

**UNIVERZITA PARDUBICE
ÚSTAV ELEKTROTECHNIKY A INFORMATIKY**

**Vyhodnocení síťového grafu se stochastickou dobou
trvání činností**

BAKALÁŘSKÁ PRÁCE

**AUTOR PRÁCE: Šárka Šlehoferová
VEDOUCÍ PRÁCE: doc. Ing. Josef Volek, Csc.**

2007

**UNIVERSITY OF PARDUBICE
INSTITUTE OF ELECTRICAL ENGINEERING
AND INFORMATICS**

**THE EVALUATION OF THE NETWORK WITH
THE STOCHASTIC DURATION OF PROCESSES**

BACHELOR WORK

**AUTHOR: Šárka Šlehoferová
SUPERVISOR: doc. Ing. Josef Volek, Csc.**

2007

Vysokoškolský ústav: Ústav elektrotechniky a informatiky
Katedra/Ústav: Ústav elektrotechniky a informatiky
Akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Pro: Šárka Šlehoferová

Studijní program: Informační technologie

Studijní obor: Informační technologie

Název tématu: Vyhodnocení síťového grafu se stochastickou dobou trvání činností

Zásady pro zpracování:

Cílem bakalářské práce je vytvoření interaktivního nástroje pro rozhodování v oblasti metod síťové analýzy. Reálnost stanovené doby trvání rozsáhlých projektů ve všech oblastech lidské aktivity je významně ovlivněna přístupem k charakteru doby trvání dílčích činností. Práce bude vycházet z důsledného studia teorie a metod síťové analýzy a z analýzy konkrétní praktické úlohy praxe.

Předpokládaná struktura bakalářské práce:

1. Úvod, motivace
2. Síťová analýza jako nástroj řízení (metody CPM,PERT)
3. Metody konstrukce síťových grafů a metody vyhodnocení
4. Praktický příklad síťového grafu
5. Počítačová implementace metody PERT
6. Vyhodnocení, závěr
7. Literatura

Seznam odborné literatury:

Operační výzkum I: *Volek Josef*

Grafy a jejich použití: *J.Nečas*

Úvod do diskrétní matematiky: *Sergej Vsevolodovic Jablonskij*

Síťová analýza v řídicí praxi: *M.S.Makower. E.Williamson*

Teorie grafů: *Jaroslav Nešetřil*

Grafy a jejich aplikace: *Jiří Demel*

Elementární statistická analýza: *Cyhelský Lubomír, Kahounová Jana, Hindls Richard*

Rozsah: 30-50 stran textu, 5 tabulek, 10 obrázků, CD s programem v jazyku C++

Vedoucí práce: doc. Ing. Josef Volek, CSc.

Vedoucí katedry (ústavu): prof. Ing. Pavel Bezoušek, CSc.

Datum zadání práce: 31. 10. 2006

Termín odevzdání práce: 18. 5. 2007

Prohlašuji:

Tuto práci jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 17.05. 2007

Šárka Šlehoferová

Poděkování

Tímto bych chtěla poděkovat svému vedoucímu bakalářské práce panu Doc., Ing. Josefu Volkovi, Csc., za pomoc a podnětné návrhy při zpracování této bakalářské práce.

ABSTRAKT

Tato práce se zabývá problematikou metody PERT, která se využívá při plánování a řízení procesů. Popisuje aplikaci, která řeší zadanou úlohu, najde kritickou cestu, vypočítá pravděpodobnost záporné rezervy či pravděpodobnost dodržení zadaného termínu. Praktické využití programu je demonstrováno na příkladu rekonstrukce výrobní linky.

OBSAH

1	Úvod a metodika řešení	1
1.1	Historie.....	11
1.2	Základní pojmy síťové analýzy.....	12
1.3	Grafické modely projektů	13
1.4	Metodika řešení.....	15
2	Metody síťové analýzy	16
2.1	Metoda CPM.....	17
2.2	Metoda PERT.....	18
2.2.1	Postup výpočtu metodou PERT	21
3	Aplikace metody PERT na konkrétní úloze	25
3.1	Zadání úlohy	25
3.2	Výpočet zadaného příkladu.....	26
4	Programátorská dokumentace	31
4.1	Použité třídy	32
4.1.1	TVrchol.....	32
4.1.2	THrana	33
4.1.3	TSit.....	34
4.1.4	TMatice	36
4.2	Struktura programu	37
5	Uživatelská dokumentace	38
5.1	Hlavní obsluha programu – Menu Síť	38
5.2	Hlavní menu Náповěda.....	42
5.3	Hlavní menu Program	43
6	Závěr	44

Seznam obrázků

Obrázek 1 Multigraf, Zdroj [10]	13
Obrázek 2 Síť	13
Obrázek 3 Nesprávný a správný způsob konstrukce síťového grafu, Zdroj [2]	14
Obrázek 4 Ohodnocení vrcholů síťového grafu, Zdroj [2]	17
Obrázek 5 Normální rozdělení pravděpodobnosti	19
Obrázek 6 Ohodnocení vrcholů síťového grafu, Zdroj [2]	22
Obrázek 7 Graf návazností	26
Obrázek 8 Síť po provedení kroku 3	28
Obrázek 9 Síť po provedení kroku 5	29
Obrázek 10 UML diagram třídy TVrchol	33
Obrázek 11 UML diagram třídy THrana	34
Obrázek 12 UML diagram třídy TSit	36
Obrázek 13 Hierarchie tříd	37
Obrázek 14 Hlavní okno programu	38
Obrázek 15 Menu Síť	38
Obrázek 16 Okno pro načtení jednotlivých hodnot	39
Obrázek 17 Zobrazení dat ve formě nakreslené sítě	39
Obrázek 18 Výpis načtených dat ve formě tabulky	40
Obrázek 19 Výpis vypočtených dat ve formě tabulky	41
Obrázek 20 Pravděpodobnost vzniku záporné rezervy	41
Obrázek 21 Okno pro výpočet pravděpodobnosti dodržení či překročení termínu	42
Obrázek 22 Hlavní menu Náповěda	42
Obrázek 23 Hlavní menu Program	43
Obrázek 24 Dialog pro uložení do souboru	43
Obrázek 25 Kritická cesta	Error! Bookmark not defined.

1 Úvod a metodika řešení

Síťová analýza se věnuje modelování složitých komplexů událostí. Jde o metody modelování souboru navazujících činností nutných k dosažení určitého cíle. K tomu je využíván především aparát teorie grafů doplněný poznatky teorie pravděpodobnosti a matematické statistiky, lineárního programování a dalších matematických metod.

Cílem síťové analýzy je najít efektivní časovou, technologickou a organizační koordinaci dílčích, vzájemně na sebe navazujících úkolů v rámci složitého komplexu činností. Síťová analýza nachází své uplatnění především při řízení výstavby velkých investičních celků, při koordinaci složitých výzkumných prací, při vývoji nových výrobků, při rekonstrukci zařízení apod. Síťová analýza v současné době nabízí desítky metod s velkým počtem jejich modifikací. Metody díky své poměrné jednoduchosti a efektivnosti doznaly značného rozšíření v praxi.

1.1 Historie

Historie síťové analýzy je pevně spjata s historií jejích dvou základních metod: metody CPM a metody PERT.

Metoda kritické cesty (CPM, Critical Path Method) byla vyvinuta pro společnost Du Pont de Nemours a to konkrétně pro efektivnější řízení a plánování výstavby chemických provozů firmy. Nová metoda byla poprvé využita při výstavbě nového chemického provozu za \$10 mil. souběžně s dosavadní metodou plánování. Tehdy se nová metoda ukázala jako méně pracná, pružnější a navíc umožnila zkrácení výstavby při minimálních dodatečných nákladech. Konkrétně byla výstavba zkrácena o 2 měsíce s možností zkrácení o další 2 měsíce při zvýšení nákladů pouze o 1%. Tyto skutečnosti vedly k rozšíření metody CPM i do ostatních odvětví.

Teorie a technika metody PERT byla vypracována v roce 1958 za účelem zefektivnění řešení otázek plánování, řízení a kontroly projektu Polaris. Stávající plán zajišťoval splnění programu ve stanoveném krajním termínu. To ve svém důsledku znamenalo, že jakýkoliv skluz dílčích úkolů by se promítl do termínu ukončení celého projektu. Proto byla vypracována nová metodika

plánování (PERT, Program Evaluation Research Task), která umožnila zjistit potřebné informace o celém komplexu činností, jeho hlavních člancích a důsledcích vlivu nastalých změn a uvažovaných opatření. Využití této metodiky plánování kromě uvedených přínosů umožnilo ukončit projekt o tři roky dříve před původně plánovaným termínem. Tento úspěch otevřel cestu metodě PERT i do hospodářských oblastí.

V následujícím období se objevila celá řada metod založených na principu metod PERT a CPM motivovaných konkrétními potřebami řešených problémů. Tak vznikly například metody RAMPS, SUPERT,

PERTCOM apod.

V současnosti existuje množství počítačových prostředků specializujících se na problematiku síťové analýzy zaměřených na konkrétní obor. Navíc většina obecných plánovacích systémů nabízí některou z metod CPM nebo PERT.

Zdroj: [9]

1.2 Základní pojmy síťové analýzy

Graf představuje grafické znázornění modelovaného projektu, tj. zobrazuje průběh akcí prováděných v rámci daného projektu.

Uzel grafu představuje časovou událost, kterou je začátek nebo konec činnosti.

Hrana grafu představuje činnost, která klade nároky na čas, zdroje a má dynamický charakter.

Konečný graf má konečný počet uzlů a hran.

Orientovaný graf je tvořen orientovanými hranami, kterým je přiřazen určitý směr.

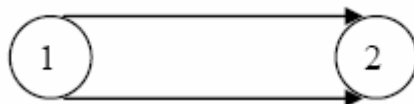
Hranově (uzlově) ohodnocený graf je graf, jehož každé hraně (uzlu) je přiřazeno alespoň jedno číslo (mapa trasy dálkového pochodu, každé spojnici mezi jednotlivými stanovišti je přiřazena její délka).

Cesta je posloupnost hran v orientovaném grafu, ve kterém každá hrana vychází z uzlu, v němž končí předcházející. Pokud cesta začíná a končí ve stejném uzlu, potom se jedná o **cyklus**.

Acyklický graf neobsahuje žádný cyklus.

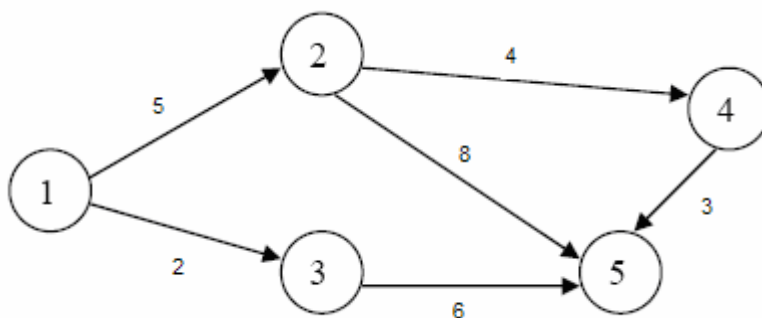
Souvislý graf je takový neorientovaný graf, pro který platí, že pro všechny dvojice jeho uzlů existuje alespoň jedna cesta, která je spojuje.

Multigraf je graf, ve kterém mezi některou dvojicí uzlů existuje více souhlasně orientovaných hran.



Obrázek 1 Multigraf, Zdroj [10]

Síť je konečný, souvislý, orientovaný, acyklický, hranově nebo uzlově ohodnocený graf, v němž existuje jeden počáteční uzel (nevstupuje do něj žádná hrana) a jeden uzel koncový (žádná hrana z něj nevystupuje).



