7,83	0,69
B(1,3)	4	6	7	5,83	0,25
C(2,3)	1	2	5	2,33	0,44
D(3,4)	2	4	6	4,00	0,44
E(4,5)	7	10	14	10,17	1,36
F(5,6)	10	12	13	11,83	0,25
G(5,7)	4	5	8	5,33	0,44
H(6,8)	3	4	6	4,17	0,25
I(7,8)	1	3	5	3,00	0,44

Kritickou cestu lze u jednodušších projektů určit i procházením síťového diagramu všemi způsoby (v našem případě existují čtyři). Přesné časové údaje jednotlivých nahrazujeme středními hodnotami. Pro každou možnou cestu určíme dobu trvání celého projektu, a to součtem středních hodnot jednotlivých činností ležících na dané cestě. Kromě středních hodnot sčítáme i příslušné rozptyly. Ze všech cest vybereme tu s maximální střední hodnotou celkové doby trvání, pro náš projekt je to **A, C, D, E, F, H**.

Metoda PERT - příklad

kritická cesta	a_{ij}	m_{ij}	b_{ij}	μ_{ij}	σ_{ij}^2
A(1,2)	5	8	10	7,83	0,69
C(2,3)	1	2	5	2,33	0,44
D(3,4)	2	4	6	4,00	0,44
E(4,5)	7	10	14	10,17	1,36
F(5,6)	10	12	13	11,83	0,25
H(6,8)	3	4	6	4,17	0,25
Σ				40,33	3,43

Metoda PERT - příklad

kritická cesta	a_{ij}	m_{ij}	b_{ij}	μ_{ij}	σ_{ij}^2
A(1,2)	5	8	10	7,83	0,69
C(2,3)	1	2	5	2,33	0,44
D(3,4)	2	4	6	4,00	0,44
E(4,5)	7	10	14	10,17	1,36
F(5,6)	10	12	13	11,83	0,25
H(6,8)	3	4	6	4,17	0,25
Σ				40,33	3,43

Celková střední doba trvání projektu je $\mu(T) = 40,33$ týdnů a celkový rozptyl činí $\sigma^2(T) = 3,43$ týdnů. Pravděpodobnostní výpočty provádíme **u projektů s velkým počtem činností**, a to za předpokladu, že **zkoumané termíny jsou nezávislé náhodné veličiny**. Zdůvodnitelný je zejména u doby trvání celého projektu. Podle centrální limitní věty platí, že rozdělení náhodné veličiny T , která je součtem velkého počtu nezávislých shodně rozdělených náhodných veličin t_{ij} , se blíží normálnímu rozdělení $N(\mu(T), \sigma^2(T))$.

Výpočet pravděpodobnosti dodržení termínu T_{pl}

Tato pravděpodobnost se určí pomocí hodnot distribuční funkce $\phi(u)$ normovaného normálního rozdělení $N(0; 1)$. Nejprve se musí náhodná veličina T **standardizovat** dle vztahu $U = \frac{T - \mu(T)}{\sigma(T)} \sim N(0; 1)$. Potom

$$P(T \leq T_{pl}) = \Phi\left(\frac{T_{pl} - \mu(T)}{\sigma(T)}\right).$$

Výpočet pravděpodobnosti dodržení termínu T_{pl}

Tato pravděpodobnost se určí pomocí hodnot distribuční funkce $\phi(u)$ normovaného normálního rozdělení $N(0; 1)$. Nejprve se musí náhodná veličina T **standardizovat** dle vztahu $U = \frac{T - \mu(T)}{\sigma(T)} \sim N(0; 1)$. Potom

$$P(T \leq T_{pl}) = \phi\left(\frac{T_{pl} - \mu(T)}{\sigma(T)}\right).$$

- Pokud je plánovaný konec dřívější než střední hodnota doby trvání projektu ($T_{pl} \leq \mu(T)$), argument funkce $\phi(u)$ je záporný. Hodnotu distribuční funkce normovaného normálního rozdělení počítáme podle vztahu $\phi(u) = 1 - \phi(-u)$. Pravděpodobnost dodržení tohoto termínu pak bude menší než 50%.
- Pokud je plánovaný konec shodný se střední hodnotou doby trvání projektu ($T_{pl} = \mu(T)$), pravděpodobnost dodržení termínu bude 50%.
- Pokud je plánovaný konec pozdější než střední hodnota doby trvání projektu ($T_{pl} \geq \mu(T)$), pravděpodobnost dodržení termínu je větší než 50%.