Obrázek 2 Síť

Síťový diagram je síťový graf, jehož hrany jsou ohodnoceny časovými údaji.

Délka cesty v síťovém diagramu představuje součet časových údajů přiřazených hranám, které tvoří uvažovanou cestu.

Kritická cesta jsou vrcholy a hrany sítě, jejichž prodloužením činností dojde k prodloužení celé délky konání projektu.

1.3 Grafické modely projektů

Projekty lze znázornit síťovým diagramem. Hrany představují jednotlivé činnosti a uzly představují začátky a konce jednotlivých činností.

Podmínky pro modelování a řízení projektu síťovým diagramem:

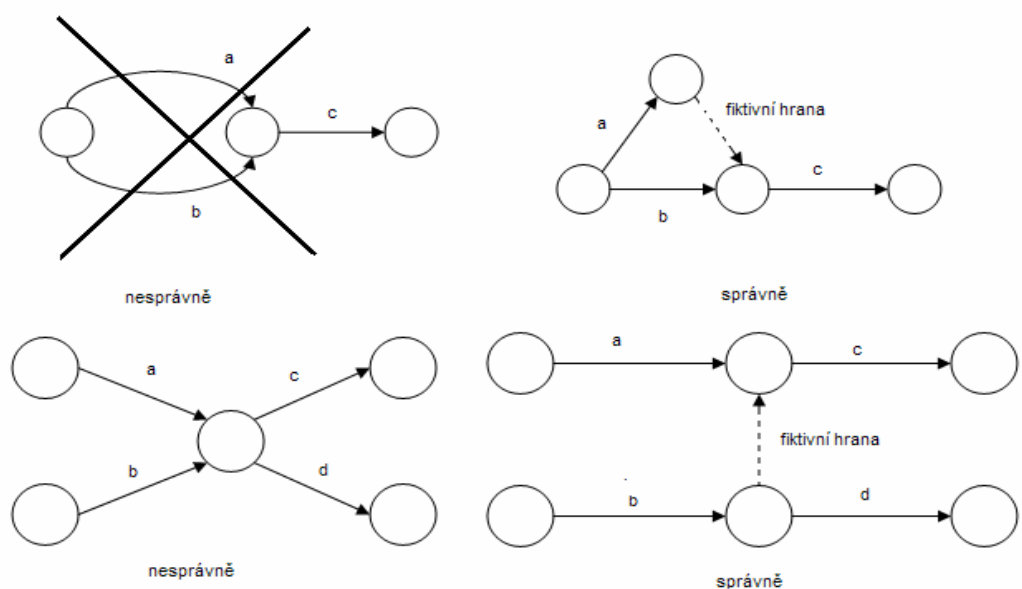
- 1) pro každou činnost je známá doba trvání
- 2) pro každou činnost je definována činnost předcházející a činnost následující
- 3) pokud je přihlíženo i k jiným kritériím optimality než doba trvání, každá činnost musí být ohodnocena příslušnými ukazateli (například kapacitní omezení)
- 4) cíl projektu je splněn, pokud jsou ve správném časovém sledu provedeny všechny činnosti, při splnění dalších podmínek například kapacitní omezení.

Síťový graf musí být zakreslen co nejpřehledněji. Délka hran nemusí odpovídat době trvání na rozdíl od harmonogramu.

Při sestavování grafu lze začít od počátečního uzlu (zvláště u známých projektů) nebo od konečného uzlu (především u doposud nerealizovaných projektů) nebo lze kombinovat oba způsoby.

Uzly jsou číslovány přirozenými čísly, počáteční uzel má nižší číslo než koncový. Hran mají buď kladné ohodnocení (u skutečných činnostech) nebo nulové ohodnocení (u fiktivních činnostech). Fiktivní činnost slouží k vyjádření návaznosti skutečných činností nebo k zamezení vzniku multigrafu.

Zdroj: [10]



Obrázek 3 Nesprávný a správný způsob konstrukce síťového grafu, Zdroj [2]

1.4 Metodika řešení

Na základě zadaného problému při rekonstrukci výrobní linky byl vytvořen daný program. Pro vypracování tohoto projektu bylo použito plánovací metody síťové analýzy (metody PERT). Nejdříve bylo potřeba na základě zadaných dat o času trvání jednotlivých úkolů potřebných pro rekonstrukci linky, určit na sebe navazující činnosti, z kterých je následně možno vytvořit hranově orientovanou síť. Zjištěné údaje o navazujících činnostech již lze zadat do programu, který je vyhodnotí a ve vytvořené síti provede veškeré důležité výpočty. Cílem programu je odhadnout délku projektu, s jakou pravděpodobností bude překročen termín či rozpočet projektu a nalézt kritickou cestu a kritické uzly sítě.

2 Metody síťové analýzy

V praxi se často setkáváme se složitými systémy lidí, strojů a materiálů, které je nutné optimálně řídit k dosažení předem stanovených cílů. Činnosti těchto komplexních systémů zpravidla dají rozdělit do dílčích činností, které na sebe navzájem určitým způsobem navazují. Příkladem takového systému může být technologický proces zpracování průběžných nákladních vlaků v seřaďovací stanici, stavba říční lodě v loděnici, ale i projekt pilotovaného letu na Mars nebo vývoj vakcíny na léčení infekčních nemocí. Každá činnost vyžaduje určitý čas k realizaci, spotřebovává zdroje (suroviny, materiály) a vyžaduje i lidskou aktivitu. Pro efektivní řízení těchto systémů je potřebné:

- stanovení objektivně nutného času k realizaci projektu,
- stanovení dílčích časů
- používat efektivních prostředků pro kontrolu postupu prací,
- používat racionální a objektivní nástroje na odstraňování neočekávaných poruch postupu prací.

Zdroj: [2]

Metody analýzy kritické cesty jsou jednou z nejrozšířenějších oblastí aplikace teorie sítí v praxi. Jedná se o souhrnné označení nástrojů modelování zkoumání relativně složitých systémů (procesů), používaných především při plánování, řízení, koordinaci a kontrole rozsáhlých projektů.

Projektem se rozumí soubor prostorově a časově vymezených činností. Celou množinu činností a jejich technologických a organizačních vztahů v určitém systému lze znázornit hranově nebo uzlově ohodnocenou sítí.

Síťová analýza je disciplína teorie grafů, která je zaměřena na analýzu projektů. Modelem projektu je síťový graf (speciální typ orientovaného grafu). Cílem síťové analýzy je nalezení kritické cesty a kritických činností popř. zjištění dalších údajů (rezervy činností apod.).

První, nejznámější a nejrozšířenější modely a metody síťové analýzy, používané při plánování a řízení složitých projektů, jsou metody CPM a PERT.

Charakteristickým rysem těchto metod je zobrazení projektu ve tvaru grafu, přesněji řečeno v souvislosti s body zahájení a ukončení realizace

projektu ve tvaru sítě a berou v úvahu pouze faktor času, takže slouží především k časové analýze kritické cesty. Síť zobrazuje příčinné vztahy mezi jednotlivými činnostmi - tedy v podstatě topologickou návaznost jednotlivých prací na sebe. Například nejprve musí být proveden výkop, potom betonáž základů a teprve pak je možné postavit zeď z cihel, ne naopak.

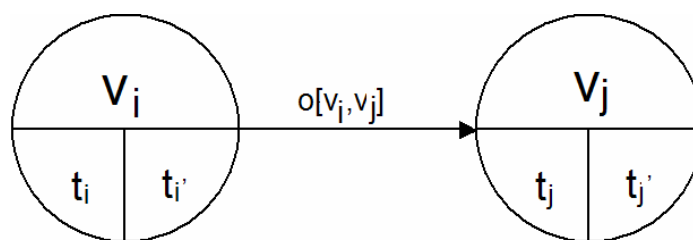
2.1 Metoda CPM

Metoda CPM patří mezi základní deterministické metody síťové analýzy, jejichž cílem je stanovení doby trvání projektu na základě tzv. kritické cesty. CPM umožňuje usnadnit efektivní časovou koordinaci dílčích, vzájemně na sebe navazujících činností v rámci projektu. Předpokládá se, že pro každou činnost známe dobu trvání, kterou je možné např. vypočítat z rozsahu práce a známé normy.

Metoda CPM předpokládá postupnou realizaci těchto kroků:

- Sestava modelu ve formě síťového grafu včetně očíslování uzlů.
- Určení doby trvání činností a propočet dílčích termínů uzlů a činností.
- Nalezení kritické cesty a její analýza.
- Výpočet časových rezerv uzlů a činností

Každý vrchol síťového grafu je ohodnocen dvěma časovými údaji:



Obrázek 4 Ohodnocení vrcholů síťového grafu, Zdroj [2]

$o[v_i, v_j]$ - doba trvání činnosti $[v_i, v_j]$

t_i - termín nejdříve možného zahájení činností, vycházejících z uzlu v_i

t_j - termín nejdříve možného ukončení činností, končících v uzlu v_j

t_i' - termín nejpozději možného zahájení činností, vycházejících z uzlu v_i

t_j' - termín nejpozději přípustného ukončení činností, končících v uzlu v_j

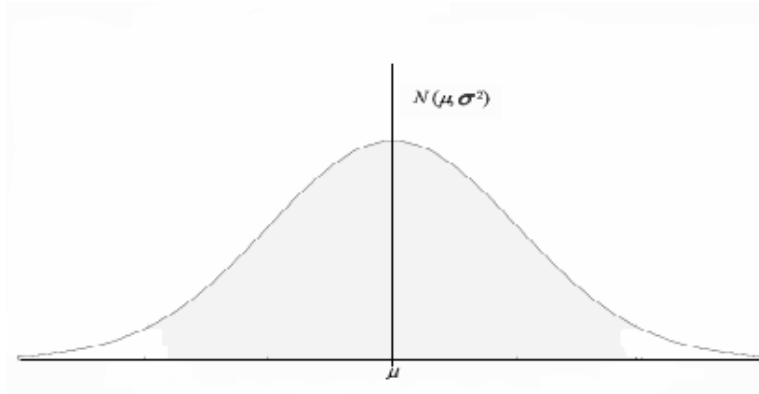
$t_i + o[v_i, v_j]$ - termín nejdříve možného konce činnosti $[v_i, v_j]$

$t_j - o[v_i, v_j]$ - termín nejpozději možného konce činnosti $[v_i, v_j]$

2.2 Metoda PERT

V metodě CPM se předpokládá jednoznačná znalost doby trvání činnosti t_{ij} (t_{ij} považujeme za časovou konstantu). Úlohy řešené metodami síťové analýzy jsou ale zpravidla unikátní, tj. neopakují se a pracuje se tedy často s odhadovanými údaji, zatíženými určitou chybou. Přestává-li deterministicky ohodnocená síť být adekvátním modelem skutečnosti, je třeba při časové analýze projektu vyjít ze stochasticky ohodnoceného síťového grafu, který je základem metody PERT.

Cílem modelů PERT (“Program Evaluation and Review Technique”) je takové uspořádání činností, které by zajistilo dodržení termínu dokončení projektu s dostatečně vysokou pravděpodobností. Základní odlišností od metody CPM je, že doba trvání činnosti není přesně známa, nýbrž je dána odhadem. Tato doba trvání není konstantou, ale náhodnou veličinou s určitým rozdělením pravděpodobnosti. Vzhledem k charakteru problémů řešených v rámci projektového řízení, bylo pro klasické postupy zvoleno rozdělení pravděpodobnosti beta. Toto rozdělení je velmi blízké rozdělení normálnímu, je spojitě, jednovrcholové, mírně asymetrické, ale na rozdíl od normálního je oboustranně ohraničené. Užitečnou vlastností beta rozdělení, kromě unimodálnosti a konečného variačního rozpětí, je že v závislosti na hodnotách parametrů může být symetrické stejně jako různým způsobem pozitivně nebo negativně sešikmené, takže umožňuje dostatečně pružně a přesně modelovat rozdělení hustoty pravděpodobnosti trvání jednotlivých činností.



Obrázek 5 Normální rozdělení pravděpodobnosti

Předpokladem výpočtu modelu PERT je kvalifikovaný odhad délek trvání jednotlivých činností a to ve formě tří časů:

a_{ij} je optimistický odhad doby trvání činnosti $[v_i, v_j]$. Činnost nemůže v žádném případě trvat kratší dobu, než je a_{ij} . Odpovídá nejkratšímu předpokládanému trvání činnosti za extrémně příznivých podmínek

b_{ij} je pesimistický odhad doby trvání činnosti $[v_i, v_j]$. Činnost nemůže v žádném případě trvat delší dobu, než je b_{ij} . Představuje nejdelší předpokládané trvání činnosti, přihlíží k možnosti, že realizace činnosti bude vzhledem k nepříznivým podmínkám obtížná

m_{ij} je nejpravděpodobnější (normální) odhad doby trvání činnosti $[v_i, v_j]$. Vychází z normálních podmínek při její realizaci.

Vyjde-li se z předpokladu beta rozdělení dob trvání činnosti, lze ze tří druhů odhadů pro každou činnost stanovit základní charakteristiky rozdělení, tj. střední hodnotu, směrodatnou odchylku a rozptyl, které označíme $t_e[h]$, δ_{ij} a δ_{ij}^2 .

Čím vyšší je hodnota rozptylu, tím větší je pravděpodobnost, že skutečná hodnota doby trvání činnosti se bude více lišit od její střední hodnoty.

Cílem výpočtů PERT je výpočet středních hodnot a rozptylů všech termínů nejdříve a nejpozději možných pro všechny činnosti a uzly a určení tzv. očekávané kritické cesty. Dále na základě pravděpodobnostní analýzy vypočtených parametrů posoudit pravděpodobnost vzniku časové rezervy uzlu, pravděpodobnost konkrétní kritické cesty, pravděpodobnost dodržení plánovaného termínu dokončení projektu atd.

Stanovíme-li pro všechny činnosti projektu odhady středních hodnot dob jejich trvání a použijeme je k ohodnocení činností v síťovém grafu, můžeme přistoupit k výpočtu kritické cesty a časových rezerv činností shodným způsobem jako v případě CPM.

Střední hodnota celého projektu je u metody PERT dána obdobně jako u CPM součtem středních hodnot trvání kritických činností, které jsou ovšem stochastickými veličinami. Proto i střední hodnota trvání projektu \bar{T} má pravděpodobnostní charakter. Stejně tak směrodatná odchylka i rozptyl doby trvání projektu δ popř. δ^2 jsou rovny součtu směrodatných odchylek popř. rozptylů trvání jednotlivých činností na kritické cestě, jsou-li doby trvání činností vzájemně nezávislé a mají-li shodná beta rozdělení. Vypočtené hodnoty jednotlivých druhů časových rezerv představují také jejich střední hodnoty.

V důsledku náhodného charakteru dob trvání činností je při časové analýze sítě pomocí PERT účelné stanovit jednak pravděpodobnost ukončení projektu za dobu \bar{T} , ale i pravděpodobnost, že např. celý projekt skončí za předem určenou dobou, která se liší od spočtené střední hodnoty \bar{T} , popř. určit střední hodnotu doby realizace projektu, připouští-li se s určitou pravděpodobností možnost, že projekt neskončí za dobu \bar{T} , ale později.

Tyto pravděpodobnostní charakteristiky je možno snadno vypočítat, dá-li se rozdělení doby trvání projektu jako náhodné veličiny aproximativně popsat např. normálním rozdělením. Tato podmínka je zpravidla splněna tehdy, jestliže počet činností na kritické cestě je dostatečně velký (alespoň 30), neboť potom na základě centrální limitní věty platí, že rozdělení náhodné veličiny T , která je součtem nezávislých shodně rozdělených náhodných veličin y_{ij} na kritické cestě, se blíží normálnímu rozdělení $N[\bar{T}, \delta^2_T]$. Hledané pravděpodobnosti získáme v takovém případě jako hodnoty distribuční funkce normálního rozdělení.

Pro náhodnou veličinu t normovanou ve tvaru

$$t = \frac{T - \bar{T}}{\delta_T},$$

kteřá má normální rozdělení s nulovou střední hodnotou a

s jednotkovým rozptylem, nalezneme odpovídající pravděpodobnosti v tabulce hodnot distribuční funkce normovaného normálního rozdělení.

Pravděpodobnost $p[T \leq \bar{T}]$, že skutečná doba realizace projektu nepřekročí spočtenou střední hodnotu trvání projektu \bar{T} , je rovna 0,5. Obdobným postupem lze dospět k vyčíslení pravděpodobnosti dodržení termínů ukončení i některých rozhodujících etap projektu - tzv. milníků nebo dokonce jednotlivých činností.

2.2.1 Postup výpočtu metodou PERT

Pro každou hranu $h[v_i, v_j] \in X$, kde X je množina všech hran v síti a vrchol $v_i \in V$, kde V je množina všech vrcholů síťového grafu postupně určíme:

Očekávaný čas trvání činnosti $t_e[h]$ (střední hodnota) - převádí tři časové odhady dob trvání činnosti v jediný odhad střední hodnoty doby trvání činnosti. Pro potřeby určení času t_e byl sestaven vzorec:

$$t_e = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6} \quad [3.1]$$

Rozptyl doby trvání činnosti $\delta^2[h]$ - pro čas t_e byla hodnota rozptylu stanovena s využitím pravidla 3σ pro normální rozdělení pravděpodobnosti na:

$$\delta_{t_e}^2 = \left(\frac{b_{ij} - a_{ij}}{6} \right)^2 \quad [3.2]$$

Nejdříve možný konec činností vcházejících do vrcholu $T_E^{v_i}$ - resp. nejdříve možný začátek činností vycházejících z vrcholu je dán součtem časů t_e , které tvoří maximální dráhu z vrcholu v_o do vrcholu v_i . Žádnou činnost nelze zahájit, dokud nejsou ukončeny všechny činnosti, které ji bezprostředně předcházejí. Čas t_i chápeme jako střední hodnotu proměnné s normálním rozdělením pravděpodobnosti.

$$T_E^{v_i} = \sum_{h \in m[v_o, v_i]} t_e[h] \quad [3.3]$$

$m[v_o, v_i]$ je maximální orientovaná cesta z v_o do v_i

Směrodatnou odchylku času $T_E^{v_i}$ – určí se analogicky podle definice času $T_E^{v_i}$

$$\delta_{T_E^{v_i}} = \sqrt{\sum_{h \in m[v_o, v_i]} \delta_{t_e}^2[h]} \quad [3.4]$$

Nejpozději nutný konec činností vcházejících do vrcholu $T_L^{v_i}$ – resp. nejpozději nutný začátek činností vycházejících z vrcholu je dán součtem časů t_e , které tvoří maximální dráhu z vrcholu v_i do vrcholu v_n . Aby nedošlo ke skluzu v době trvání projetu musí všechny činnosti vcházející do daného vrcholu být ukončeny v jistém časovém okamžiku, tak aby bylo možné realizovat následující činnosti v požadovaných termínech. Čas chápeme jako střední hodnotu proměnné s normálním rozdělením pravděpodobnosti.

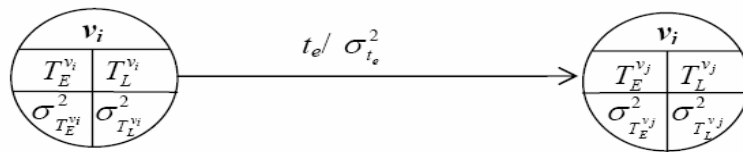
$$T_L^{v_i} = T_L^{v_n} - \sum_{h \in m[v_i, v_n]} t_e[h] \quad [3.5]$$

$m[v_i, v_n]$ je maximální orientovaná cesta z v_o do v_n

Směrodatnou odchylku času $T_L^{v_i}$ – určí se analogicky podle definice času $T_L^{v_i}$

$$\delta_{T_L^{v_i}} = \sqrt{\sum_{h \in m[v_i, v_n]} \delta_{t_e}^2[h]} \quad [3.6]$$

Tyto údaje se zapisují k příslušným hranám a vrcholům grafu následovně



Obrázek 6 Ohodnocení vrcholů síťového grafu, Zdroj [2]

Pravděpodobnost vzniku záporné rezervy – pravděpodobnost, že rezerva bude záporná $P\{R \leq 0\}$ určíme podle následujícího vztahu:

$$P\{R \leq 0\} = \phi(u) = \phi\left(\frac{T_E - T_L}{\sqrt{\delta_{T_E}^2 + \delta_{T_L}^2}}\right) = \phi\left(\frac{-R}{\delta_{T_R}}\right) \quad [3.7]$$

Pravděpodobnost dodržení plánovaného termínu – pravděpodobnost dodržení plánovaného termínu $T_S^{v_i}$, tedy $P\{ T_E^{v_i} \leq T_S^{v_i} \}$ určíme podle následujícího vztahu:

$$P\{ T_E^{v_i} \leq T_S^{v_i} \} = \phi(u) = \phi\left(\frac{T_S^{v_i} - T_E^{v_i}}{\delta_{T_E^{v_i}}}\right) \quad [3.8]$$

Pravděpodobnost překročení plánovaného termínu – pravděpodobnost překročení plánovaného termínu $T_S^{v_i}$, tedy $P\{ T_E^{v_i} > T_S^{v_i} \}$ určíme podle následujícího vztahu:

$$P\{ T_E^{v_i} > T_S^{v_i} \} = \phi(u) = \phi\left(\frac{T_E^{v_i} - T_S^{v_i}}{\delta_{T_E^{v_i}}}\right) \quad [3.9]$$

Jestliže $T_S^{v_i} > T_E^{v_i}$ je pravděpodobnost dodržení termínu $p > 0.5$.

Jestliže $T_S^{v_i} = T_E^{v_i}$ je 50 % naděje na dodržení termínu.

Jestliže $T_S^{v_i} < T_E^{v_i}$ je pravděpodobnost dodržení termínu $p < 0.5$

Zdroj: [2]

2.2.1.1 Algoritmus výpočtu

Krok 1. Podle příslušných vztahů určíme pro každou hranu $h_{ij} \in X$ hodnotu očekávané doby trvání činnosti a jejího rozptylu.

Krok 2. V počátečním vrcholu v_o položíme čas nejdříve možného začátku projektu roven nule $T_E^{v_o} = 0$, rozptyl času $T_E^{v_o} = 0$ je také roven nule $\delta_{T_E^{v_o}}^2 = 0$.

Krok 3. V síťovém grafu určíme pro všechny vrcholy $v_i \in V$, pro $i = 1, 2, \dots, n$ čas nejdříve možného začátku dílčích činností vycházejících z daného vrcholu - $T_E^{v_i}$ a příslušný rozptyl těchto hodnot - $\delta_{T_E^{v_i}}^2$. Čas $T_E^{v_i}$ je dán maximální hodnotou z nejdříve možných konců činností vcházejících do vrcholu v_i tzn. délkou maximální dráhy mezi počátečním vrcholem v_o a vrcholem v_i . Rozptyl $\delta_{T_E^{v_i}}^2$ času $T_E^{v_i}$ určíme jako součet rozptylů $\delta_{t_e}^2$ na maximální dráze mezi počátečním vrcholem v_o a vrcholem v_i , která je dána

součtem dob trvání dílčích činností $t_e[h]$, $h \in m[v_0, v_i]$.