Určení doby trvání projektu T_r při zvolené míře rizika r

Tuto dobu lze stanovit rovněž s využitím tabulek funkce $\phi(u)$. Je-li velikost rizika r v procentech, v tabulce **kvantilů** standardizovaného normálního rozdělení najdeme hodnotu $u_{1-r/100}$. Odpovídající dobu trvání projektu T_r zjistíme ze vztahu $u_{1-r/100} = \frac{T_r - \mu(T)}{\sigma(T)}$, ze kterého vyjádříme dobu trvání projektu $T_r = \mu(T) + \sigma(T) \cdot u_{1-r/100}$

Určení doby trvání projektu T_r při zvolené míře rizika r

Tuto dobu lze stanovit rovněž s využitím tabulek funkce $\phi(u)$. Je-li velikost rizika r v procentech, v tabulce **kvantilů** standardizovaného normálního rozdělení najdeme hodnotu $u_{1-r/100}$. Odpovídající dobu trvání projektu T_r zjistíme ze vztahu $u_{1-r/100} = \frac{T_r - \mu(T)}{\sigma(T)}$, ze kterého vyjádříme dobu trvání projektu $T_r = \mu(T) + \sigma(T) \cdot u_{1-r/100}$

Příklad : Pro výše řešený příklad projektu s celkovou střední dobou trvání 40,33 týdnů a rozptylem 3,43 určete

- 1 s jakou pravděpodobností bude ukončen nejpozději v čase 42 týdnů!

Řešení: $P(T \leq 42) = \phi\left(\frac{42-40,33}{\sqrt{3,43}}\right) = \phi(0,90)$. V tabulce distribuční funkce standardizovaného normálního rozdělení najdeme pravděpodobnost. Projekt bude ukončen nejpozději ve 42 týdnu s pravděpodobností 81,59%.

- 2 dobu realizace projektu, která bude dodržena s rizikem 20%.

Řešení: 20-tiprocentnímu riziku odpovídá 80-tiprocentní pravděpodobnost splnění termínu. Pro tuto pravděpodobnost zjistíme kvantil $u_{0,8} = 0,84$. Po dosažení zjistíme požadovanou dobu $T_r = 40,33 + \sqrt{3,43} \cdot 0,84$. S 80ti procentní pravděpodobností můžeme očekávat, že projekt skončí dříve než v čase 41,88 týdne.

Časově - nákladová analýza projektu

Metoda CPM a PERT přihlíží pouze k časovým vztahům v projektech, přičemž optimální časový rozvrh činností nemusí být vždy hospodárný. Základním kritériem efektivnosti projektu jsou zpravidla **náklady spojené s jeho realizací** a ty úzce souvisejí s dobou trvání:

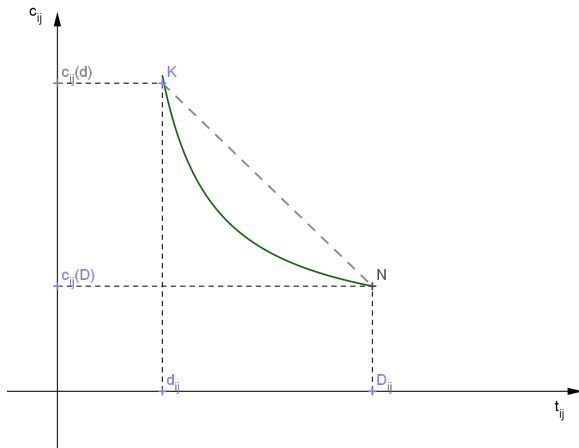
- Náklady nepřímé – souvisí s realizací projektu jako celku (režijní náklady, ztráty vzniklé pozdním dokončením projektu....). Jsou rostoucí (lineární) funkcí doby trvání projektu
- Náklady přímé – souvisejí s jednotlivými činnostmi (materiál, mzdy).
- Součtem přímých nákladů na jednotlivé činnosti získáme přímé náklady na celý projekt.

Přímé náklady na realizaci činnosti (i, j) v čase t_{ij} označíme c_{ij} . Budeme předpokládat opět lineární závislost na době trvání (v tomto případě funkce nerostoucí – se zkrácením doby trvání rostou náklady). Tvar nákladové funkce odvodíme podle těchto pojmů:

D_{ij} ... **normální doba trvání** činnosti (i, j) , které odpovídají minimální náklady $c_{ij}(D)$

d_{ij} ... **krajní doba trvání** činnosti (i, j) při maximálně intenzivním režimu s vysokými náklady $c_{ij}(d)$.

Časově - nákladová analýza projektu: přímé náklady



Přímka KN aproximuje graf závislosti přímých nákladů na době trvání příslušné činnosti. Rovnice této přímky je: $c_{ij} = b_{ij} - a_{ij}t_{ij}$, kde

$$b_{ij} = a_{ij}d_{ij} + c_{ij}(d), \quad a_{ij} = \frac{c_{ij}(d) - c_{ij}(D)}{D_{ij} - d_{ij}}$$

Minimalizace přímých nákladů při dané době trvání projektu

Pro projekt lze úhrnné přímé náklady vyjádřit takto:

$$c_P = \sum_{(i,j) \in P} (b_{ij} - a_{ij}t_{ij}) .$$

Koeficient a_{ij} představuje **nákladový spád** mezi dvojicí bodů odpovídajících normálnímu a maximálně intenzivnímu režimu (v opačném směru jde o nákladový růst).

Minimalizace přímých nákladů při dané době trvání projektu

Pro projekt lze úhrnné přímé náklady vyjádřit takto:

$$c_P = \sum_{(i,j) \in P} (b_{ij} - a_{ij} t_{ij}) .$$

Koeficient a_{ij} představuje **nákladový spád** mezi dvojicí bodů odpovídajících normálnímu a maximálně intenzivnímu režimu (v opačném směru jde o nákladový růst).

Pro přímé náklady c_P spojené s realizací celého projektu v čase T platí:

$$\sum_{(i,j) \in P} c_{ij}(D) \leq c_P \leq \sum_{(i,j) \in P} c_{ij}(d)$$

Při zachování doby trvání projektu T lze tyto náklady snížit **prodloužením doby trvání nekritických činností** až do dosažení jejich normální doby trvání a až do vyčerpání jejich časových rezerv.

Minimalizace přímých nákladů při dané době trvání projektu - příklad

Příklad : Uvažujme projekt zadaný tabulkou (VR_{ij} označuje volné rezervy a červeně jsou označeny kritické činnosti):

(i, j)	t_{ij}	d_{ij}	D_{ij}	$c_{ij}(d)$	$c_{ij}(D)$	a_{ij}	VR_{ij}	c_{ij}
(1,2)	3	3	4	23	20	3	0	23
(1,3)	4	3	5	17	15	1		16
(2,5)	3	2	5	16	10	2	5	14
(3,4)	3	2	4	26	22	2	0	24
(3,5)	7	5	7	38	30	4		30
(4,5)	1	1	1	10	10	-	3	10
Σ				130	107			117

Minimalizace přímých nákladů při dané době trvání projektu - příklad

Příklad : Uvažujme projekt zadaný tabulkou (VR_{ij} označuje volné rezervy a červeně jsou označeny kritické činnosti):