Krok 4. Čas nejpozději nutného ukončení projektu $T_L^{v_n}$ je roven času nejdříve možného končení projektu $T_L^{v_n} = T_E^{v_n}$, rozptyl času $T_L^{v_n}$ je roven nule $\delta_{T_L^{v_n}}^2 = 0$.

Krok 5. Určíme časy nejpozději nutného ukončení dílčích činností $T_L^{v_i}$ a jejich rozptyly $\delta_{T_L^{v_i}}^2$ pro $v_i \in V$, kde $i = 0, 1, \dots, n - 1$. Čas nejpozději nutného ukončení činnosti vcházejících do vrcholu je dán rozdílem času ukončení projektu $T_L^{v_n}$ a maximální hodnotou součtu dob trvání dílčích činností mezi vrcholem v_i a vrcholem v_n .

Krok 6. Z výsledného síťového grafu určíme vrcholy a hrany, které leží na kritické cestě. Pro vrcholy kritické cesty platí, že $T_E^{v_i} = T_L^{v_i}$. Pro hrany ležící na kritické cestě platí, že časový interval mezi okamžikem nejdříve možného začátku této činnosti a okamžikem jejího nejpozději nutného konce je roven době trvání této činnosti, tzn.

$$T_L^{v_j} - T_E^{v_i} = t_e^{ij}.$$

Krok 7: Určíme pravděpodobnosti vzniku záporné rezervy ve vrcholech zadaného grafu $v_i \in V$, kde $i = 0, 1, \dots, n$, pravděpodobnost dodržení a překročení požadovaného termínu T_s .

Zdroj: [2]

3 Aplikace metody PERT na konkrétní úloze

3.1 Zadání úlohy

Přiložený program je univerzální pro jakýkoliv problém, ale univerzální grafické zobrazení sítě je algoritmicky velmi náročné, proto provede výpis pouze ve formě tabulky. Pro následující úlohu je program optimalizován, tzn., že kromě tabulkového výpisu dojde i ke grafickému zobrazení. Následující kapitola se zabývá tímto konkrétním zadáním.

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.

Činnost	Popis činnosti	Doba trvání		
		a	m	B
1	Demontáž starého zařízení	5	8	10
2	Oprava střechy výrobní haly	4	6	7
3	Oprava podlahy	1	2	5
4	Vnitřní stavební úpravy	2	4	6
5	Generální oprava elektroinstalace	7	10	14
6	Montáž nového výrobního zařízení	10	12	13
7	Montáž klimatizačního zařízení	4	5	8
8	Zkušební provoz	3	4	6
9	Dokončení vnitřních stavebních úprav	1	3	5

Rozborem souvislostí mezi dílčími činnostmi bylo zjištěno, že demontáž starého zařízení a oprava střechy mohou probíhat současně. Vnitřní stavební úpravy lze provádět po skončení opravy střechy a podlahy, přičemž opravu podlahy lze provést až po demontáži zařízení. Generální oprava elektroinstalace může být provedena po dokončení vnitřních stavebních úprav. Montáž nového výrobního a klimatizačního zařízení lze provádět současně, ale musí být ukončena generální oprava elektroinstalace. Zkušební provoz může být zahájen po skončení montáže výrobního zařízení a dokončovací úpravy mohou probíhat nezávisle na zkušebním provozu, jakmile byla provedena montáž klimatizačního zařízení.

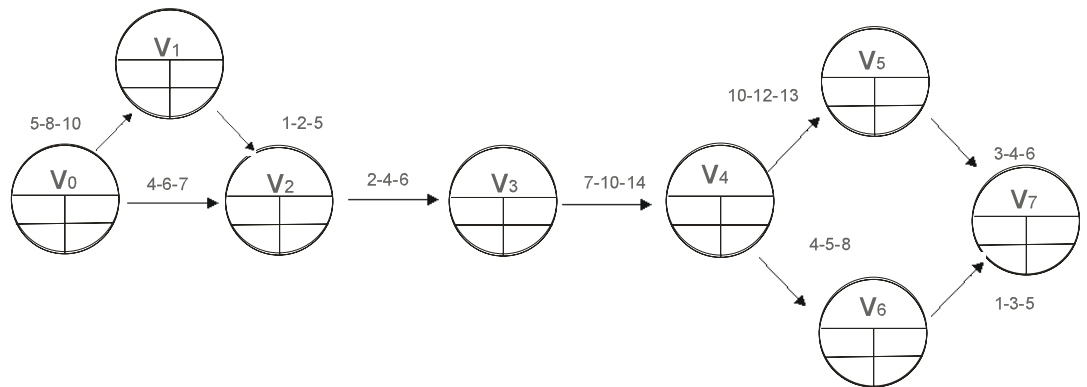
Zdroj:[10]

3.2 Výpočet zadaného příkladu

V tabulce 1 a obrázku 7 jsou uvedeny všechny činnosti rekonstrukce výrobní linky. Konstrukce grafu respektuje návaznosti uvedené v kapitole 4.1.

činnost	Optimist.čas	Nejpravděp.čas	Pesimist.čas
t ₀₁	5	8	10
t ₀₂	4	6	7
t ₁₂	1	2	5
t ₂₃	2	4	6
t ₃₄	7	10	14
t ₄₅	10	12	13
t ₄₆	4	5	8
t ₅₇	3	4	6
t ₆₇	1	3	5

Tabulka 1 Tabulka návazností



Obrázek 7 Graf návazností

Krok 1. Podle vzorce [3.1] určíme pro každou hranu $h_{ij} \in X$ hodnotu očekávané doby trvání činnosti a jejího rozptylu:

$$t_e[v_0, v_1] = \frac{5 + 4.8 + 10}{6} = 7,83$$

$$\delta_{t_e}^2[v_0, v_1] = \left(\frac{10 - 5}{6}\right)^2 = 0,69$$

$$t_e[v_0, v_2] = \frac{4 + 4.6 + 7}{6} = 5,83$$

$$\delta_{t_e}^2[v_0, v_2] = \left(\frac{7 - 4}{6}\right)^2 = 0,25$$

$$t_e[v_1, v_2] = \frac{1+4.2+5}{6} = 2,33$$

$$\delta_{t_e}^2[v_1, v_2] = \left(\frac{5-1}{6}\right)^2 = 0,44$$

$$t_e[v_2, v_3] = \frac{2+4.4+6}{6} = 4$$

$$\delta_{t_e}^2[v_2, v_3] = \left(\frac{6-2}{6}\right)^2 = 0,44$$

$$t_e[v_3, v_4] = \frac{7+4.10+14}{6} = 10,17$$

$$\delta_{t_e}^2[v_3, v_4] = \left(\frac{14-7}{6}\right)^2 = 1,36$$

$$t_e[v_4, v_5] = \frac{10+4.12+13}{6} = 11,83$$

$$\delta_{t_e}^2[v_4, v_5] = \left(\frac{13-10}{6}\right)^2 = 0,25$$

$$t_e[v_4, v_6] = \frac{4+4.5+8}{6} = 5,33$$

$$\delta_{t_e}^2[v_4, v_6] = \left(\frac{8-4}{6}\right)^2 = 0,44$$

$$t_e[v_5, v_7] = \frac{3+4.4+6}{6} = 4,17$$

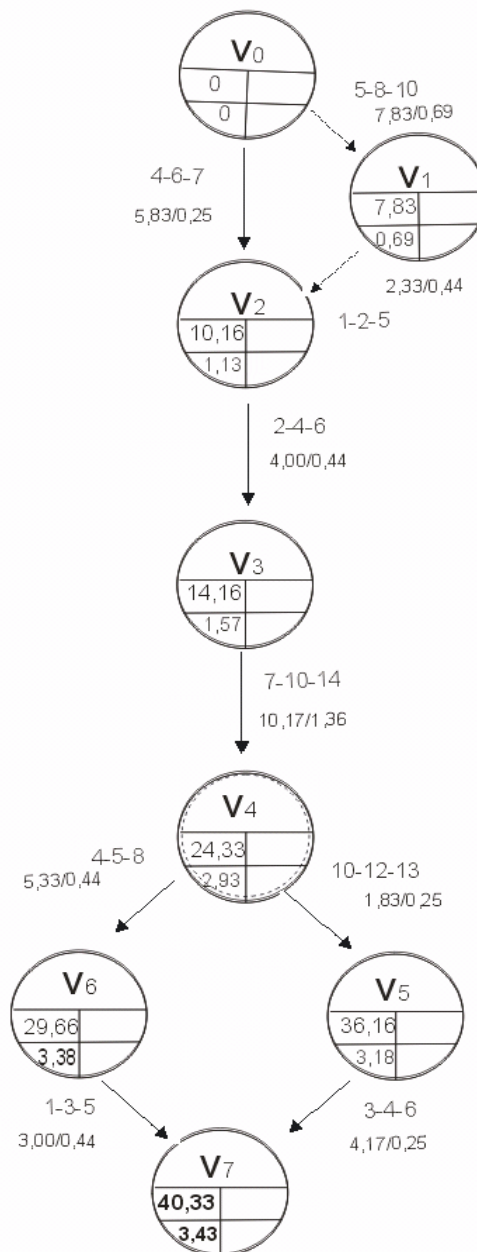
$$\delta_{t_e}^2[v_5, v_7] = \left(\frac{6-3}{6}\right)^2 = 0,25$$

$$t_e[v_6, v_7] = \frac{1+4.3+5}{6} = 3$$

$$\delta_{t_e}^2[v_6, v_7] = \left(\frac{5-1}{6}\right)^2 = 0,44$$

Krok 2. V počátečním vrcholu v_0 položíme čas nejdříve možného začátku projektu roven nule $T_E^{v_0} = 0$, rozptyl času $T_E^{v_0}$ je také roven nule $\delta_{T_E^{v_0}}^2 = 0$.

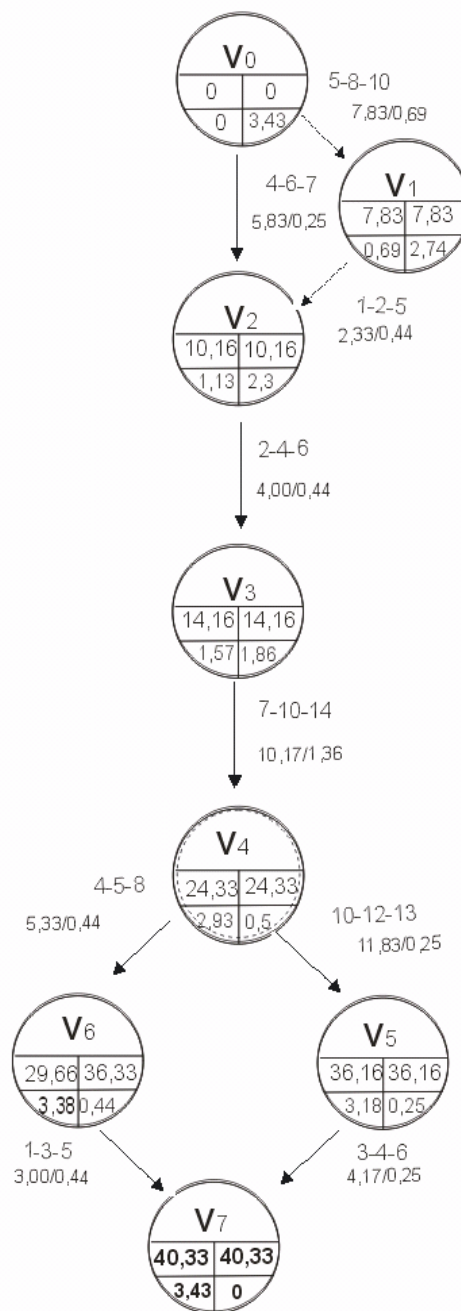
Krok 3. Dle postupu v kapitole 3.2 vypočítáme čas nejdříve možného začátku dílčích činností vycházejících z daného vrcholu - $T_E^{v_i}$ a příslušný rozptyl těchto hodnot $\delta_{T_E^{v_i}}^2$.



Obrázek 8 Síť po provedení kroku 3

Krok 4. V koncovém vrcholu v_n položíme čas nejpozději nutného ukončení činnosti projektu roven času nejdříve možného začátku činnosti v_n . $T_L^{v_n} = T_E^{v_n}$ a rozptyl času $T_L^{v_n}$ je roven nule $\delta_{T_L^{v_n}}^2 = 0$.

Krok 5. Vypočítáme čas nejpozději nutného ukončení dílčích činností vycházejících z daného vrcholu - $T_L^{v_n}$ a příslušný rozptyl těchto hodnot $\delta_{T_L^{v_n}}^2$.



Obrázek 9 Síť po provedení kroku 5

Krok 6. Určíme kritickou cestu projektu. Pro vrcholy kritické cesty platí, že čas nejdříve nutný je roven času nejpozději možnému. Pro hrany ležící na kritické cestě platí, že časový interval mezi okamžikem nejdříve možného začátku této činnosti a okamžikem jejího nejpozději nutného ukončení je roven době trvání této činnosti. Síťový graf po spočítání kritické cesty nalezneme v příloze F.

Krok 7. Určení pravděpodobnosti vzniku záporné rezervy.

Pro vrcholy ležící na kritické cestě platí, že pravděpodobnost vzniku záporné rezervy se rovná 50%. V našem případě na kritické cestě leží vyjma vrcholu v_6 všechny vrcholy síťového grafu.

$$P\{R \leq 0\} = \phi(u) = \phi(0) = 50\%$$

Pro vrchol v_6 určíme pravděpodobnost vzniku záporné rezervy následovně:

$$P\{R \leq 0\} = \phi(u) = \phi\left(\frac{29,66 - 36,33}{\sqrt{3,38 + 0,44}}\right) = 1 - \phi(3,42) = 0,03\%$$

Malá pravděpodobnost vzniku záporné rezervy na vrcholu v_6 plyne z velké časové rezervy v tomto uzlu

$$R = 36,33 - 29,66 = 6,67$$

Krok 8. Pravděpodobnost dodržení/překročení termínu

Nejdříve určíme termín T_s , pro který chceme vypočítat pravděpodobnost.

$$T_s = 42$$

$$P\{T_E^{v_i} \leq 42\} = \phi(u) = \phi\left(\frac{42 - 40,33}{\sqrt{3,43}}\right) = \phi(0,9) = 81,59\%$$

$$P\{42 \leq T_E^{v_i}\} = \phi(u) = \phi\left(\frac{40,33 - 42}{\sqrt{3,43}}\right) = 1 - \phi(0,9) = 18,41\%$$

4 Programátorská dokumentace

Tento program byl vyvinut ve vývojovém prostředí Borland C++ Builder .

Pro interpretaci sítě byla použita datová struktura binární strom x binární strom, pro který platí, že v levém podstromu jakéhokoliv uzlu mají všechny uzly hodnotu menší než je hodnota uzlu a v pravém podstromu mají všechny uzly hodnotu větší než je hodnota uzlu. Binární strom má časovou složitost $O(\log n)$, zatímco obyčejný seznam má složitost lineární. Pro procházení stromu se většinou využívá rekurzivního algoritmu. Při procházení se kontroluje, zda hodnota přidávaného nebo hledaného uzlu je větší nebo menší než hodnota aktuálního uzlu a podle toho se prvek uloží.

Seznam souborů v programu:

- *TSit.cpp, TSit.h*

Zde je definována třída *TSit*, která využívá základní třídy *TVrchol* a *THrana*.

- *konkret.cpp, konkret.h*

Tento soubor obsahuje třídy na obsluhu vrcholů a hran *TVrchol* a *THrana*.

- *distr.h, distr.cpp*

Tento soubor obsahuje třídu a metody pro práci s načtením a výpočty v tabulce distribuční funkce.

- *Unit1.cpp, Unit1.h*

Hlavní okno programu a jeho funkce

- *Unit2.cpp, Unit2.h*

Formulář pro zadávání jednotlivých dat sítě uživatelem.

- *Unit3.cpp, Unit3.h*

Okno pro výpočet pravděpodobnosti překročení či dodržení termínu.

- *Unit4.cpp, Unit4.h*

Nápověda k programu

- *Unit5.cpp, Unit5.h*

Zadání konkrétního problému.

4.1 Použité třídy

V programu jsou použity tři hlavní třídy, jedna pomocná třída a jedna třída vložená. Nejdůležitější třídy jsou TSit, TVrchol a THrana.

4.1.1 TVrchol

Tato třída obsahuje informace o vrcholu v síti.

Atributy této třídy:

int nazev – Atribut pro název vrcholu.

double TeVi – Čas udávající konec nejdříve možný.

double TlVi – Čas udávající začátek nejpozději nutný.

double SmOdTe – Směrodatná odchylka času TeVi.

double SmOdTl – Směrodatná odchylka času TlVi.

float DistribucniFunkce – Vypočítaná rezerva pro daný vrchol.

TVrchol MensiPotomek* – Atribut ukazující na menšího potomka daného vrcholu.

TVrchol VetsiPotomek* – Atribut ukazující na většího potomka daného vrcholu.

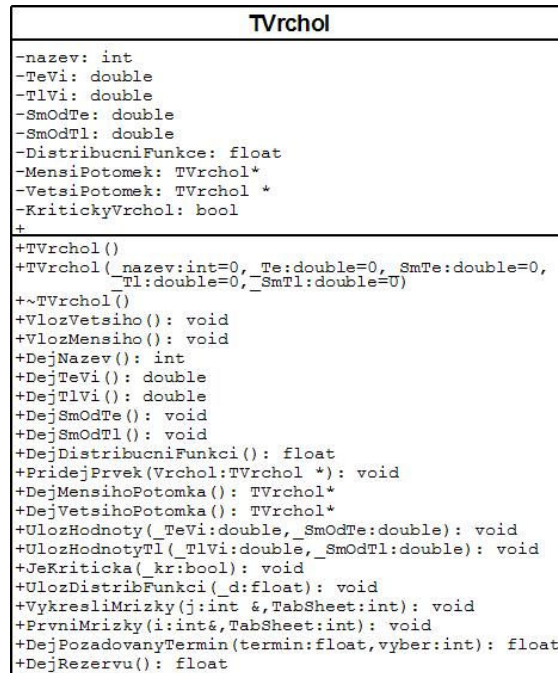
bool KritickyVrchol – Obsahuje informaci, zda daný vrchol náleží kritické cestě.

Hlavní metody třídy:

void VykresliMrizky(int &j, int TabSheet)- Vypíše hodnoty vrcholu ve formě tabulky. Pro vykreslení tabulky je použito pole prvků typu TEdit. Proměnná *j* odpovídá řádce tabulky a proměnná *TabSheet* zastupuje záložku, ve které se mají dané vrcholy vypsát.

float DejRezervu() – Podle zmíněného vzorce spočítá pravděpodobnost záporné rezervy.

float DejPozadovanyTermin(float termin, int vyber) – Algoritmus pro výpočet pravděpodobnosti dodržení či překročení termínu.



Obrázek 10 UML diagram třídy TVrchol

4.1.2 THrana

Tato třída obsahuje informace o hraně v síti.

Atributy této třídy:

int a - Optimistický odhad doby trvání činnosti.

int m - Nejpravděpodobnější odhad doby trvání činnosti.

int b - Pesimistický odhad doby trvání činnosti.

int nazev – Název (číslo) činnosti.

double Te – Střední hodnota doby trvání činnosti.

double RozptylTe – Rozptyl času *Te*.

int PocVrchol – Vrchol, ze kterého činnost vystupuje.

int KonVrchol - Vrchol, do kterého činnost vstupuje.

THrana MensiPotomek* – Atribut ukazující na menšího potomka dané hrany.

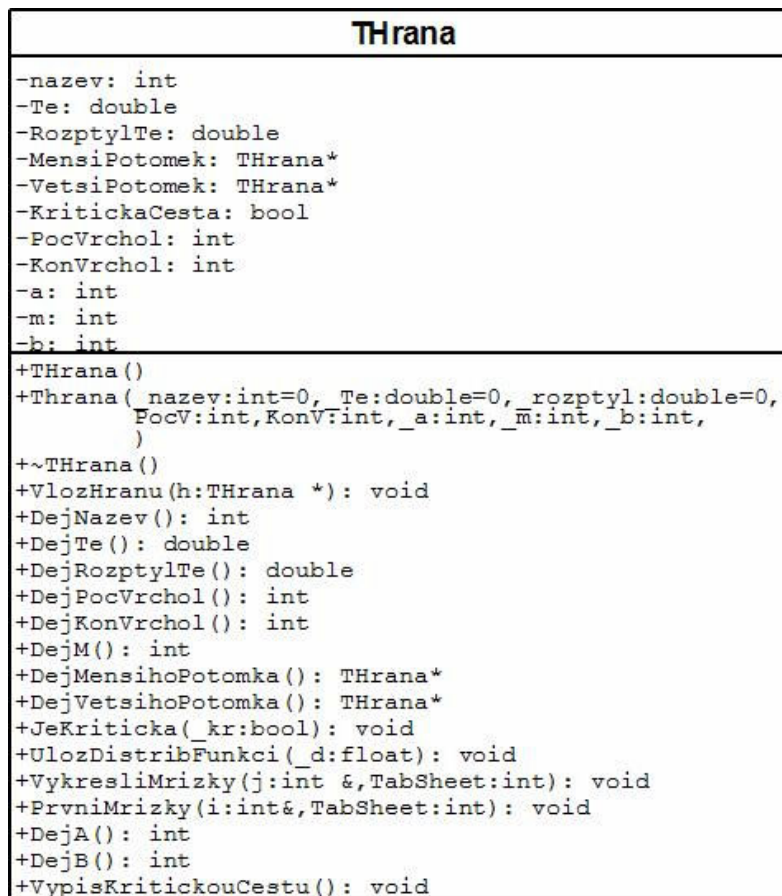
THrana VetsiPotomek* – Atribut ukazující na většího potomka dané hrany.

bool KritickaCesta – Pokud je atribut roven *true*, je tato hrana součástí kritické cesty.

Důležité metody třídy:

void VykresliMrizky(int &j, int TabSheet) - Vypíše hodnoty hrany ve formě tabulky.

void VypisKritickouCestu() – Vypíše kritickou cestu sítě.



Obrázek 11 UML diagram třídy THrana

4.1.3 TSit

Třída, která využívá základní třídy TVrchol a THrana. Kromě atributů a metod také obsahuje vnořenou třídu TZas pro práci se zásobníkem.

Hlavní atributy třídy TSit

TVrchol AktualniVrchol* – Ukazatel na aktuální vrchol sítě.

THrana AktualniCesta* – Ukazatel na aktuální hranu sítě.

int PocHran – Obsahuje počet hran sítě.

Hlavní metody třídy:

*void PridejHranu(THrana *h)* – Uloží hranu do binárního stromu pomocí rekurze. Porovnává číslo (název) přidávané hrany s aktuální hranou a na základě toho uloží hranu jako menšího nebo většího potomka aktuální hrany.

Zdrojový kód viz příloha A.

*void PridejPrvek(TVrchol *Vrchol, bool Mensi)* - Uloží vrchol do binárního

stromu.

void VypisHrany(int TabSheet) – Zavolá metodu třídy *THrana VykresliMřížky*.