(i, j)	t_{ij}	d_{ij}	D_{ij}	$c_{ij}(d)$	$c_{ij}(D)$	a_{ij}	VR_{ij}	c_{ij}
(1,2)	3	3	4	23	20	3	0	23
(1,3)	4	3	5	17	15	1		16
(2,5)	3	2	5	16	10	2	5	14
(3,4)	3	2	4	26	22	2	0	24
(3,5)	7	5	7	38	30	4		30
(4,5)	1	1	1	10	10	-	3	10
Σ				130	107			117

S realizací daného projektu jsou spojeny přímé náklady ve výši 117 nákladových jednotek (NJ). Tyto náklady lze snížit u nekritické činnosti (2,5) o dvě časové jednotky. Náklady klesnou o 4 NJ, tedy $117 - 4 = 113$.

Prodloužíme dobu trvání nekritických činností maximálně o dobu

$\Delta t_{ij} = \min(VR_{ij}; D_{ij} - t_{ij})$ a přednostně prodloužíme činnosti s velkým a_{ij} .

Stanovení optimální celkové doby trvání projektu

Z hlediska efektivnosti je optimální doba trvání charakterizovaná minimálními celkovými náklady, které jsou součtem přímých a nepřímých nákladů. Jak již bylo uvedeno, přímé náklady se při zkracování doby trvání činnosti zvyšují a nepřímé náklady se snižují. Při optimální době trvání projektu jsou celkové náklady nejnižší.

Stanovení optimální celkové doby trvání projektu

Z hlediska efektivnosti je optimální doba trvání charakterizovaná minimálními celkovými náklady, které jsou součtem přímých a nepřímých nákladů. Jak již bylo uvedeno, přímé náklady se při zkracování doby trvání činnosti zvyšují a nepřímé náklady se snižují. Při optimální době trvání projektu jsou celkové náklady nejnižší. Tato doba se nalezne tak, že se zkracují doby trvání činností ležících na kritické cestě (nejprve u činnosti s nejnižším koeficientem nákladového růstu). Kritické činnosti se zkracují vždy do dosažení krajní doby trvání. Při takovémto zkracování může dojít ke vzniku další kritické cesty, kterou je nutné potom také sledovat.

Stanovení optimální celkové doby trvání projektu

Z hlediska efektivnosti je optimální doba trvání charakterizovaná minimálními celkovými náklady, které jsou součtem přímých a nepřímých nákladů. Jak již bylo uvedeno, přímé náklady se při zkracování doby trvání činnosti zvyšují a nepřímé náklady se snižují. Při optimální době trvání projektu jsou celkové náklady nejnižší. Tato doba se nalezne tak, že se zkracují doby trvání činností ležících na kritické cestě (nejprve u činnosti s nejnižším koeficientem nákladového růstu). Kritické činnosti se zkracují vždy do dosažení krajní doby trvání. Při takovémto zkracování může dojít ke vzniku další kritické cesty, kterou je nutné potom také sledovat.

Výpočet si budeme ilustrovat na výše řešeném příkladu. Předpokládáme, že všechny činnosti mají normální dobu trvání.

T	(i,j)	t_{ij}	Δc_{ij}	NN	PN	CN=PN+NN
12	-	-	-	60	107	167
11	(1,3)	4	1	55	108	163
10	(1,3)	3	1	50	109	159
9	(3,5)	6	4	45	113	158
8	(3,5)	5	4	40	117	157
	(2,5)	4	2			

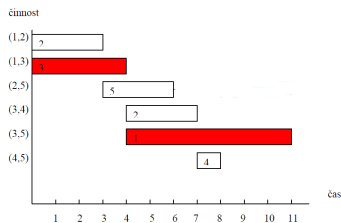
Vznikne další kritická cesta 1-2-5, kterou je nutné brát rovněž v úvahu.