Pro procházení stromu je použit zásobník. Viz příloha C.

void VypisVrcholy(int TabSheet); Zavolá metodu třídy *TVrchol VykresliMřížky*.

void SpojInformace() – Pro každou hranu najde pomocí metod *NajdiVrchol* a *NajdiHranu* počáteční a konečný vrchol. Těmto vrcholům vypočítá časy *TeVi* a *TlVi* a jejich směrodatné odchylky . Určí, zda daný vrchol a daná hrana náleží kritické cestě. Zdrojový kód viz příloha D.

void Nactizesouboru(char JmenoSouboru)* – Metoda pro načtení základních dat ze souboru. Po načtení hrany zavolá metodu *PridejHrany* a na základě koncového vrcholu hrany vytvoří tento vrchol a zavolá pro něj metou *PridejPrvek*.

Viz příloha E.

void Nacti(int i) – Metoda pro načtení základních dat od uživatele.

THrana NajdiHranu(int KonV)* – Prohledá strom a vrátí hranu s daným názvem. Pro prohledání stromu je použita rekurze – viz příloha B.

TVrchol NajdiPrvek(int NazVrcholu)* - Prohledá strom a vrátí vrchol s daným názvem.

void PravdepodobnostZaporneRezervy() – Metoda pro výpočet pravděpodobnosti záporné rezervy. Pomocí zásobníku prochází vrcholy a ke každému vrcholu uloží jeho spočítanou rezervu.

void PravdepodobnostPrekroceniTerminu(float termin,int vyber) - Metoda pro výpočet pravděpodobnosti překročení či dodržení termínu.

void SmazHrany() – Pomocí zásobníku vymaže celý binární strom hran.

void VypisDelkuProjektu() – Vypíše délku celého projektu.

void UlozHranyDoSouboru(const char JmenoSouboru,FILE *f)* – Uloží hrany do zadaného souboru.

void UlozVrcholyDoSouboru(const char JmenoSouboru,FILE *f)* - Uloží vrcholy do zadaného souboru.



Obrázek 12 UML diagram třídy TSit

4.1.3.1 TZas

Šablona třídy zásobník používaný při procházení binárního stromu. Obsahuje atribut *Next*, který ukazuje na další prvek zásobníku a atribut *Prvek*, který obsahuje informace o daném prvku zásobníku. Jediné dvě metody této třídy jsou *Vlož* pro vkládání prvku do zásobníku a *Vyber* pro vyjímání prvků ze zásobníku.

4.1.4 TMatice

Třída typu dvourozměrné pole pro práci s tabulkou distribučních funkcí, která se načte ze souboru do matice.

Důležité atributy třídy:

int PocSloupcu, PocRadku – Počet sloupců a řádků matice

*float **Pole* – Dvourozměrné pole.

Metody třídy:

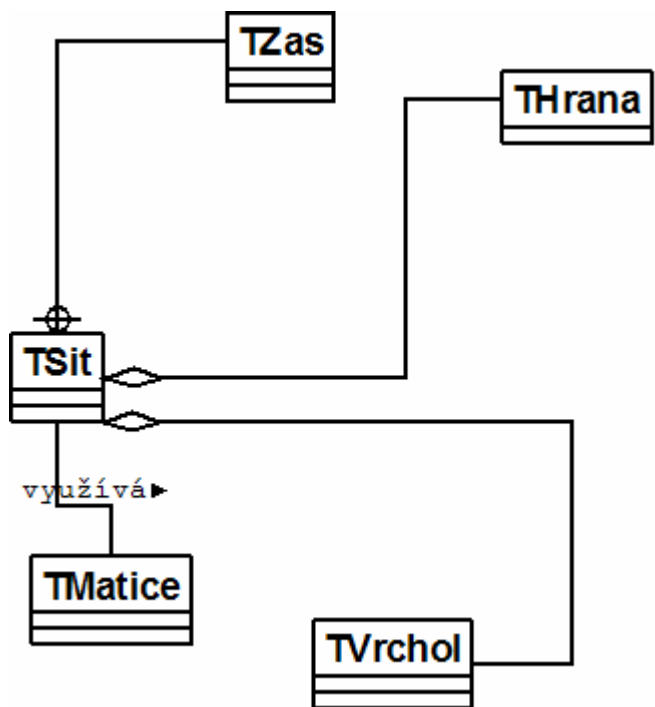
void Nacti(const char NazevSouboru)* – Načte tabulku distribučních funkcí do matice.

void Alokuj() – Alokuje matici.

void DeAlokuj() – Dealokuje matici.

float Distrib(float cislo) – Pro dané číslo nalezne hodnotu její distribuční funkce.

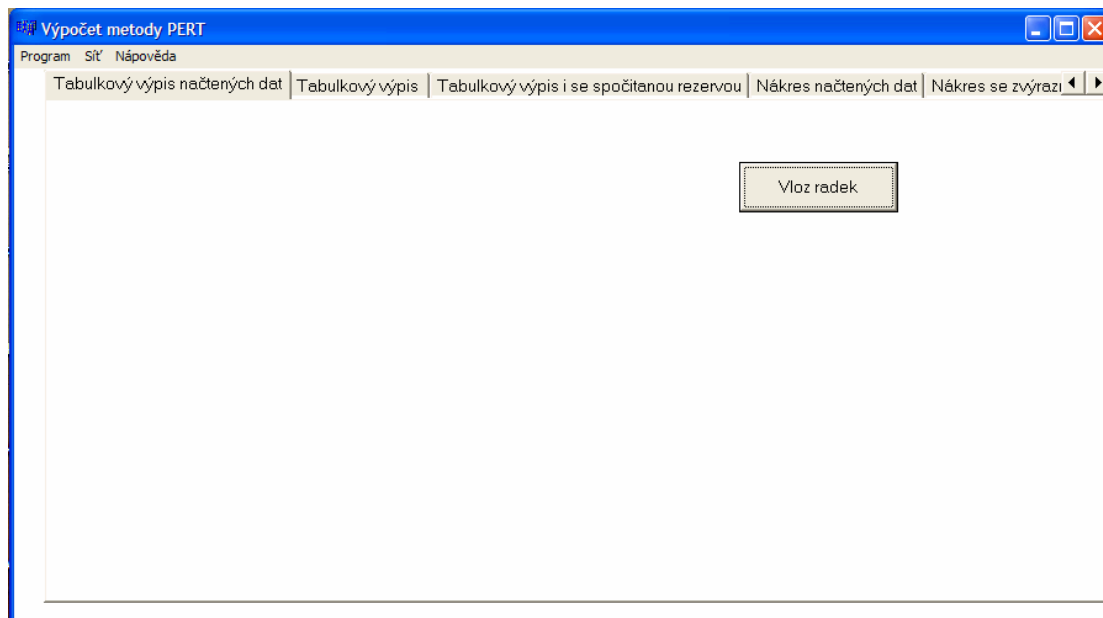
4.2 Struktura programu



Obrázek 13 Hierarchy tříd

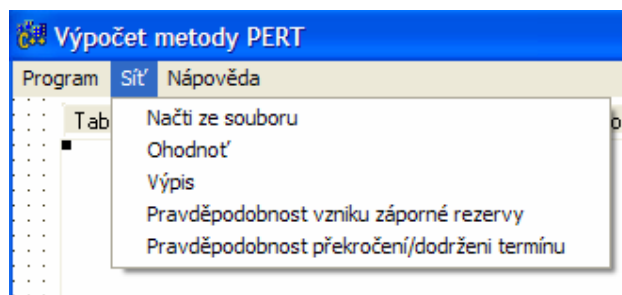
5 Uživatelská dokumentace

Po spuštění programu se před uživatelem objeví uvítací okno, následováno hlavním oknem programu. Hlavní okno obsahuje pět záložek, ve kterých jsou postupně vypisovány vypočtená data.



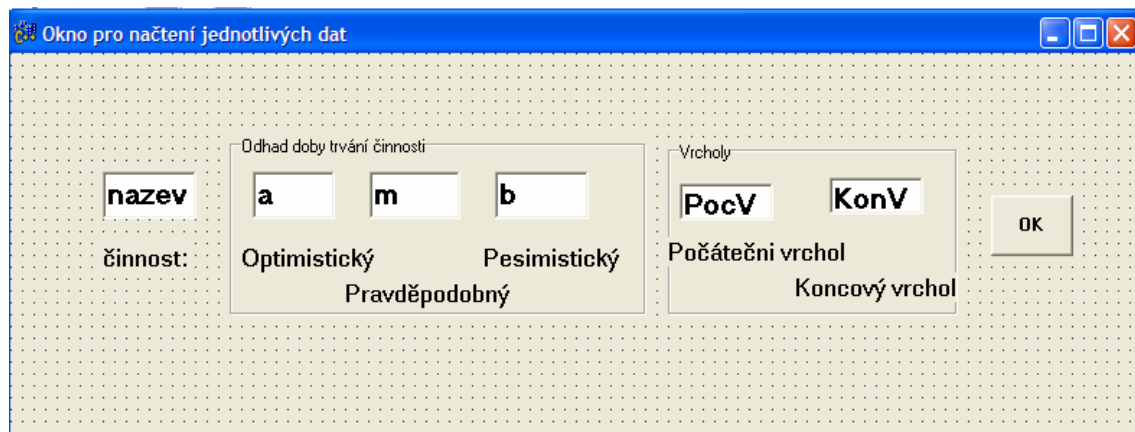
Obrázek 14 Hlavní okno programu

5.1 Hlavní obsluha programu – Menu Síť

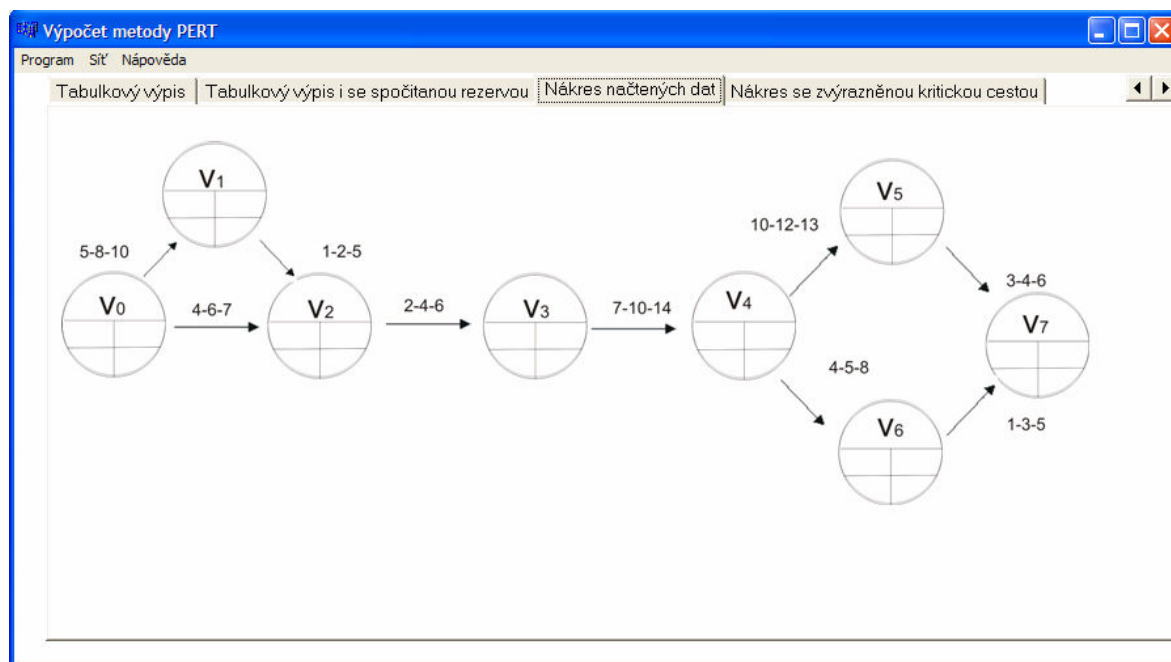


Obrázek 15 Menu Síť

Načtení počítaných dat ze souboru se provede přes hlavní menu *Síť* kliknutím na odkaz *Načti ze souboru*, nebo postupně řádek po řádku přes tlačítko *Vlož radek*. Načtená data se vypíší ve formě tabulky nebo nakreslené sítě. Ve formě tabulky se vypíší v první záložce okna nazvané *Tabulkový výpis načtených dat*, ve formě grafu se data zobrazí ve čtvrté záložce nazvané *Nákres načtených dat*.