Časově zdrojová analýza projektu

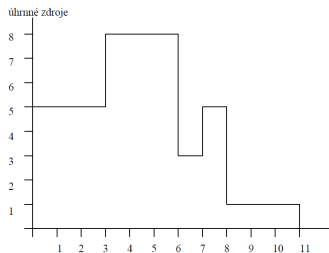
S realizací projektu je vždy spojeno čerpání zdrojů (práce, materiál, finance, atd.). V řízeném projektu je snaha čerpání zdrojů rovnoměrně rozložit na celou dobu trvání projektu. V některých případech při vzniku kapacitních špiček vznikne nedostatek zdroje (jeho potřeba převyšuje jeho disponibilní množství). **Součtová čára** (diagram potřeby zdrojů) – graficky vyjadřuje úhrnné nároky na zdroje v každém okamžiku trvání projektu za předpokladu, že každá činnost začne ve svém nejdříve možném začátku. Součtová čára mění svůj průběh v okamžiku, kdy začíná nebo končí nějaká činnost.

Časově zdrojová analýza projektu - příklad

Pro výše řešenou úlohu máme Ganttův diagram

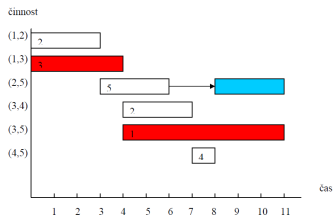


Jeho součtová čára je



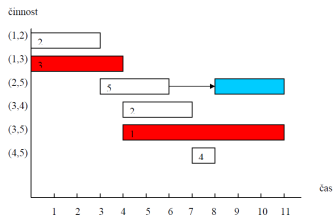
Časově zdrojová analýza projektu - příklad

V časovém intervalu 3,6 lze zvýšený nárok na zdroje odstranit tak, že využijeme volnou časovou rezervu činnosti (2,5) a zahájíme ji až v čase 8 (místo původního začátku v čase 3), který je jejím nejpozději přípustným začátkem.



Časově zdrojová analýza projektu - příklad

V časovém intervalu 3,6 lze zvýšený nárok na zdroje odstranit tak, že využijeme volnou časovou rezervu činnosti (2,5) a zahájíme ji až v čase 8 (místo původního začátku v čase 3), který je jejím nejpozději přípustným začátkem.



Poznámka : V některých případech prodlužujeme dobu trvání nekritických činností, čímž se také sníží potřeba zdroje na tuto činnost v průběhu času.

- Fletcher, Roger: Practical methods of optimization, 1st ed., Chichester : John Wiley and Sons, 1987.
- Šubrt, Tomáš a kol.: Ekonomicko-matematické metody, Plzeň : Vydavatelství a nakladatelství Aleš Čeněk, 2011
- Pánková, Václava: Nelineární optimalizace pro ekonomy, 1. vyd., Praha : Professional Publishing, 2003
- Intriligator, Michael D.: Mathematical optimization and economic theory, Philadelphia : Siam, 2002
- Jablonský, Josef, Petr Fiala, Miroslav Maňas: Vícekriteriální rozhodování, 1. vyd., Praha : Vysoká škola ekonomická v Praze, 1994.
- Klapka, Jindřich, Jiří Dvořák, Pavel Popela Metody operačního výzkumu, Vyd. 2., Brno : VUTIUM, 2001
- Klein, Michael W.: Mathematical methods for economics, 2nd ed., Boston : Addison-Wesley, c2002
- JABLONSKÝ, Josef: Operační výzkum : kvantitativní modely pro ekonomické rozhodování, 1. vyd. Praha: Professional Publishing, 2002
- PLEVNÝ, Miroslav a Miroslav ŽIŽKA: Modelování a optimalizace v manažerském rozhodování, Vyd. 2. Plzeň: Západočeská univerzita, 2010
- GROS, Ivan: Kvantitativní metody v manažerském rozhodování, 1. vyd. Praha: Grada, 2003