Obrázek 16 Okno pro načtení jednotlivých hodnot



Obrázek 17 Zobrazení dat ve formě nakreslené sítě

Program Sít' Nápověda

Tabulkový výpis načtených dat | Tabulkový výpis | Tabulkový výpis i se spočítanou rezervou | Nákres načtených dat | Nákres se zvýrazní

nazev	PocVrchol	KonVrchol	a	m	b	Te	RozptylTe
1	0	1	5	8	10	7,83	0,69
2	0	2	4	6	7	5,83	0,25
3	1	2	1	2	5	2,33	0,44
4	2	3	2	4	6	4	0,44
5	3	4	7	10	14	10,17	1,36
6	4	5	10	12	13	11,83	0,25
7	4	6	4	5	8	5,33	0,44
8	5	7	3	4	6	4,17	0,25
9	6	7	1	3	5	3	0,44

Vloz radek

Obrázek 18 Výpis načtených dat ve formě tabulky

Po načtení dat se provede výpočet doby trvání projektu a zjištění kritické cesty kliknutím na *Ohodnot'* a následný výpis kliknutím na *Vypiš*. Výpis se provede opět buď ve formě tabulky ve druhé záložce pojmenované *Tabulkový výpis* nebo nakreslené síť v poslední páté záložce pojmenované *Nákres se zvýrazněnou kritickou cestou*. Kritická cesta je u tabulkového výpisu vypsána pod tabulkou. Hrany a vrcholy, které jsou součástí kritické cesty jsou v tabulce zvýrazněny šedou barvou. Pod tabulkou je také vypsána délka trvání celého projektu. U vykreslené síť je kritická cesta zvýrazněna červenými šipkami.

Tabulkový výpis načtených dat

název	TeV	TM	SmOdT	SmOdI
0	0	0	0	3,43
1	7,83	7,83	0,69	2,74
2	10,16	10,16	1,13	2,3
3	14,16	14,16	1,57	1,86
4	24,33	24,33	2,93	0,5
5	36,16	36,16	3,18	0,25
6	29,66	37,33	3,37	0,44
7	40,33	40,33	3,43	0

Tabulkový výpis i se spočítanou rezervou

název	PočVýchod	KonVýchod	a	m	b	Te	Prozplýte
1	0	1	5	8	10	7,83	0,69
2	0	2	4	6	7	5,83	0,25
3	1	2	1	2	5	2,33	0,44
4	2	3	2	4	6	4	0,44
5	3	4	7	10	14	10,17	1,36
6	4	5	10	12	13	11,83	0,25
7	4	6	4	5	8	5,33	0,44
8	5	7	3	4	6	4,17	0,25
9	6	7	1	3	5	3	0,44

Tabulkový výpis po provedení všech výpočtů

Délka celého projektu je: 40.33

Kritická cesta: [5,7] [4,5] [3,4] [2,3] [1,2] [0,1]

Obrázek 19 Výpis vypočtených dat ve formě tabulky

Dále je možné vypočítat pravděpodobnost záporné rezervy. To se provede přes nabídku *Pravděpodobnost vzniku záporné rezervy*. Výpis se provede ve třetí záložce pojmenované *Tabulkový výpis i se spočítanou rezervou*.

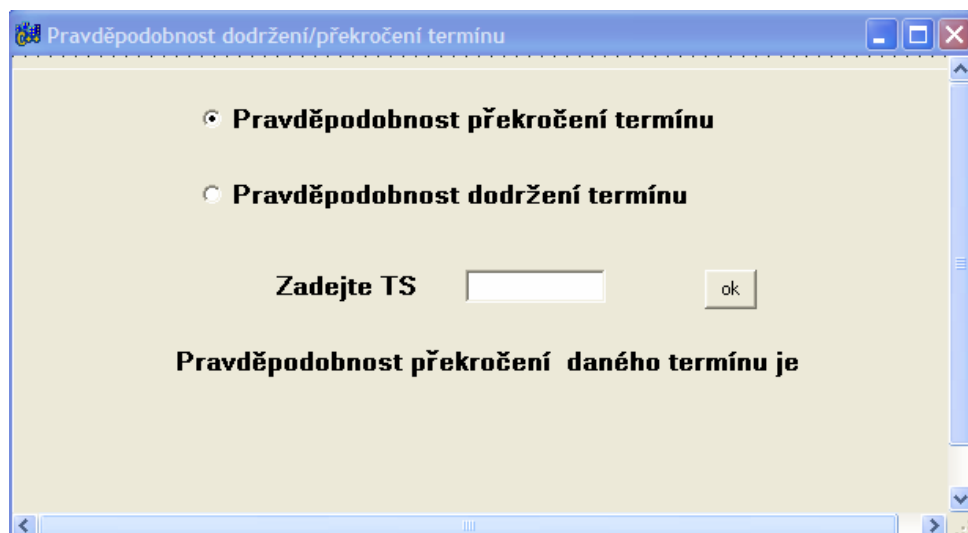
Tabulkový výpis i se spočítanou rezervou

název	TeV	TM	SmOdTe	SmOdTI	rezerva
0	0	0	0	3,43	50.00%
1	7,83	7,83	0,69	2,74	50.00%
2	10,16	10,16	1,13	2,3	50.00%
3	14,16	14,16	1,57	1,86	50.00%
4	24,33	24,33	2,93	0,5	50.00%
5	36,16	36,16	3,18	0,25	50.00%
6	29,66	37,33	3,37	0,44	0.003%
7	40,33	40,33	3,43	0	50.00%

Obrázek 20 Pravděpodobnost vzniku záporné rezervy

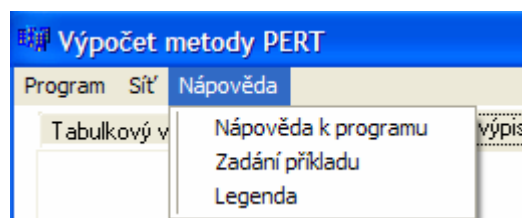
Po kritické cestě je nejdůležitější zjistit pravděpodobnost, že zadaný termín T_s bude překročen nebo dodržen. To se provede přes nabídku *Pravděpodobnost překročení/dodržení termínu*. Po kliknutí na tuto nabídku se

zobrazí okno, kde je potřeba zadat daný termín T_s a vybrat, zda se má počítat pravděpodobnost překročení či dodržení termínu. Po kliknutí na tlačítko *OK* se provede výpočet a pravděpodobnost se vypíše v procentech do spodní části okna.



Obrázek 21 Okno pro výpočet pravděpodobnosti dodržení či překročení termínu

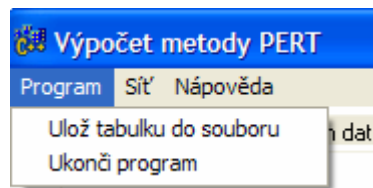
5.2 Hlavní menu *Nápověda*



Obrázek 22 Hlavní menu *Nápověda*

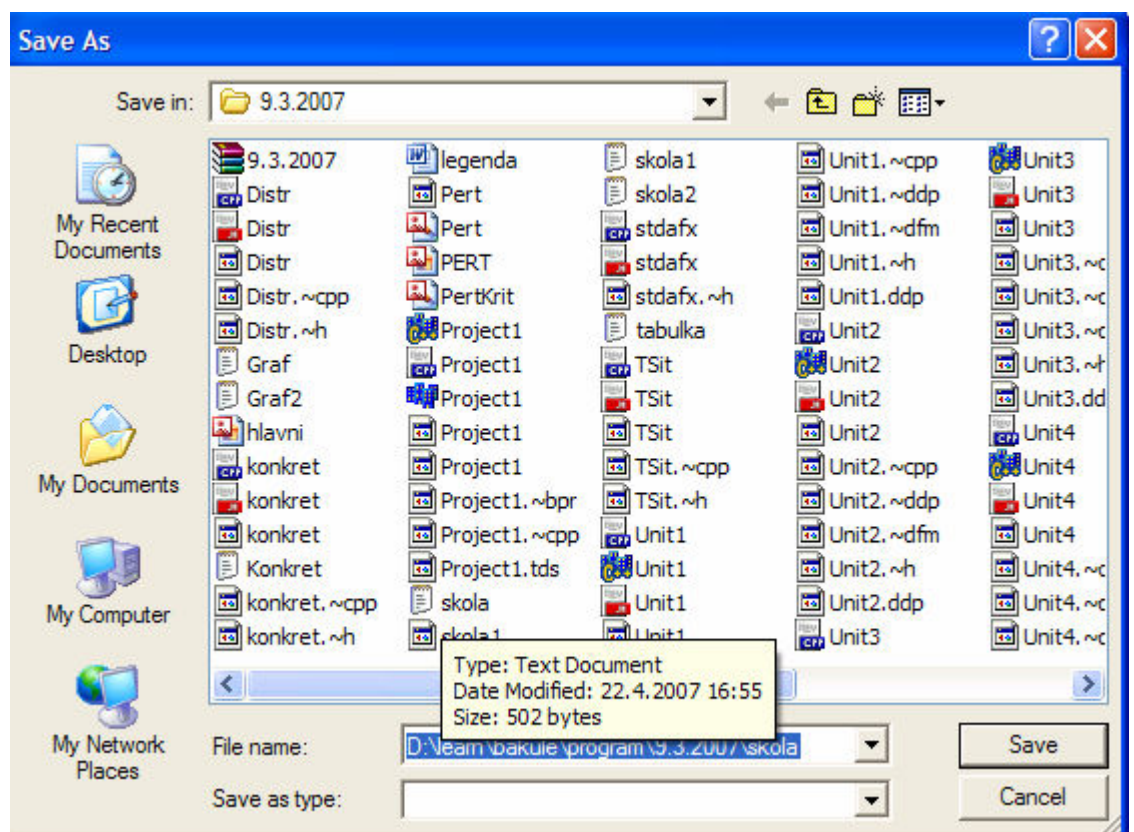
V tomto menu uživatel nalezne nápovědu k programu, Okno se zadáním příkladu a legendy k programu. *Nápověda k programu* pomůže uživateli obsluhovat daný program, poradí, jak načítat, ohodnocovat a vypisovat data. V legendě nalezneme nejdůležitější zkratky a termíny v programu.

5.3 Hlavní menu Program



Obrázek 23 Hlavní menu Program

Po výpočtu všech hodnot je možno uložit vypsání tabulky do souboru. V menu Program kliknutím na Ulož tabulku do souboru. Zobrazí se dialog pro výběr souboru, kde je možno vybrat již existující soubor, nebo zadat jméno nového souboru, který se automaticky sám vytvoří po ukončení programu. Program bude ukončen kliknutím na *Ukonči*.



Obrázek 24 Dialog pro uložení do souboru

6 Závěr

Bakalářská práce se zabývá metodami síťové analýzy, především pak metodou PERT. Teta metoda se využívá již od roku 1958 při řešení různých problémů skoro ve všech profesních odvětvích od rekonstrukce výrobní linky až po plánování pilotovaného letu na Mars.

Navržený softwarový nástroj byl vytvořen především pro efektivnější řešení zmíněné metody a optimalizován pro problematiku dané výrobní linky. Je možné ho používat při veškerých projektech, kde je používána metoda PERT.

Toto téma jsem si vybrala, neboť mě zajímá celá problematika teorie grafů, její algoritmizace i využití v praxi. Program byl vytvořen ve vývojovém prostředí Borland Builder C++, prostředí velmi podobné Delphi. Na práci jsem začala pracovat v září roku 2006. Nejdříve pouze shromažďování veškerého materiálu a učení práce s Borland Builderem. Na samotném programu jsem začala pracovat kolem v lednu roku 2007. V první řadě jsem řešila otázku vhodných datových struktur, kterou jsem konzultovala s řadou specialistů na problematiku. Poté jsem navrhla vlastní algoritmus a program se postupně začal vyvíjet do finální podoby. Samozřejmě jsem měla spoustu problémů, na které jsem časem narazila, jako například použití jedné implementace pro dva formy, ale po konzultaci se spolužáky a hledání informací na internetu jsem postupně všechny problémy vyřešila. Koncem dubna byly dopracovány i poslední úpravy v programu. Bakalářská práce byla ukončena sepsáním písemné zprávy.

Využití programového nástroje není pouze při řešení problematiky různých projektů, ale může být využíván studenty i pedagogy, kteří si tak mohou rychle zkontrolovat své vlastní výpočty a řešení. Program může být též využít jako výukový.

Příloha A – Kód pro procházení stromu a přidání jednotlivých hran

```
void TSit::PridejHranu(THrana *h){
    if (PrvniHrana==NULL)Pom=0;
    if (AktC==NULL){
        if(Pom!=NULL)Pom->VlozHranu(h);
        AktC=h;
        AktC->VlozHranu(0);
        if (PrvniHrana==NULL)PrvniHrana=h;
        else AktC=PrvniHrana;
    }
    else{
        if(AktC->DejNazev()>h->DejNazev()){
            AktC=AktC->DejMensihoPotomka();
            PridejHranu(h);
        }
        else{
            Pom=AktC;
            AktC=AktC->DejVetsihoPotomka();
            PridejHranu(h);
        }
    }
}
```

Příloha B – Zdrojový kód pro hledání hran

```
THrana* TSit::NajdiHranu(int KonV){
    if(AktC==NULL) return(AktC);
    else if(AktC->DejNazev()==KonV)return AktC;
    if(AktC->DejNazev()>KonV){
        AktC=AktC->DejMensihoPotomka();
        return(NajdiHranu(KonV));
    }
    else{
        AktC=AktC->DejVetsihoPotomka();
        return(NajdiHranu(KonV))}
}
```

Příloha C – Zdrojový kód pro výpis hran

```
void TSit::VypisHrany(int TabSheet){
    TZas<TVrchol> *zasobnik=NULL;
    if (Prvni){
        zasobnik->Vloz(Prvni,&zasobnik);
        while(zasobnik!=NULL){
            AktV=zasobnik->Vyber(&zasobnik);
            if(AktV->DejMP())zasobnik->Vloz(AktV->DejMP(),&zasobnik);
            if(AktV->DejVP())zasobnik->Vloz(AktV->DejVP(),&zasobnik);
            AktC->VykresliMrizky(i,TabSheet);
        }
    }
}
```

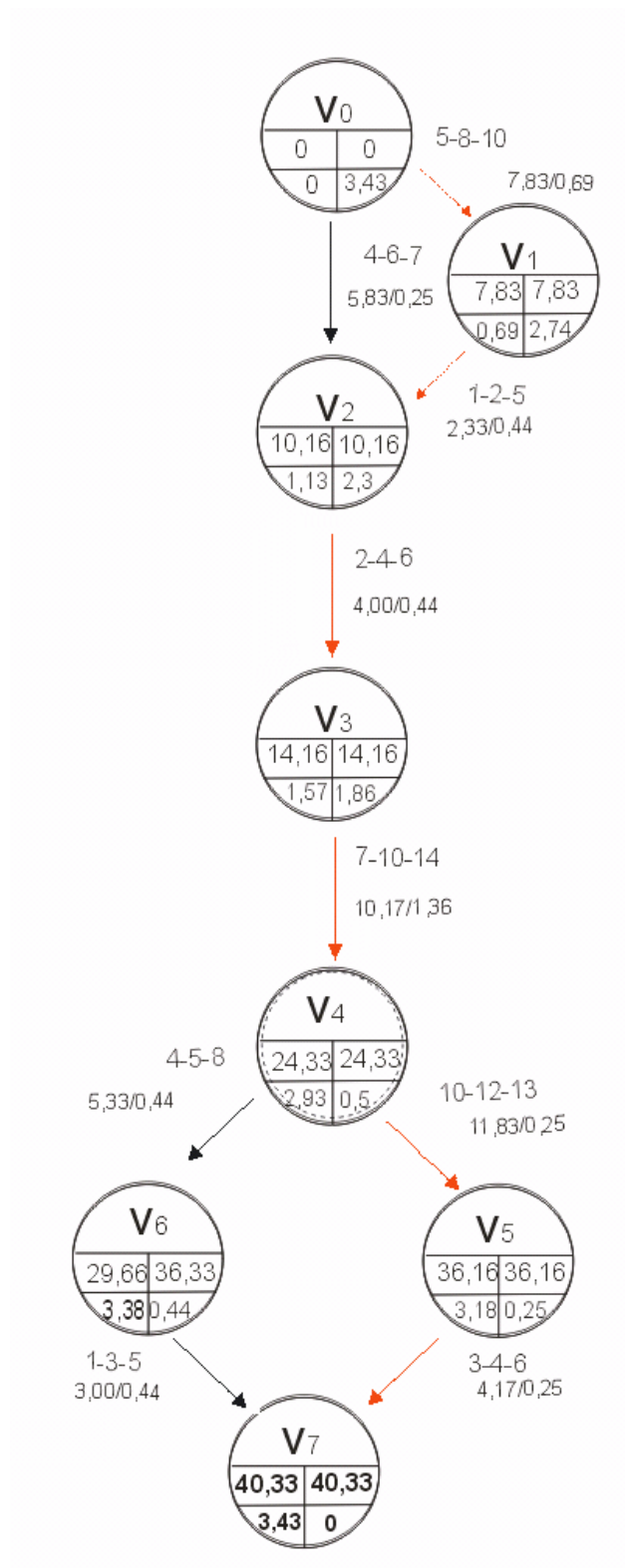
Příloha D – Kód pro propojení hran a vrcholů a ohodnocení vrcholů

```
void TSit::SpojInformace(){
    AktC=PrvniHrana;
    double Pom,Pom1,Pom2;
    for(int i=1;i<=PocHran;i++){
        AktC=PrvniHrana;
        AktC=NajdiHranu(i);
        TVrchol vrch;
        AktV=Prvni;
        AktV=Najdiprvek(AktC->DejPocVrchol());
        Pom=round(AktV->DejTeVi()+AktC->DejTe(),2);
        Pom1=round(AktV->DejSmOdTe()+AktC->DejRozptylTe(),2);
        AktV=Prvni;
        AktV=Najdiprvek(AktC->DejKonVrchol());
        AktV->UlozHodnoty(Pom,Pom1);
        if(i==PocHran){
            AktV->UlozHodnotyTl(AktV->DejTeVi(),0);
            AktC=NajdiHranu(i);
            Posledni=AktV;
        }
    }
    for(int i=PocHran;i>0;i--){
        AktC=PrvniHrana;
        AktC=NajdiHranu(i);
        TVrchol vrch;
        AktV=Prvni;
        AktV=Najdiprvek(AktC->DejKonVrchol());
        Pom=AktV->DejTlVi()-AktC->DejTe();
        Pom1=AktV->DejSmOdTl()+AktC->DejRozptylTe();
        Pom2=AktV->DejTlVi();
        AktV=Prvni;
        AktV=Najdiprvek(AktC->DejPocVrchol());
        AktV->UlozHodnotyTl(Pom,Pom1);
        if(round(AktV->DejTeVi(),2)==round(AktV->DejTlVi(),2)){
            AktV->JeKriticka(true);
            float mezi;
            mezi=round(Pom2,2)-round(AktV->DejTeVi(),2);
            if(round(mezi,2)==round(AktC->DejTe(),2)){
                AktC->JeKriticka(true);
                AktC->VypisKritickouCestu();
            }
        }
        else {
            AktV->JeKriticka(false);
            AktC->JeKriticka(false);
        }
    }
}
```

Příloha E – Zdrojový kód pro načtení dat ze souboru

```
void TSit::Nactizesouboru(char* JmenoSouboru){
int Velikost;
FILE *f;
if ((f = fopen(JmenoSouboru, "r")) == 0){
ShowMessage("Error opening file");
exit(1);
}
TVrchol *vrchol=new TVrchol(0,0,0,0,0);
PridejPrvek(vrchol,0);
while(!feof(f)){
fscanf(f,"%d %d %d %d %d %d\n",&NazHr,&PocV, &KonV, &UdA,
&UdM, &UdB);
PomTe= UdA+4*UdM+UdB;
PomTe=PomTe/6;
PomRoz=(UdB-UdA)*(UdB-UdA);
PomRoz=PomRoz/36;
THrana* hrana = new
THrana(NazHr,PocV,KonV,UdA,UdM,UdB,round(Po
mTe,2),round(PomRoz,2));
PridejHranu(hrana);
PocHran++;
v=NajdiPrvek(KonV);
if (v->DejNazev() !=KonV){
TVrchol *vrchol=new TVrchol(KonV,0,0,0,0);
PridejPrvek(vrchol,0);
}
}
PosledniHrana=NajdiHranu(PocHran);
}
```

Příloha F – Zobrazení sítě po výpočtu kritické cesty



ÚDAJE PRO KNIHOVNICKOU DATABÁZI

Název práce	Vyhodnocení síťového grafu se stochastickou dobou trvání činnosti
Autor práce	Šárka Šlehoferová
Obor	IT
Rok obhajoby	2007
Vedoucí práce	doc. Ing. Josef Volek, CSc.
Anotace	Tato práce se zabývá síťovou analýzou a jejími metodami, především pak metodou PERT.
Klíčová slova	PERT, analýza síťového grafu, CPM, kritická cesta, stochastická doba trvání činnosti

SEZNAM POUŽITÉ LITERATURY

- [1] LAUBER, J., HUŠEK, R.: *Operační výzkum*. Praha: Ministerstvo školství, mládeže a tělovýchovy ČR, 1990,
- [2] VOLEK, J., *Operační výzkum I*, Pardubice : Univerzita Pardubice, 2002, ISBN 80-7194-410-6.
- [3] WALTER, J., VEJMOLA, S., FIALA, P., *Aplikace metod síťové analýzy v řízení a plánování*, Praha: SNTL 1989, ISBN: 80-03-00101-3.
- [4] KLUSOŇ, V., *Kritická cesta a PERT v řídicí praxi*, Praha: SNTL 1973.
- [5] GRYZCZ, V., *Použití metody PERT při řízení projektů*, Dostupný z www: <http://www.fce.vutbr.cz/veda/dk2003texty/pdf/5-3/np/grycz.pdf>
- [6] DVOŘÁK, J., *Síťová analýza*, Dostupný z www: www.uai.fme.vutbr.cz/~jdvorak/vyuka/tsoa/PredO10.ppt
- [7] ŠUBRT, T., *Projektové řízení*, Dostupný z www: http://etext.czu.cz/php/skripta/kapitola.php?titul_key=77
- [8] SOUŠEK, R., *Využití PC pro plánování složitých návazných procesů*, Dostupný z www: www.cep.mdcz.cz/odd540/doc/seminar/pc_use.doc
- [9] *Síťová analýza*, Dostupný z www: <http://www.control.zcu.cz/~pesek/OA2000/user/data/OA.pdf>
- [10] KLICNAROVÁ, J., *Síťová analýza*, Dostupný z www: <http://www2.zf.jcu.cz/~janaklic/emm/pr01.pdf>